



7450 Ethernet Service Switch
7750 Service Router
7950 Extensible Routing System
Releases Up To 23.10.R2

MD-CLI Advanced Configuration Guide - Part I

3HE 19549 AAAC TQZZA
Edition: 01
March 2024

Nokia is committed to diversity and inclusion. We are continuously reviewing our customer documentation and consulting with standards bodies to ensure that terminology is inclusive and aligned with the industry. Our future customer documentation will be updated accordingly.

This document includes Nokia proprietary and confidential information, which may not be distributed or disclosed to any third parties without the prior written consent of Nokia.

This document is intended for use by Nokia's customers ("You"/"Your") in connection with a product purchased or licensed from any company within Nokia Group of Companies. Use this document as agreed. You agree to notify Nokia of any errors you may find in this document; however, should you elect to use this document for any purpose(s) for which it is not intended, You understand and warrant that any determinations You may make or actions You may take will be based upon Your independent judgment and analysis of the content of this document.

Nokia reserves the right to make changes to this document without notice. At all times, the controlling version is the one available on Nokia's site.

No part of this document may be modified.

NO WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF AVAILABILITY, ACCURACY, RELIABILITY, TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, IS MADE IN RELATION TO THE CONTENT OF THIS DOCUMENT. IN NO EVENT WILL NOKIA BE LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL OR ANY LOSSES, SUCH AS BUT NOT LIMITED TO LOSS OF PROFIT, REVENUE, BUSINESS INTERRUPTION, BUSINESS OPPORTUNITY OR DATA THAT MAY ARISE FROM THE USE OF THIS DOCUMENT OR THE INFORMATION IN IT, EVEN IN THE CASE OF ERRORS IN OR OMISSIONS FROM THIS DOCUMENT OR ITS CONTENT.

Copyright and trademark: Nokia is a registered trademark of Nokia Corporation. Other product names mentioned in this document may be trademarks of their respective owners.

© 2024 Nokia.

Table of contents

List of tables.....	6
List of figures.....	8
Preface.....	25
About This Guide.....	25
System Management.....	26
Event Handling System.....	27
SR OS NETCONF Server Basics.....	40
Interface Configuration.....	43
Port Cross-Connect (PXC).....	44
Router Configuration.....	75
6PE Next-Hop Resolution.....	76
Aggregate Route Indirect Next-Hop Option.....	98
Bi-Directional Forwarding Detection.....	106
LFA Policies Using OSPF as IGP.....	142
PBR/PBF Redundancy.....	164
Weighted ECMP for 6PE over RSVP-TE LSPs.....	190
Unicast Routing Protocols.....	200
Advertising IPv4 NLRI with IPv6 Next-Hop.....	201
Associating Communities with Static and Aggregate Routes.....	211
BGP Add-Path.....	237
BGP Add-Path Policy Control.....	267
BGP Autonomous System Override.....	284
BGP Conditional Route Advertisement.....	300
BGP Convergence — Delayed Route Advertisement.....	313
BGP Default Route Origination.....	329
BGP Fast Reroute.....	343
BGP Fast Reroute Policy Control.....	357
BGP FlowSpec for IPv4 and IPv6.....	377

BGP Multipath.....	396
BGP Optimal Route Reflection for Hierarchical Networks.....	425
BGP Optimal Route Reflection for Non-Hierarchical Networks.....	442
BGP Prefix Limit per Address Family.....	458
BGP Remove-Private ASN.....	470
BGP Route Leaking.....	490
BGP Route Refresh.....	526
BGP Unresolved Route Leaking from Base Router to VPRN.....	538
BGP Weighted ECMP.....	560
Dynamic BGP Peers.....	578
EBGP Default Reject Policy.....	592
EBGP Route Resolution to a Static Route.....	601
Flexible Algorithm for IS-IS.....	617
IS-IS Link Bundling.....	637
Next-Hop Resolution for Labeled BGP Routes.....	652
Policy Chaining and Logical Expressions.....	687
Pop-Label for /32 Label-IPv4 BGP Routes.....	718
Route Policy Action to Suppress BGP Route Installation.....	731
Separate BGP RIBs for Labeled Routes.....	746
MPLS.....	769
Automatic Creation of RSVP-TE LSPs.....	770
BFD for RSVP-TE and LDP LSPs.....	786
BFD for RSVP-TE LSPs with Failure Action.....	805
DiffServ Traffic Engineering.....	819
IGP Shortcuts.....	858
Inter-Area TE Point-to-Point LSPs.....	914
LDP FEC to BGP Label Route Stitching.....	938
LDP over RSVP Using OSPF as IGP.....	967
LDP Point-to-Point LSPs.....	1008
LDP-IGP Synchronization.....	1023
LDP-SR Stitching for IPv4 Prefixes (IS-IS).....	1035
MPLS LDP FRR using ISIS as IGP.....	1051
Multicast Label Distribution Protocol.....	1076
Path MTU Discovery.....	1108
Remote Loop-Free Alternate Node Protection.....	1130

RSVP Point-to-Point LSPs.....	1147
Seamless MPLS: Isolated IGP/LDP Domains and Labeled BGP.....	1189
Shared Risk Link Groups for RSVP-Based LSPs.....	1210
Static Point-to-Point LSPs.....	1227
Topology-Independent Loop-Free Alternate for Link Protection.....	1238
Tunneling of ICMP Reply Packets over MPLS LSPs.....	1265
Unnumbered Interfaces in RSVP-TE and LDP.....	1289
Segment Routing and PCE.....	1336
BGP Segment Routing Using the Prefix SID Attribute.....	1337
BGP Signaled Segment Routing Policy.....	1350
Inter-AS Model C VPRN Using MPLS Forwarding Policies and Segment Routing Policies.....	1373
Parallel Adjacency Sets in Segment Routing.....	1394
Seamless BFD for SR-TE LSPs.....	1411
Segment Routing – Traffic Engineered Tunnels.....	1434
Segment Routing over IPv6.....	1453
Segment Routing over IPv6 for VPRN.....	1481
Segment Routing with IS-IS Control Plane.....	1505
SR-TE LSP Path Computation Using Local CSPF.....	1533
SRv6 Encapsulation in the Base Routing Instance.....	1564
SRv6 Loop-Free Alternate.....	1597
VSR Installation and Setup.....	1635
VSR Hypervisor Configuration.....	1636

List of tables

Table 1: Primary and secondary forwarding actions.....	169
Table 2: Supported address families for BGP prefix limit.....	458
Table 3: Status of the links A, B, C, and D.....	639
Table 4: Default preferences in route table.....	660
Table 5: Policy chaining versus policy logical expressions.....	687
Table 6: Boolean values for the policy actions.....	698
Table 7: Actions for the logical operators.....	699
Table 8: Mapping the final result of an expression to a policy action.....	699
Table 9: Assigned LP and communities for the import logical expressions.....	705
Table 10: Assigned LP and communities for the import logical expressions.....	708
Table 11: Assigned LP for the import logical expressions.....	714
Table 12: Comparison bandwidth constraint models.....	826
Table 13: RSVP LSP Role As Outcome of LSP Level and IGP Level Configuration Options.....	881
Table 14: Terminology.....	1076
Table 15: MTU types.....	1108
Table 16: MTU values for Ethernet frames.....	1109
Table 17: Values of the max-sr-frr-labels parameter in TI-LFA.....	1252
Table 18: Use of CO bits.....	1357
Table 19: RFC 7880 S-BFD terms.....	1413
Table 20: SRv6 shortest path routing.....	1453
Table 21: SRv6 source routing.....	1454

Table 22: SRv6 endpoint behaviors supported in SR OS Release 21.10.R1.....	1455
Table 23: Mode comparison.....	1508

List of figures

Figure 1: Example topology.....	28
Figure 2: NETCONF client-server communication.....	40
Figure 3: Example topology.....	45
Figure 4: Non-redundant PXC.....	47
Figure 5: PXC redundant mode with LAG.....	50
Figure 6: AS mode with redundant FPE.....	62
Figure 7: IPv6 provider edge (6PE).....	76
Figure 8: Example topology.....	78
Figure 9: 6PE next hop resolved to an LDP tunnel.....	83
Figure 10: 6PE next hop resolved to an RSVP-TE tunnel.....	85
Figure 11: 6PE next hop resolved to an SR-ISIS tunnel.....	89
Figure 12: Example topology for seamless MPLS.....	89
Figure 13: Configured protocols for seamless MPLS.....	91
Figure 14: BGP labeled IPv4 tunnel for 192.0.2.4/32 using LDP tunnels.....	97
Figure 15: Aggregate routes.....	98
Figure 16: Example topology.....	99
Figure 17: BFD centralized sessions.....	108
Figure 18: BFD interface configuration.....	109
Figure 19: BFD for ISIS.....	112
Figure 20: BFD for OSPF.....	115
Figure 21: BFD for OSPF and PIM.....	117

Figure 22: BFD for static routes.....	118
Figure 23: BFD for IES over spoke SDP.....	121
Figure 24: BFD for RSVP.....	125
Figure 25: BFD for T-LDP.....	128
Figure 26: BFD for OSPF PE-CE interfaces.....	131
Figure 27: BFD for VRRP.....	134
Figure 28: Example topology.....	143
Figure 29: PBF in the "VPLS-3" service on PE-1.....	165
Figure 30: Example topology.....	170
Figure 31: PBF in the "VPLS-1" service on PE-1.....	171
Figure 32: PBR in a VPRN.....	185
Figure 33: Weighted ECMP in AS 64496.....	191
Figure 34: Example Topology for 6PE over RSVP-TE LSPs.....	193
Figure 35: Capability value field format.....	201
Figure 36: Example topology with IPv6 interfaces.....	203
Figure 37: Loopback addresses and advertised IPv4, label-IPv4, and VPN-IPv4 routes.....	203
Figure 38: Example topology.....	212
Figure 39: CE connections for next-hops.....	214
Figure 40: CE-7 connectivity.....	227
Figure 41: CE-6 connectivity.....	231
Figure 42: RR advertises best path only – path A preferred over path B.....	238
Figure 43: Reconvergence after path failure (without add-path).....	239
Figure 44: Advertised paths when BGP add-path is enabled in PEs and RR.....	240

Figure 45: Reconvergence after path failure when BGP add-path is enabled.....	241
Figure 46: Example topology.....	244
Figure 47: Example topology with VPRNs.....	259
Figure 48: BGP add-paths before policy control.....	268
Figure 49: BGP add-paths after policy control.....	268
Figure 50: Example topology - IPv4.....	269
Figure 51: Example topology - VPN-IPv4.....	276
Figure 52: PE-2 detects AS-path loop and advertises the route to PE-3 as invalid.....	284
Figure 53: BGP AS override replaces the peer ASN in the AS-path with the local ASN.....	285
Figure 54: Example topology.....	286
Figure 55: PE-2 detects AS loop and advertises a route to PE-3 as invalid.....	290
Figure 56: No AS loop when BGP AS override is enabled for group "eBGP" on PE-2 and PE-4.....	292
Figure 57: Example topology with VPRN 1 on all PEs.....	293
Figure 58: AS loop when BGP AS override is not configured in VPRN 1 on PE-2.....	296
Figure 59: Routes advertised when BGP AS override is enabled in VPRN 1 on the PEs.....	297
Figure 60: Conditional BGP Route Advertisement - ISP Peering.....	300
Figure 61: Conditional BGP Route Advertisement Implementation Example.....	301
Figure 62: Example Topology.....	302
Figure 63: Default SR OS behavior when the BGP process restarts.....	314
Figure 64: BGP convergence tuning with delayed route advertisement.....	314
Figure 65: BGP convergence timers.....	315
Figure 66: BGP convergence states.....	316
Figure 67: Example topology.....	317

Figure 68: Example topology with IPv4 addresses.....	330
Figure 69: Example topology with IPv6 addresses.....	331
Figure 70: Core PIC.....	344
Figure 71: Edge PIC.....	344
Figure 72: BGP FRR topology.....	345
Figure 73: Community addition on PE-1 and PE-2.....	358
Figure 74: FRR policy on PE-3.....	359
Figure 75: Example topology - IPv4.....	360
Figure 76: Example topology - VPN-IPv4.....	368
Figure 77: Example topology.....	379
Figure 78: Example topology.....	398
Figure 79: BGP multipath with eBGP limit 2.....	401
Figure 80: eBGP multipath with limit 2 and ECMP disabled.....	402
Figure 81: BGP multipath with iBGP limit 3 and ECMP limit 8.....	404
Figure 82: BGP multipath with limit 6 and eBGP preferred.....	405
Figure 83: BGP multipath with limit 6, eBGP equal to iBGP, and other path options identical.....	407
Figure 84: BGP multipath configured with restriction to the same neighbor AS.....	409
Figure 85: BGP multipath restricted to the same neighbor AS: AS paths with same length.....	410
Figure 86: BGP multipath restricted to the same neighbor AS: AS paths of different lengths.....	411
Figure 87: BGP multipath restricted to the same neighbor AS: AS paths of different lengths, AS path ignored.....	412
Figure 88: BGP multipath restricted to exact same AS. All AS paths are different.....	414
Figure 89: BGP multipath restricted to exact same AS. All AS paths are identical.....	415

Figure 90: BGP multipath for the IPv4 address family.....	417
Figure 91: BGP multipath for the label-IPv4 address family.....	419
Figure 92: BGP multipath for the label-IPv6 address family.....	420
Figure 93: Best IPv4 path originates from a non-multipath-eligible BGP neighbor.....	422
Figure 94: Two IPv4 paths from multipath-eligible BGP peers are used.....	424
Figure 95: Centralized route reflection.....	426
Figure 96: Centralized route reflection with ORR.....	427
Figure 97: Example hierarchical networking using OSPF.....	429
Figure 98: Suboptimal route reflection.....	436
Figure 99: Optimal route reflection.....	440
Figure 100: Centralized route reflection.....	443
Figure 101: Centralized route reflection with ORR.....	444
Figure 102: Example non-hierarchical networking using IS-IS.....	446
Figure 103: Suboptimal route reflection.....	453
Figure 104: Optimal route reflection.....	456
Figure 105: Post-import option.....	459
Figure 106: Example topology.....	460
Figure 107: Use case 1 topology.....	471
Figure 108: PE-2 adds its ASN and keeps all ASNs in the AS path (default).....	474
Figure 109: PE-2 adds its own ASN and removes all private ASNs.....	475
Figure 110: PE-2 adds its own ASN and replaces all private ASNs with its own ASN.....	477
Figure 111: Use case 2 topology.....	477
Figure 112: PE-2 adds its own private ASN and its public ASN (default).....	479

Figure 113: PE-2 adds only its own public ASN when local ASN is configured as private.....	480
Figure 114: PE-2 removes the private ASNs until the first public ASN.....	481
Figure 115: PE-2 replaces the private ASNs until the first public ASN.....	483
Figure 116: Use case 3 topology with private ASN 64513 on CE-1 and CE-6.....	483
Figure 117: PE-2 adds its public ASN to the AS path.....	485
Figure 118: PE-2 removes the private ASNs except peer ASN 64513.....	486
Figure 119: PE-2 replaces the private ASNs except peer ASN 64513.....	488
Figure 120: BGP route leaking process.....	491
Figure 121: Example topology.....	492
Figure 122: BGP IPv4 route leaking between VPRNs.....	493
Figure 123: BGP IPv4 route leaking from VPRN to GRT.....	503
Figure 124: BGP IPv4 route leaking from GRT to VPRN.....	509
Figure 125: BGP IPv6 route leaking between VPRNs.....	514
Figure 126: BGP IPv6 route leaking from GRT and VPRN to VPRN.....	518
Figure 127: Example topology.....	527
Figure 128: BGP route leaking process between BGP routing instances X and Y.....	538
Figure 129: Example topology.....	540
Figure 130: Leaked route 10.14.0.0/16 with next-hop resolved in VPRN 1 using IS-IS.....	544
Figure 131: Leaked route 10.24.0.0/16 with next-hop resolved in VPRN 2 using VPN-IP.....	549
Figure 132: Leaked route 10.34.0.0/16 with next-hop resolved in VPRN 2 using eBGP.....	554
Figure 133: Standard ECMP - Equal Bandwidth Links.....	561
Figure 134: Standard ECMP - Unequal Bandwidth Links.....	561
Figure 135: Link Bandwidth Extended Community Advertisement.....	562

Figure 136: Weighted ECMP - Unequal Bandwidth Links.....	563
Figure 137: Weighted ECMP - Link Aggregation Group.....	563
Figure 138: Standard ECMP - Unequal Bandwidth Links with eBGP.....	564
Figure 139: Weighted ECMP - Unequal Bandwidth Links with VPRN.....	564
Figure 140: Example Topology - BGP Weighted ECMP for IPv4 Family.....	566
Figure 141: Establishing dynamic BGP sessions.....	578
Figure 142: Dynamic BGP peers.....	580
Figure 143: Example topology with VPRN 1 in different ASs.....	587
Figure 144: Example topology.....	593
Figure 145: Advertised BGP and BGP-LU IPv4 routes.....	595
Figure 146: Advertised BGP and BGP-LU IPv6 routes.....	595
Figure 147: Example topology.....	602
Figure 148: BGP peering.....	602
Figure 149: IS-IS FAD sub-TLV.....	618
Figure 150: Application Identifier Bit Mask.....	619
Figure 151: Flexible Algorithm example in an SR-MPLS domain.....	621
Figure 152: Example topology.....	622
Figure 153: Example topology with modified IS-IS Level-1/2 capabilities.....	634
Figure 154: Link bundle schematic.....	637
Figure 155: Effect of single link failure on bundle group.....	638
Figure 156: Double link failure.....	639
Figure 157: Example topology.....	640
Figure 158: Link failure.....	647

Figure 159: Second link failure.....	649
Figure 160: Example topology.....	654
Figure 161: VPRN 1 in AS 64496.....	670
Figure 162: VPRN 2 in AS 64496 and in AS 64500.....	673
Figure 163: VPRN 3 - inter-AS VPRN model C.....	681
Figure 164: Example topology.....	700
Figure 165: Stitching RSVP/LDP tunnels to BGP tunnels.....	718
Figure 166: Example topology.....	720
Figure 167: Example topology.....	732
Figure 168: PE-1 exports BGP IPv4 and BGP-LU IPv4 routes to RR-2.....	734
Figure 169: RR-1 with separate labeled-IPv4 RIB implementation.....	747
Figure 170: Seamless MPLS - Separate labeled-IPv4 implementation.....	748
Figure 171: System architecture with separate RIBs for labeled-unicast and unlabeled routes.....	749
Figure 172: Example IPv4 topology.....	750
Figure 173: BGP sessions.....	751
Figure 174: PE-1 applies next-hop-self toward neighbor PE-2.....	756
Figure 175: Applying next-hop-self to unlabeled IP-4 routes to neighbor PE-2.....	758
Figure 176: PE-1 advertises prefixes 1.1.1.1/32 and 11.11.11.11/32.....	760
Figure 177: RR with labeled and unlabeled BGP sessions.....	763
Figure 178: Updates from unlabeled sessions not propagated to labeled sessions (default).....	765
Figure 179: RIB leaking from IPv4 BGP RIB to labeled-IPv4 BGP RIB.....	767
Figure 180: Example Topology.....	771
Figure 181: IGP Shortcuts with RSVP-TE Auto-Mesh.....	774

Figure 182: Example Topology for Single-Hop LDP over RSVP with ECMP.....	782
Figure 183: MPLS LSP BFD session establishment: BFD handshake.....	787
Figure 184: BFD for RSVP-TE LSPs - topology.....	788
Figure 185: BFD for LDP LSPs - topology.....	798
Figure 186: Topology for failure action failover.....	806
Figure 187: Topology for failure action down.....	813
Figure 188: Example topology.....	819
Figure 189: Mapping of TE classes.....	820
Figure 190: Bandwidth reservation for the CTs.....	822
Figure 191: Bandwidth reservation in Maximum Allocation Model for three CTs.....	826
Figure 192: Bandwidth reservation in Russian Doll Model for three CTs.....	827
Figure 193: Paths from PE-1 to PE-3.....	832
Figure 194: MAM bandwidth allocation.....	835
Figure 195: Reserved and unreserved bandwidth.....	837
Figure 196: Reserved and unreserved bandwidth on PE-1.....	840
Figure 197: Bandwidth reservation.....	841
Figure 198: Russian Doll Model for three class types.....	843
Figure 199: Reserved bandwidth for LSP with CT2 (one session).....	846
Figure 200: Bandwidth reservation for LSP with CT2 and LSP with CT1 (two sessions).....	848
Figure 201: Reserved bandwidth on both interfaces of PE-1 (three sessions).....	849
Figure 202: Reserved bandwidth on both interfaces on PE-1 (four sessions).....	851
Figure 203: Reserved bandwidth on both interfaces of PE-1 (five sessions).....	852
Figure 204: Reserved bandwidth on both interfaces on PE-1 (six sessions).....	855

Figure 205: Normal SPF Tree Sourced by PE-1.....	859
Figure 206: SPF Tree Sourced by PE-1 Using LSP Shortcuts.....	859
Figure 207: Example Topology.....	860
Figure 208: LSPs Between PE-1 and PE-6.....	874
Figure 209: RSVP Shortcuts LFA Use Case Example.....	887
Figure 210: Network Topology to Verify Installation of Shortcuts into the RTM.....	890
Figure 211: Shortcuts Within a VRF Topology Network.....	907
Figure 212: Inter-area TE LSP setup.....	915
Figure 213: Inter-area TE LSP path.....	916
Figure 214: ABR protection.....	925
Figure 215: Protection of all nodes/links along the LSP path.....	926
Figure 216: Admin group example.....	929
Figure 217: Share Risk Link Groups.....	933
Figure 218: LDP FEC to BGP label route stitching.....	939
Figure 219: Example topology.....	941
Figure 220: BGP enabled with P-4 as RR.....	944
Figure 221: End-to-end Epipe service.....	950
Figure 222: Label stacks for traffic from AN-1 to AN-8.....	957
Figure 223: Block BGP label bindings to LDP DU peer PE-9.....	962
Figure 224: Initial example topology.....	968
Figure 225: VPRN 1 with LDP over RSVP and no intra-area PE connectivity.....	984
Figure 226: VPRN 1 with LDP over RSVP and intra-area PE connectivity.....	1004
Figure 227: Generic MPLS network, MPLS label operations.....	1009

Figure 228: MPLS example topology.....	1010
Figure 229: Example topology.....	1027
Figure 230: Shortest path between PE-1 and PE-5.....	1027
Figure 231: Rerouting via P-3 and P-4 until LDP synchronization timer terminates.....	1030
Figure 232: Restored link with one LDP synchronization timer terminated.....	1032
Figure 233: Example topology.....	1036
Figure 234: Initial example topology.....	1052
Figure 235: Data verification in the direction from PE-1 to PE-5 using Epipe service.....	1063
Figure 236: LFA computation: Inequality 1 for prefix PE-5 (D) on PE-1 (S).....	1069
Figure 237: LFA computation: Inequality 3 for prefix PE-5 (D) on PE-1 (S).....	1070
Figure 238: IS-IS overload on PE-2, Inequality 1 for 192.168.24.0/30 (D) on PE-1 (S).....	1075
Figure 239: Setup of mLDP P2MP LSP.....	1077
Figure 240: Example topology.....	1078
Figure 241: LDP P2MP LSP.....	1085
Figure 242: New LDP P2MP LSP after metric change.....	1103
Figure 243: L2 services MTUs for Ethernet frames.....	1110
Figure 244: Minimum network port MTU for Ethernet frames in MPLS encapsulation.....	1111
Figure 245: Path MTU.....	1111
Figure 246: ICMP "Destination Unreachable" Message - Fragmentation Needed.....	1113
Figure 247: ICMPv6 "Packet Too Big" message.....	1113
Figure 248: Example topology.....	1114
Figure 249: Multiple Epipes Using LDP SDPs.....	1118
Figure 250: Multiple Epipes between PE-1 and PE-4 - IPv6.....	1121

Figure 251: BGP-IPv4.....	1123
Figure 252: BGP-IPv6.....	1126
Figure 253: LFA node protection - topology & denominations.....	1131
Figure 254: Node protecting extended P-space.....	1132
Figure 255: Link protecting Q-space.....	1132
Figure 256: One candidate PQ-router – repair tunnel.....	1133
Figure 257: Two candidate PQ routers – repair tunnel.....	1134
Figure 258: Example topology.....	1135
Figure 259: Link protection extended P-space calculation.....	1138
Figure 260: Link protecting Q-space calculation.....	1138
Figure 261: Repair tunnel.....	1139
Figure 262: Node protecting extended P-space calculation.....	1142
Figure 263: Link protecting Q-space calculation.....	1143
Figure 264: Validating candidate PQ routers - repair tunnel calculation.....	1144
Figure 265: Generic MPLS network, MPLS label operations.....	1148
Figure 266: MPLS example topology.....	1150
Figure 267: LSP with dynamic path takes IGP best route.....	1158
Figure 268: RSVP-TE LSP with dynamic path using TE metric.....	1161
Figure 269: Fast reroute one-to-one detour tunnels.....	1164
Figure 270: FRR facility bypass tunnels.....	1167
Figure 271: FRR facility without node protection.....	1171
Figure 272: Admin groups 'blue' and 'red'.....	1173
Figure 273: LSP and bypass within admin group 'blue'.....	1175

Figure 274: LSP and FRR bypass tunnel excluding admin group 'red'.....	1177
Figure 275: Seamless MPLS - network topology, control and data plane.....	1190
Figure 276: Seamless MPLS - IGP/LDP domains.....	1191
Figure 277: Seamless MPLS - BGP.....	1193
Figure 278: End-to-End Epipe service.....	1197
Figure 279: L3 VPN service.....	1200
Figure 280: Label stacks for traffic from AN-1 to AN-5.....	1203
Figure 281: Example topology.....	1211
Figure 282: SRLG topology.....	1212
Figure 283: Path primary RSVP-TE LSP.....	1217
Figure 284: FRR bypass tunnels originating in PE-1 with and without SRLG.....	1219
Figure 285: SRLG for secondary path.....	1222
Figure 286: SRLG database example.....	1225
Figure 287: Generic MPLS network, MPLS label operations.....	1228
Figure 288: MPLS example topology.....	1229
Figure 289: Static LSP running over PE-1, PE-2, PE-5, PE-6.....	1231
Figure 290: Post-failure LFA path does not match post-convergence path.....	1239
Figure 291: Post-failure TI-LFA path matches post-convergence path.....	1241
Figure 292: Example topology.....	1242
Figure 293: Example topology with regular LFA configured on PE-4.....	1243
Figure 294: No post-failure LFA path when PE-4 loops back traffic.....	1246
Figure 295: Example topology for remote LFA.....	1248
Figure 296: PQ node in remote LFA.....	1249

Figure 297: Extended P space of PE-1 and Q space of PE-4 are one hop apart.....	1252
Figure 298: Directed LFA with P router and Q router one hop apart.....	1254
Figure 299: Post-failure TI-LFA path coincides with post-convergence path.....	1255
Figure 300: Use of TTL: uniform versus pipe.....	1266
Figure 301: Use of TTL in an L2 VPN service in pipe mode.....	1267
Figure 302: Use of TTL in an L3 VPN service in pipe mode.....	1267
Figure 303: Tunneling of ICMP reply packets over an MPLS LSP.....	1268
Figure 304: MPLS label stack object.....	1273
Figure 305: ICMP extension header.....	1274
Figure 306: ICMP extension object: object header and payload.....	1274
Figure 307: Example configuration.....	1275
Figure 308: Tunnel from iLER PE-3 to eLER PE-6 via LSR PE-2.....	1279
Figure 309: UDP traceroute in VPRN with iLER in uniform mode.....	1281
Figure 310: UDP traceroute in VPRN without TTL propagation to LDP.....	1283
Figure 311: Tunnel from iLER PE-3 to eLER PE-6 with multiple LSRs.....	1284
Figure 312: UDP traceroute with iLER in uniform mode.....	1285
Figure 313: Example topology for unnumbered interfaces in RSVP and LDP.....	1290
Figure 314: Configuration example for unnumbered Interfaces in RSVP and LDP.....	1297
Figure 315: LSP-PE-4-PE-2 on unnumbered interfaces.....	1301
Figure 316: LSP and FRR facility bypass tunnels.....	1308
Figure 317: FRR one-to-one only supported on numbered interfaces.....	1311
Figure 318: FRR on iLER: no bypass on unnumbered interfaces.....	1313
Figure 319: FRR facility and admin groups.....	1316

Figure 320: SRLG-FRR strict: no bypass on PE-4.....	1319
Figure 321: LDP FRR LFA link protection on PE-4.....	1333
Figure 322: BGP-LU IPv4 route with prefix SID BGP path attribute.....	1338
Figure 323: BGP signaling overview.....	1338
Figure 324: Example topology.....	1340
Figure 325: Example topology with VPRN 1.....	1347
Figure 326: SR TE policy NLRI.....	1351
Figure 327: Binding SID (BSID) anchor.....	1352
Figure 328: Example topology.....	1353
Figure 329: Inter-AS VPRN Model C using MPLS forwarding policy and SR policies.....	1375
Figure 330: Example topology.....	1376
Figure 331: Inter-AS VPRN using MPLS forwarding policy and SR policies: Traffic to PE-7.....	1381
Figure 332: Inter-AS VPRN using MPLS forwarding policy and SR policies: Traffic to PE-1.....	1384
Figure 333: Parallel and non-parallel adjacency sets.....	1395
Figure 334: Parallel adjacency set.....	1395
Figure 335: MPLS label stack.....	1407
Figure 336: Classical BFD handshake.....	1412
Figure 337: Relationship between S-BFD terms.....	1414
Figure 338: S-BFD session establishment - continuity check.....	1415
Figure 339: Example topology.....	1416
Figure 340: Failure on remote link in primary path.....	1423
Figure 341: Segment routing network schematic.....	1435
Figure 342: Node and adjacency SIDs.....	1439

Figure 343: PCC computed strict path between PCC-1 and PCC-2.....	1440
Figure 344: PCC computed LSP hop-to-label translation.....	1442
Figure 345: SR-TE LSP with loose path.....	1443
Figure 346: VPRN service schematic.....	1446
Figure 347: Epipe service schematic.....	1449
Figure 348: SRv6 SID encoding.....	1454
Figure 349: SRv6 SID encoding example.....	1456
Figure 350: End SID for PE-1.....	1457
Figure 351: End.X SID for PE-1.....	1457
Figure 352: IPv6 header defined in RFC 8200.....	1458
Figure 353: Position of the SRH in the protocol stack.....	1458
Figure 354: SRH defined in RFC 8754.....	1458
Figure 355: SRv6 node types.....	1459
Figure 356: Data forwarding of SRv6 encapsulated packets using SRv6 SIDs.....	1460
Figure 357: USP mode - egress router PE-1 processes and removes SRH.....	1462
Figure 358: Penultimate SRH hop P-2 processes and removes the SRH.....	1463
Figure 359: Example topology.....	1464
Figure 360: SRv6 router locator prefixes.....	1466
Figure 361: SRv6 End SID on PE-1.....	1469
Figure 362: SRv6 End.X SIDs on PE-2.....	1472
Figure 363: Example topology.....	1482
Figure 364: Example topology.....	1506
Figure 365: RLFA traffic path during protection.....	1524

Figure 366: Example topology.....	1535
Figure 367: Empty path from PE-2 to PE-11.....	1540
Figure 368: Path from PE-2 to PE-11 via strict hops P-4 and P-3.....	1544
Figure 369: Path from PE-2 to PE-11 via loose hops P-3 and P-9.....	1549
Figure 370: Loose path from PE-2 to PE-11 including unprotected link.....	1560
Figure 371: Loose path from PE-2 to PE-11 including only protected links.....	1562
Figure 372: Example topology.....	1565
Figure 373: Example topology.....	1598
Figure 374: Example topology with metric 21 between PE-2 and P-5.....	1613
Figure 375: Example topology with metric 21 between P-3 and P-5.....	1624
Figure 376: Numbering scheme Airframe with hyperthreading enabled.....	1641
Figure 377: Numbering scheme Dell server without hyperthreading.....	1647
Figure 378: Numbering scheme Dell server with hyperthreading.....	1649
Figure 379: Numbering scheme HPE server without hyperthreading.....	1660
Figure 380: Numbering scheme HPE server with hyperthreading.....	1662

Preface

About This Guide

The Advanced Configuration Guide is divided into three volumes, the Part I Guide, the Part II Guide, and the Part III Guide.

- Part I provides advanced configurations for basic systems, system management, interface configuration, router configuration, unicast routing protocols, MPLS, OAM and diagnostics, and VSR Installation and Setup.
- Part II provides advanced configurations for services overview, Layer 2 and EVPN services, Layer 3 services, and Quality of Service.
- Part III provides advanced configurations for Multi-Service Integrated Service Adapter (MS-ISA) – Extended Services Appliance (ESA), and Triple Play Service Delivery Architecture (TPSDA).

The MD-CLI Advanced Configuration Guide is divided into two volumes, the Part I Guide and the Part II Guide.

- Part I provides advanced configurations for basic systems, system management, interface configuration, router configuration, unicast routing protocols, MPLS, OAM and diagnostics, and VSR Installation and Setup.
- Part II provides advanced configurations for services overview, Layer 2 and EVPN services, Layer 3 services, Multi-Service Integrated Service Adapter (MS-ISA) – Extended Services Appliance (ESA), and Triple Play Service Delivery Architecture (TPSDA).

The guide is organized alphabetically within each category and provides feature and configuration explanations, CLI descriptions and overall solutions. The chapters in the Advanced Configuration Guide are written for and based on several Releases, up to 23.10.R2. The Applicability section in each chapter specifies on which release the configuration is based.

The Advanced Configuration Guide supplements the user configuration guides listed in the 7450 ESS, 7750 SR, and 7950 XRS Guide to Documentation.

Audience

This manual is intended for network administrators who are responsible for configuring the routers. It is assumed that the network administrators have a detailed understanding of networking principles and configurations.

System Management

This section provides configuration information for the following topics:

- [Event Handling System](#)
- [SR OS NETCONF Server Basics](#)

Event Handling System

This chapter provides information about event handling systems (EHS).

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 13.0.R3. The MD-CLI in the current edition is based on SR OS Release 23.7.R2.

SR OS Release 13.0.R1 introduced event handling system (EHS).

SR OS Release 14.0.R4 introduced EHS script enhanced capabilities, such as static variables, advanced syntax (shell scripting commands), and so on. The examples in this chapter do not include these enhancements,

Overview

The event handling system (EHS) in SR OS allows operators to configure user-defined actions defined in CLI scripts that the router executes in response to an event. The event is referred to as the trigger, where the trigger can be all or part of any event message generated by the event-control framework. The user-defined action is controlled by the script-control function. This script-control function references one or more scripts that are able to execute any command available in CLI when the trigger event occurs.

This feature allows for customized automated event management based on specific operator requirements.

Configuration

The topology shown in [Figure 1: Example topology](#) provides an example of an EHS configuration. All routers within the example topology participate in the same IS-IS level-2 area and run LDP. All routers are BGP speakers and form part of autonomous system 64496, exchanging routes for IPv4 address family only.

both upstream and downstream traffic transit through the same PE router, simplifies troubleshooting, QoS configuration, and reconciliation of ingress/egress statistics.

In normal operation, PE-2 is the VRRP master and advertises the BGP prefix 172.16.1.0/29 with a local preference of 100 (default value). Similarly, PE-3 is the VRRP backup and advertises the BGP prefix 172.16.1.0/29 with a local preference of 50, using the BGP export policy "redundant-interface":

```
# on PE-3:
configure {
  policy-options {
    prefix-list "172.16.1.0/29" {
      prefix 172.16.1.0/29 type exact {
      }
    }
  }
  policy-statement "redundant-interface" {
    entry 10 {
      from {
        prefix-list ["172.16.1.0/29"]
      }
      to {
        protocol {
          name [bgp]
        }
      }
      action {
        action-type accept
        local-preference 50
        origin igp
      }
    }
  }
}
```

Therefore, upstream and downstream traffic normally transit through PE-2. The following shows that the VRRP instance on "redundant-interface" on PE-3 is backup.

```
[/]
A:admin@PE-3# show router vrrp instance

=====
VRRP Instances
=====
Interface Name          VR Id  Own  Adm  State      Base Pri  Msg Int
                        IP     Opr  Opr  Pol Id     InUse Pri  Inh Int
-----
redundant-interface    1      No   Up   Backup     253      1
                        IPv4   Up   n/a   253        253      No
  Backup Addr: 172.16.1.1
-----
Instances : 1
=====
```

When PE-3 is backup, it advertises the prefix 172.16.1.0/29 with a local preference of 50, as follows:

```
[/]
A:admin@PE-3# show router bgp routes 172.16.1.0/29 hunt | match 'Network|Nexthop|To|Local Pref'
Network      : 172.16.1.0/29
Nexthop      : 192.0.2.2
Res. Nexthop : 192.168.23.1
Local Pref.  : 100
Network      : 172.16.1.0/29
Nexthop      : 192.0.2.3
Interface Name : int-PE-3-PE-2
```

```
To           : 192.0.2.6
Res. Nexthop : n/a
Local Pref.  : 50
Interface Name : NotAvailable
```

When PE-3 transitions from backup to primary, it must modify its local preference attribute for prefix 172.16.1.0/29 to a value of 150 to attract downstream traffic destined toward CE-1. Similarly, when PE-3 reverts to backup, it must advertise the prefix with a local preference of 50.

Script control

The first step in configuring event handling is to configure a script containing the CLI commands to be executed when the event is triggered. This script can be stored locally on the compact flash, or it can be stored off-node at a defined remote URL, where it can be accessed using FTP or TFTP. When the script is stored locally on the compact flash and the router is equipped with redundant CPMs, the script must be manually saved on the same compact flash on both CPMs, because it is not synchronized automatically.

The first requirement is to modify the local preference of the prefix 172.16.1.0/29 to 150 on transition to VRRP master. The script, which in this example is held locally on CF3:/, therefore contains the following commands (where the policy-statement, redundant-interface, is the name of the export policy used to advertise the 172.16.1.0/29 prefix):

```
[/]
A:admin@PE-3# file show cf3:/vrrp-master.txt
File: vrrp-master.txt
-----
exit all
configure global
policy-options {
policy-statement "redundant-interface" {
entry 10 {
action {
action-type accept
local-preference 150
commit
exit all
}
}
}
=====
```

There is no syntax checking when the script file is created; instead, the script will fail with a command error. Also, transactional CLI (for example the **edit** command) cannot be used in the script, and will fail with a command error.

Within the **system script-control** context, the script is assigned a name and reference is made to its location. It is then configured with **admin-state enable**. When the script has been defined, a **script-policy** is configured that calls the previously configured script. The script-policy also specifies a location and filename for a results file that records the successful or unsuccessful conclusion of each script run and each command executed during that run. Each time the script is run, the results are recorded in a file with the name specified for results, followed by an underscore and the date and time when the script was run. A results file must be specified in order for the script to successfully run. The results file can be on the local compact flash, or a remote URL can be specified. As with the script, the script-policy must also be administratively enabled.

```
# on PE-3:
configure {
system {
script-control {
```

```

script "vrrp-master-script" owner "TiMOS CLI" {
    admin-state enable
    location "cf3:/vrrp-master.txt"
}
script-policy "vrrp-master-policy" owner "TiMOS CLI" {
    admin-state enable
    max-completed 4
    results "cf3:/script-results.txt"
    lifetime forever
    script {
        name "vrrp-master-script"
    }
}

```

The optional **lifetime** command specifies the maximum time that the script may run. The **max-completed** command specifies the maximum number of script run history status entries to be retained. An optional **expire-time** command specifies the maximum time that the system keeps the run history status (default is 1 h). The system maintains the script run history table, which has a maximum size of 255 entries. Entries are removed from this table when the max-completed or expire-time thresholds are crossed. If the table reaches the maximum value, subsequent script launch requests are not run until older run history entries expire (due to expire-time), or entries are manually cleared. To manually clear entries, the following command is used:

```
clear system script-control script-policy completed <script-policy-name>
```

The script run history status information can be viewed using the following command (in this case, after one successful run of the corresponding script) :

```

[/]
A:admin@PE-3# show system script-control script-policy "vrrp-master-policy"

=====
Script-policy Information
=====
Script-policy           : vrrp-master-policy
Script-policy Owner     : TiMOS CLI
Administrative status   : enabled
Operational status     : enabled
Script                  : vrrp-master-script
Script owner            : TiMOS CLI
Python script           : N/A
Source location         : cf3:/vrrp-master.txt
Results location        : cf3:/script-results.txt
Max running allowed    : 1
Max completed run histories : 4
Max lifetime allowed   : 248d 13:13:56 (21474836 seconds)
Completed run histories : 1
Executing run histories : 0
Initializing run histories : 0
Max time run history saved : 0d 01:00:00 (3600 seconds)
Script start error      : N/A
Python script start error : N/A
Last change             : 2023/09/14 07:36:06 N/A
Max row expire time     : never
Last application        : event-script
Last auth. user account : not-specified

=====
Script Run History Status Information
-----
Script Run #1

```

```
-----
Start time   : 2023/09/14 09:38:29 CEST
End time     : 2023/09/14 09:38:29 CEST
Elapsed time : 0d 00:00:00      Lifetime      : 0d 00:00:00
State        : terminated      Run exit code : noError
Result time  : 2023/09/14 09:38:29 CEST
Keep history : 0d 00:58:50
Error time   : never
Source file  : cf3:/vrrp-master.txt
Results file : cf3:/script-results.txt_20230914-073829-UTC.306729.out
Run exit     : Success
Error       : N/A
Application  : event-script      Auth. user ac*: not-specified
* indicates that the corresponding row element may have been truncated.
=====
```

Event handler

The second step in configuring event handling is to assign actions to be performed as a result of the trigger event. These actions are typically one or more configured scripts defined as entries in an action list. In the following output, the event handler is assigned the name `event-handler-1`, and the action list consists of a single entry. This entry calls the previously configured script policy `vrrp-master-policy` (which in turn references the previously defined script `vrrp-master-script`). If multiple actions are required based on a single event trigger, they can be configured in the action list with subsequent entries, which are run in sequence (up to 1500 action list entries are supported).

For this example, only a single entry is required; therefore, there is a one to one relationship between the event handler and the action list entry. Both the entry within the action list and the handler should be configured with `admin-state enable`.

```
# on PE-3:
configure {
  log {
    event-handling {
      handler "event-handler-1" {
        admin-state enable
        entry 10 {
          script-policy {
            name "vrrp-master-policy"
          }
        }
      }
    }
  }
}
```

Event trigger

The final step in configuring event handling is to configure the event trigger. The event trigger defines the event that triggers the running of the script. The event trigger is based on any event generated by the event-control framework, and can match against the application and event number (`event_id`). Log filters can also be used to match against specific events using the subject and/or message fields. Regular expressions can be used where required. EHS will not use any message that is suppressed through event-control configuration, or any event message that is throttled.

The general format for an event in an event log is as follows:

```
nnnn YYYY/MM/DD HH:MM:SS.SS Zone <severity>:<application> #
```

```
<event_id> <router-name> <subject> description
```

Where:

nnnn	The log entry sequence number
YYYY/MM/DD	The UTC date stamp for the log entry: YYYY - Year MM - Month DD - Date
HH:MM:SS.SS	The UTC time stamp for the event HH - Hours (24 hour format) MM - Minutes SS.SS - Seconds
TZONE	The timezone
<severity>	The severity level name of the event
<application>	The application generating the log message
<event_id>	The application's event ID number for the event
<subject>	The subject/affected object for the event
<message>	A textual description of the event

In the example, the following event message is generated when PE-3 becomes VRRP primary:

```
160 2023/09/14 09:37:25.606 CEST MINOR: VRRP #2001 Base Becoming Master
"VRRP virtual router instance 1 on interface redundant-interface
(primary address 172.16.1.3) changed state to master"
```

Therefore, the event-trigger configuration is based on an application of VRRP and an event `vrpTrapNewMaster` (with event number 2001). In the following snippet, `vrp event vrrpTrapNewMaster` is configured. The trigger entry is defined as 1, and in this example, there is only one trigger event. Up to 1500 trigger entries can be included, each of which can act as a potential trigger event. The trigger entry also references the previously configured `event-handler-1`. (Recall that the event handler references the script control, which in turn references the script that should be run.)

```
# on PE-3:
configure {
  log {
    event-trigger {
      vrrp event vrrpTrapNewMaster {
        admin-state enable
        entry 1 {
          filter "itf 172.31.1.3 becomes primary"
          handler "event-handler-1"
        }
      }
    }
  }
}
```

Finally, there is a reference to log-filter `"itf 172.31.1.3 becomes primary"`. Without more explicit filtering, event handling will be triggered on any event with the application of VRRP and event number 2001. There may be multiple VRRP instances running on this router, but the requirement is that event handling should only be triggered when the VRRP instance running on `redundant-interface` transitions to master at PE-3. Therefore, log filter `"itf 172.31.1.3 becomes primary"` is used to define a more explicit match using the message field, which contains an explicit reference to the interface. Both the trigger entry and the event handler must be administratively enabled.

```
configure {
  log {
    filter "itf 172.31.1.3 becomes primary"
    default-action drop
    named-entry "newPrimary" {
```

```

        action forward
        match {
            message {
                eq "interface redundant-interface
                    (primary address 172.16.1.3) changed state to master"
            }
        }
    }
}

```

The configuration of the example event handling for the failure event (PE-3 transitions to VRRP primary) is now complete. By disabling the spoke-SDP between PE-1 and PE-2, it is possible to simulate a failure event where the VRRP message path is broken. Therefore, five events are generated.

- The first indicates that PE-3 has become VRRP master for the interface named redundant-interface.
- The second indicates that EHS handler event-handler-1 was invoked by a CLI user.
- The third indicates that a script file has initiated an attempt to execute CLI commands contained in script file vrrp-master.txt.
- The fourth indicates that a commit by Cron/EHS was successful.
- The fifth indicates that the attempt to execute those CLI commands was successful.

```

164 2023/09/14 09:38:29.306 CEST MINOR: VRRP #2001 Base Becoming Master
"VRRP virtual router instance 1 on interface redundant-interface (primary address 172.16.1.3)
changed state to master"

165 2023/09/14 09:38:29.306 CEST MINOR: SYSTEM #2069 Base EHS script
"Ehs handler : "event-handler-1" with the description : "" was invoked by the cli-user account
"not-specified"."

166 2023/09/14 09:38:29.309 CEST MAJOR: SYSTEM #2052 Base CLI 'exec'
"A CLI user has initiated an 'exec' operation to process the commands in the SROS CLI file
cf3:/vrrp-master.txt"

167 2023/09/14 09:38:29.314 CEST WARNING: SYSTEM #2121 Base Commit
"Commit to configure by (Cron/EHS) from Cron/EHS succeeded."

168 2023/09/14 09:38:29.315 CEST MAJOR: SYSTEM #2053 Base CLI 'exec'
"The CLI user initiated 'exec' operation to process the commands in the SROS CLI file
cf3:/vrrp-master.txt has completed with the result of success"

```

A successful script run shows the commands contained in the script, followed by an indication that the commands were executed.

```

[/]
A:admin@PE-3# file show cf3:/script-results.txt_20230914-073829-UTC.306729.out
File: script-results.txt_20230914-073829-UTC.306729.out
-----
exit all
configure global
INFO: CLI #2054: Entering global configuration mode
policy-options {
policy-statement "redundant-interface" {
entry 10 {
action {
action-type accept
local-preference 150
commit
exit all
INFO: CLI #2056: Exiting global configuration mode
Executed 10 lines in 0.0 seconds from file "cf3:/vrrp-master.txt"

```


The following output confirms that PE-3 is VRRP primary:

```
[/]
A:admin@PE-3# show router vrrp instance

=====
VRRP Instances
=====
Interface Name          VR Id Own Adm State      Base Pri  Msg Int
                       IP      Opr Pol Id   InUse Pri  Inh Int
-----
redundant-interface    1     No  Up  Master    253      1
                       IPv4    Up  n/a     253      No
Backup Addr: 172.16.1.1
-----
Instances : 1
=====
```

Also, the local preference attribute for prefix 172.16.1.0/29 has changed to a value of 150. The result of this action is that PE-3 will now be the transit router for both upstream and downstream traffic.

```
[/]
A:admin@PE-3# show router bgp routes 172.16.1.0/29 hunt | match 'Network|Nexthop|To|Local Pref'
Network      : 172.16.1.0/29
Nexthop      : 192.0.2.3
Res. Nexthop : Unresolved
Local Pref. : 150                               Interface Name : NotAvailable
---snip---
Network      : 172.16.1.0/29
Nexthop      : 192.0.2.3
To           : 192.0.2.6
Res. Nexthop : n/a
Local Pref. : 150                               Interface Name : NotAvailable
```

The event handler indicates that the referenced script was triggered and run using the command shown in the following output. The Handler Action-List Entry Execution Statistics window provides statistics on the number of times an action (script) was queued to run, and the number of times an error was experienced, both during launch and due to a non-operational admin status. The remainder of the fields in the output are self-explanatory.

```
[/]
A:admin@PE-3# show log event-handling handler "event-handler-1"

=====
Event Handling System - Handlers
=====

Handler      : event-handler-1
=====
Description  : (Not Specified)
Admin State  : up                               Oper State : up
-----

Handler Execution Statistics
Success      : 1
Err No Entry : 0
Err Adm Status : 0
```

```

Total          : 1
-----
-----
Handler Action-List Entry
-----
Entry-id       : 10
Description    : (Not Specified)
Admin State    : up                Oper State : up
Script
  Policy Name  : vrrp-master-policy
  Policy Owner : TiMOS CLI
Min Delay      : 0
Last Exec      : 09/14/23 09:38:29 CEST
-----
Handler Action-List Entry Execution Statistics
Success        : 1
Err Min Delay  : 0
Err Launch     : 0
Err Adm Status : 0
Total          : 1
=====

```

The example includes an event trigger and script to meet the requirements of a fail-forward where PE-3 becomes VRRP primary. Now, configuration is needed for when PE-3 reverts to VRRP backup. Without another event trigger and script, PE-3 will continue to advertise the prefix 172.16.1.0/29 with a local preference of 150 and upstream/downstream traffic will be asymmetric through PE-2/PE-3 respectively.

As before, a script is required. Because PE-2 advertises the prefix with a local preference of 100 (default), PE-3 needs to advertise the same prefix with a lower value (50 in the following output), so that PE-2 is the preferred next hop.

```

[/]
A:admin@PE-3# file show cf3:vrrp-backup.txt
File: vrrp-backup.txt
-----
exit all
configure global
policy-options {
policy-statement "redundant-interface" {
  entry 10 {
    action {
      action-type accept
      local-preference 50
    }
  }
}
commit
exit all
=====

```

The script must then be configured within the script-control context, and subsequently referenced in a script policy as vrrp-backup-policy.

```

# on PE-3:
configure {
  system {
    script-control {
      script "vrrp-backup-script" owner "TiMOS CLI" {
        admin-state enable
        location "cf3:/vrrp-backup.txt"
      }
      script-policy "vrrp-backup-policy" owner "TiMOS CLI" {

```

```

        admin-state enable
        max-completed 4
        results "cf3:/script-revert-results.txt"
        lifetime forever
        script {
            name "vrrp-backup-script"
        }
    }

```

The event handler acts as the interface between the configured script policy and event trigger. Therefore, a second event handler is configured with an entry referencing the newly configured vrrp-backup-policy.

```

# on PE-3:
configure {
    log {
        event-handling {
            handler "event-handler-2" {
                admin-state enable
                entry 10 {
                    script-policy {
                        name "vrrp-backup-policy"
                    }
                }
            }
        }
    }
}

```

Finally, the event trigger is configured. To revert to VRRP Backup, the application is VRRP and the event is `tmnxVrrpBecameBackup` (event number 2006). The configuration is filtered on the message field, as before, using log filter `"itf 172.16.1.3 state becomes backup"`, so that it is specific to the interface named `redundant-interface`.

```

# on PE-3:
configure {
    log {
        filter "itf 172.16.1.3 state becomes backup"
        default-action drop
        named-entry "becameBackup" {
            action forward
            match {
                message {
                    eq "interface redundant-interface changed state to backup"
                }
            }
        }
    }
}

```

```

# on PE-3:
configure {
    log {
        event-trigger {
            vrrp event tmnxVrrpBecameBackup {
                admin-state enable
                entry 1 {
                    filter "itf 172.16.1.3 state becomes backup"
                    handler "event-handler-2"
                }
            }
        }
    }
}

```

The configuration of the example event handling for the revertive failure event (PE-3 transitions to VRRP backup) is now complete. By re-enabling the spoke-SDP between PE-1 and PE-2, the VRRP message path is restored, and PE-2 again becomes the VRRP master. The following events are generated:

- The first indicates that PE-3 has become VRRP backup for the interface named redundant-interface.
- The second indicates that EHS handler event-handler-2 was invoked by a CLI user.
- The third indicates that a script file has initiated an attempt to execute CLI commands contained in script file vrrp-backup.txt.
- The fourth indicates that a commit by Cron/EHS was successful.
- The fifth indicates that the attempt to execute those CLI commands was successful.

```
175 2023/09/14 09:46:07.110 CEST MINOR: VRRP #2006 Base Becoming Backup
"VRRP virtual router instance 1 on interface redundant-interface changed state to backup
- current master is 172.16.1.2"

176 2023/09/14 09:46:07.110 CEST MINOR: SYSTEM #2069 Base EHS script
"Ehs handler : "event-handler-2" with the description : "" was invoked by the cli-user
account "not-specified"."
```

```
177 2023/09/14 09:46:07.113 CEST MAJOR: SYSTEM #2052 Base CLI 'exec'
"A CLI user has initiated an 'exec' operation to process the commands in the SROS CLI file
cf3:/vrrp-backup.txt"
```

```
178 2023/09/14 09:46:07.118 CEST WARNING: SYSTEM #2121 Base Commit
"Commit to configure by (Cron/EHS) from Cron/EHS succeeded."
```

```
179 2023/09/14 09:46:07.118 CEST MAJOR: SYSTEM #2053 Base CLI 'exec'
"The CLI user initiated 'exec' operation to process the commands in the SROS CLI file
cf3:/vrrp-backup.txt has completed with the result of success"
```

The following outputs confirm that PE-3 is VRRP backup, and that the local preference attribute for prefix 172.16.1.0/29 has changed to a value of 50. The result of this action is that PE-2 will now be the transit router for both upstream and downstream traffic.

```
[/]
A:admin@PE-3# show router vrrp instance

=====
VRRP Instances
=====
Interface Name          VR Id Own Adm State      Base Pri  Msg Int
                       IP      Opr Pol Id    InUse Pri  Inh Int
-----
redundant-interface    1      No  Up  Backup    253      1
                       IPv4    Up  n/a      253      No
  Backup Addr: 172.16.1.1
-----
Instances : 1
=====
```

```
[/]
A:admin@PE-3# show router bgp routes 172.16.1.0/29 hunt | match 'Network|Nexthop|To|Local Pref'
```

Network	: 172.16.1.0/29	
Nexthop	: 192.0.2.2	
Res. Nexthop	: 192.168.23.1	
Local Pref.	: 100	Interface Name : int-PE-3-PE-2
Network	: 172.16.1.0/29	
Nexthop	: 192.0.2.3	
To	: 192.0.2.6	
Res. Nexthop	: n/a	
Local Pref.	: 50	Interface Name : NotAvailable

Conclusion

EHS allows operators to configure user-defined actions on the router when an event occurs. The event trigger can be anything that is generated by the event-control framework, and explicit filtering is possible using regular expressions. A user-defined action typically runs a script that allows any CLI commands to be executed. Multiple actions are permitted, running multiple scripts if required.

SR OS NETCONF Server Basics

This chapter provides information about SR OS NETCONF server basics.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

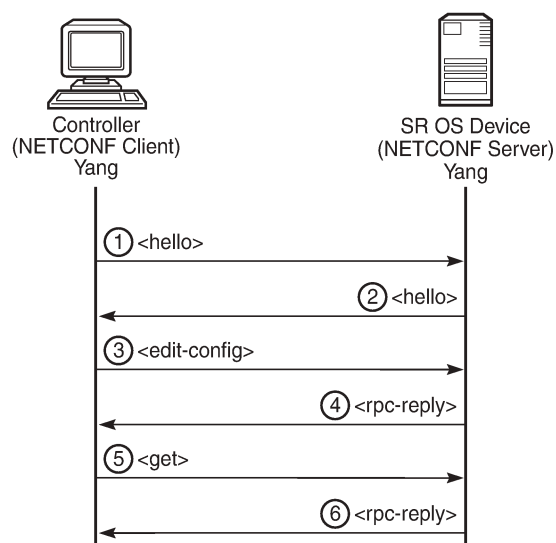
Applicability

This chapter was initially written for SR OS Release 16.0.R4, but the MD-CLI in the current edition corresponds to SR OS Release 21.5.R2.

Overview

The SR OS Network Configuration Protocol (NETCONF) server can communicate with a NETCONF client, that is, exchange hello messages, receive requests, and reply with responses. Before communicating with the SR OS NETCONF server, some SR OS configurations are prerequisites, and others are optional. This chapter describes the basic configurations needed for a seamless interaction with the SR OS NETCONF server. [Figure 2: NETCONF client-server communication](#) shows the NETCONF client-server communication between the controller and the SR OS node.

Figure 2: NETCONF client-server communication



28626

Configuration

The following steps describe the procedure to configure a NETCONF server on SR OS.

- Because NETCONF uses SSH for transport, enable the SSH server in SR OS:

```
configure {
  system {
    security {
      ssh {
        server-admin-state enable
      }
    }
  }
}
```

- Enable the NETCONF server:

```
configure {
  system {
    management-interface {
      netconf {
        admin-state enable
      }
    }
  }
}
```

- Enable the YANG modules to use with NETCONF; for example, the Nokia combined modules:

```
configure {
  system {
    management-interface {
      yang-modules {
        nokia-submodules false
        nokia-combined-modules true
      }
    }
  }
}
```



Note:

The Nokia combined modules and the Nokia submodules are mutually exclusive and cannot both be set to true at the same time.

- Configure an "nc_user" user with administrative privileges (**access netconf**):

```
configure {
  system {
    security {
      user-params {
        local-user {
          user "nc-user" {
            password <password>
            access {
              console true
              netconf true
            }
            console {
              member ["administrative"]
            }
          }
        }
      }
    }
  }
}
```

- Optionally, enable NETCONF auto-config-save, which auto-saves the data (that is, makes it persistent) after each successful NETCONF commit:

```
configure {
  system {
    management-interface {
      netconf {
        auto-config-save enable
      }
    }
  }
}
```

```
auto-config-save true
```

- Optionally, grant the NETCONF user permission to **lock** a datastore through NETCONF:

```
configure {
  system {
    security {
      aaa {
        local-profiles {
          profile "administrative" {
            netconf {
              base-op-authorization {
                lock true
              }
            }
          }
        }
      }
    }
  }
}
```

- Optionally, grant the NETCONF user permission to kill an open NETCONF session:

```
configure {
  system {
    security {
      aaa {
        local-profiles {
          profile "administrative" {
            netconf {
              base-op-authorization {
                kill-session true
              }
            }
          }
        }
      }
    }
  }
}
```

- Commit the configuration:

```
commit
```

- If MD-CLI auto-config-save is disabled, save the configuration manually:

```
admin save
```

Conclusion

This chapter describes general SR OS NETCONF server configurations.

Interface Configuration

This section provides interface configuration information for the following topics:

- [Port Cross-Connect \(PXC\)](#)

Port Cross-Connect (PXC)

This chapter provides information about Port Cross-Connect (PXC).

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The chapter was initially written for SR OS Release 14.0.R5, but the MD-CLI in the current edition is based on SR OS Release 21.5.R1.

Overview

The Port Cross-Connect (PXC) feature allows for a port, or number of ports, to be logically looped to themselves. The purpose of looping a port in this manner is to provide an "anchor point" function, such that traffic may ingress the node through any interface/port and be redirected to that anchor point.

When traffic is passed through the egress data path of the PXC, it can be used for additional packet processing that cannot be supported on the ingress data path, such as the removal of an encapsulation header. When traffic is looped back to the ingress data path of the PXC, it is processed as if it were the conventional service termination point. This essentially decouples the Input/Output (I/O) port through which packets ingress the node from the I/O port that implements the service termination. This decoupling removes the previous constraint for pseudowire-port (pw-port) whereby the I/O port through which packets ingress and egress the node was bound and could not be changed during, for example, a reconvergence event.

PXC provides two modes of operation: Distributed Versatile Service Module (DVSM) mode and Application Specific (AS) mode.

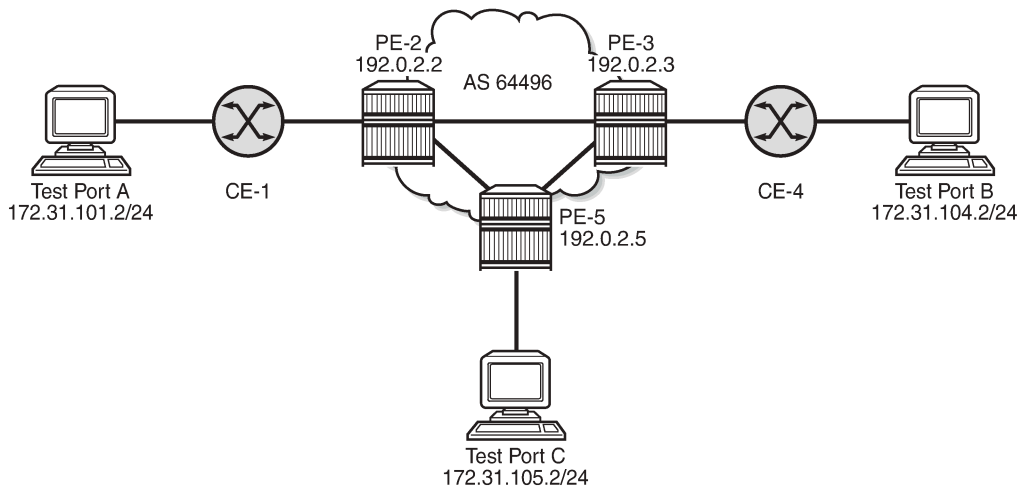
- The DVSM mode provides functionality like that of the VSM2 card, enabling the user to create an internal loopback through the card. This allows for back-to-back configurations similar to a VLAN cross-connect.
- The AS mode creates a Forwarding Path Extension (FPE) context through which the system can automatically create cross-connects to simplify user provisioning. Use-case examples for AS mode include PW port for business VPN services, VXLAN termination on a non-system interface, ESM over Pseudowire, and GRE tunnel termination.

This chapter describes the generic principles of PXC, combined with examples of both DVSM mode and AS mode.

Example topology

The topology shown in [Figure 3: Example topology](#) is used within this chapter to illustrate the use of PXC. PE-2, PE-3, and PE-5 form part of Autonomous System 64496 and run IS-IS level 2 together with LDP for the MPLS control plane. PE-2, PE-3, and PE-5 also peer in IBGP for the VPN-IPv4 address family. Test ports are connected to all PEs (in the case of PE-2 and PE-3, via CE routers) for the purpose of validating IP connectivity.

Figure 3: Example topology



26223

PE-5 will host the PXC.

Configuration

PXC configuration

A PXC can consist of a single non-redundant port, or for redundancy and increased capacity, can consist of multiple ports that form member links of a Link Aggregation Group (LAG). Both options are described here.

Non-redundant PXC

The non-redundant PXC is created within the **port-xc** context and can be numbered from 1 to 64. The PXC sub-ports are identified by *a* and *b* suffixes of the parent PXC (in this example, pxc-1) and need to be enabled. The port assigned to the PXC also needs to be enabled. The configuration of PXC 1 on PE-5 is as follows:

```
# on PE-5:
configure {
  port-xc {
    pxc 1 {
      admin-state enable
      description "PXC non-redundant"
```

```

        port-id 1/2/1
    }
}
port pxc-1.a {
    admin-state enable
}
port pxc-1.b {
    admin-state enable
}
port 1/2/1 {
    admin-state enable
    ethernet {
        mode hybrid
        dot1x {
            tunneling true
        }
    }
}
    
```

There is no requirement for any kind of optical transceiver to be inserted in the port assigned to the PXC; it is only a logical loopback. When the port is assigned to the PXC, it cannot be used for any other purpose besides a PXC-based service assignment (for example, a regular SAP could not be configured on this port).

The PXC sub-ports are created in hybrid mode with an MTU of 8700 bytes, both of which are non-configurable. The 8700-byte MTU represents the default port MTU (in this example, 8704 bytes) minus four bytes to allow for an internal VLAN tag that is used to identify each back-to-back sub-port. Finally, the encapsulation is set to dot1q, which is the default for hybrid ports. The following command shows the details for both PXC sub-ports of PXC 1:

```

[/]
A:admin@PE-5# show port pxc 1

=====
Ports on Port Cross Connect 1
=====
Port      Admin Link Port  Cfg Oper LAG/ Port Port Port  C/QS/S/XFP/
Id        State State State MTU MTU  Bndl Mode Encp Type  MDIMDX
-----
pxc-1.a   Up    Yes  Up    8700 8700  -  hybr dotq xgige
pxc-1.b   Up    Yes  Up    8700 8700  -  hybr dotq xgige
=====
    
```

Q-in-Q encapsulation is also supported. It is also possible to configure dot1q encapsulation on one PXC sub-port and Q-in-Q encapsulation on the opposing PXC sub-port if, for example, there is a requirement to expose more VLAN tags on one side of the loop than the other side of the loop.

The command in the following output can be used to verify the operational state of the PXC:

```

[/]
A:admin@PE-5# show port-xc pxc 1

=====
Port Cross-Connect Information
=====
PXC      Admin   Oper    Port    Description
Id       State  State   Id
-----
1        Up      Up      1/2/1   PXC non-redundant
=====
    
```

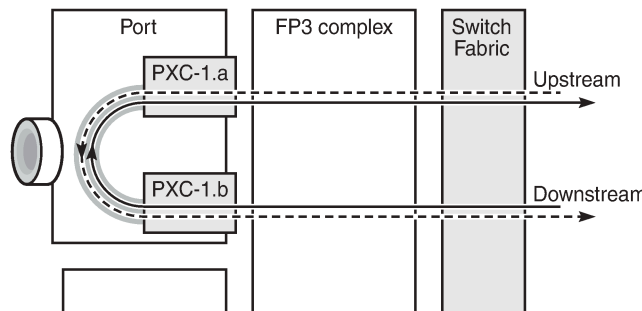
Similarly, the operational state of each of the sub-ports can be verified as follows. The physical link is indicated as being present even though there is no transceiver installed in this port.

```
*A:PE-5# show port pxc-1.a

=====
Ethernet Interface
=====
Description      : Port cross-connect
Interface        : pxc-1.a                Oper Speed      : 10 Gbps
Link-level      : Ethernet                Config Speed    : N/A
Admin State     : up                      Oper Duplex     : full
Oper State      : up
Config Duplex   : N/A
Physical Link : Yes                    MTU             : 8700
Single Fiber Mode : No                    Min Frame Length : 64 Bytes
IfIndex         : 1090523137              Hold time up    : 0 seconds
Last State Change : 05/25/2021 10:24:01  Hold time down  : 0 seconds
Last Cleared Time : N/A
Phys State Chng Cnt: 0
---snip---
```

Figure 4: Non-redundant PXC shows a representation of the non-redundant PXC configuration. Both upstream and downstream traffic will pass twice through the FP data-path and port. For example, downstream traffic passes through the FP complex and PXC-1.b. The traffic is then looped back to PXC-1.a, and back into the FP complex. Similarly, upstream traffic passes through the FP complex to PXC-1.a. It is then looped back to PXC-1.b and back into the FP complex.

Figure 4: Non-redundant PXC



26224

When using a PXC, the physical port effectively simulates two (sub-)ports, which creates two egress traffic paths: one upstream and one downstream. When the receive side of the PXC port receives those paths, it needs to distinguish between them, and this is where the internal additional VLAN tag is used.

The difference between this PXC configuration and a conventional port not looped or configured as PXC is as follows. With a conventional port, ingress traffic passes through the port and ingress data-path of the FP complex only once, and egress traffic passes through the egress data-path of the FP complex and port only once.

Redundant PXC

For a redundant PXC, the fundamental building blocks are identical to those of the non-redundant PXC, but there are a few additional configuration steps required to construct the LAGs to which the redundant PXC ports belong.

The redundant PXC example consists of two ports: 1/2/2 and 1/2/3 in the following output. In this case, the redundant PXC ports belong to the same IMM, but different IMM can be used for increased redundancy. Two PXC are created and each one is assigned one of the redundant PXC ports. Both PXC are enabled, as well as their sub-ports and the associated physical ports. The configuration of PXC 2 and PXC 3 on PE-5 is as follows:

```
# on PE-5:
configure {
  port-xc {
    pxc 2 {
      admin-state enable
      description "PXC redundant"
      port-id 1/2/2
    }
    pxc 3 {
      admin-state enable
      description "PXC redundant"
      port-id 1/2/3
    }
  }
  port pxc-2.a {
    admin-state enable
  }
  port pxc-2.b {
    admin-state enable
  }
  port pxc-3.a {
    admin-state enable
  }
  port pxc-3.b {
    admin-state enable
  }
  port 1/2/2 {
    admin-state enable
    ethernet {
      mode hybrid
      dot1x {
        tunneling true
      }
    }
  }
  port 1/2/3 {
    admin-state enable
    ethernet {
      mode hybrid
      dot1x {
        tunneling true
      }
    }
  }
}
```

The details of the sub-ports of the PXC can be verified as follows:

```
[/]
A:admin@PE-5# show port pxc 2
```

```

=====
Ports on Port Cross Connect 2
=====
Port      Admin Link Port   Cfg  Oper  LAG/  Port Port Port  C/QS/S/XFP/
Id        State      State MTU  MTU  Bndl Mode Encp Type MDIMDX
-----
pxc-2.a   Up    Yes  Up    8700 8700  -  hybr dotq xgige
pxc-2.b   Up    Yes  Up    8700 8700  -  hybr dotq xgige
=====
    
```

```

[/]
A:admin@PE-5# show port pxc 3
    
```

```

=====
Ports on Port Cross Connect 3
=====
Port      Admin Link Port   Cfg  Oper  LAG/  Port Port Port  C/QS/S/XFP/
Id        State      State MTU  MTU  Bndl Mode Encp Type MDIMDX
-----
pxc-3.a   Up    Yes  Up    8700 8700  -  hybr dotq xgige
pxc-3.b   Up    Yes  Up    8700 8700  -  hybr dotq xgige
=====
    
```

The operational state of the PXC's can be verified as follows:

```

[/]
A:admin@PE-5# show port-xc pxc 2
    
```

```

=====
Port Cross-Connect Information
=====
PXC      Admin   Oper    Port      Description
Id       State  State   Id
-----
2        Up     Up      1/2/2     PXC redundant
=====
    
```

```

[/]
A:admin@PE-5# show port-xc pxc 3
    
```

```

=====
Port Cross-Connect Information
=====
PXC      Admin   Oper    Port      Description
Id       State  State   Id
-----
3        Up     Up      1/2/3     PXC redundant
=====
    
```

The PXC sub-ports are then associated with two LAGs to essentially form an internal back-to-back LAG. To do this, both sub-ports with the .a suffix belong to one LAG instance, and both sub-ports with the .b suffix belong to the other LAG instance. Like any other LAG member links, PXC sub-ports in a LAG must be configured with the same physical attributes, such as speed and duplex. Both LAG instances are configured with **mode hybrid** to match the mode of the physical ports. Setting the mode to **hybrid** automatically sets the **encap-type** to **dot1q**.

```

# on PE-5:
configure {
    lag "lag-1" {
        admin-state enable
    }
}
    
```

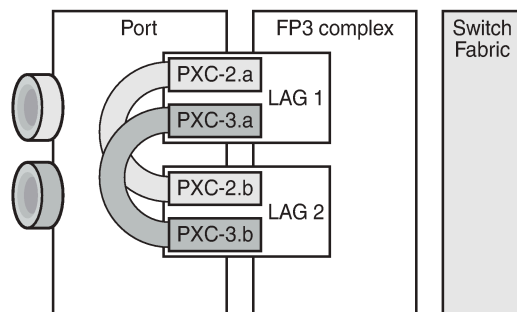
```

mode hybrid
max-ports 64
port pxc-2.a {
}
port pxc-3.a {
}
}
lag "lag-2" {
admin-state enable
mode hybrid
max-ports 64
port pxc-2.b {
}
port pxc-3.b {
}
}
}

```

Figure 5: PXC redundant mode with LAG shows a representation of the redundant PXC with LAG. Both upstream and downstream traffic will pass twice through the FP data-path and port.

Figure 5: PXC redundant mode with LAG



26225

When the LAGs are configured and the associated PXC sub-ports assigned as member links, the operational status can be verified. Note that at the LAG level, each of the configured LAG instances is not aware that it is internally connected to another LAG instance, even though the member sub-ports are logically looped. It would be possible, for example, to configure LAG 1 with **admin-state disable** and not affect the operational state of LAG 2. LACP is not supported for PXC LAG; however, it is possible to run the 802.3ah Ethernet in the First Mile (EFM) at PXC sub-port level, if required.

```

[/]
A:admin@PE-5# show lag "lag-1" detail
=====
LAG Details
=====
Description      : N/A
-----
Details
-----
Lag-id           : 1                Mode           : hybrid
Lag-name         : lag-1
Adm              : up                Opr            : up
Thres. Last Cleared : 05/25/2021 10:00:34  Thres. Exceeded Cnt : 0
Dynamic Cost     : false            Encap Type     : dot1q
Configured Address : 02:1f:ff:00:01:41      Lag-IfIndex    : 1342177281
Hardware Address  : 02:1f:ff:00:01:41      Adapt Qos (access) : distribute

```



```

Hold-time Down      : 0.0 sec          Port Type           : standard
Per-Link-Hash       : disabled
Include-Egr-Hash-Cfg: disabled
Per FP Ing Queuing  : disabled        Per FP Egr Queuing  : disabled
Per FP SAP Instance : disabled
Access Bandwidth    : N/A             Access Booking Factor: 100
Access Available BW : 0
Access Booked BW    : 0
LACP                : disabled
Standby Signaling   : lacp
Port hashing        : port-speed       Port weight speed   : 0 gbps
Ports Up            : 2
Weights Up          : 2                Hash-Weights Up     : 20
Monitor oper group  : N/A
Adaptive loadbal.   : disabled         Tolerance           : N/A

```

Port-id	Adm	Act/Stdby	Opr	Primary	Sub-group	Forced	Prio
pxc-2.a	up	active	up	yes	1	-	32768
pxc-3.a	up	active	up		1	-	32768

DVSM mode

DVSM mode enables the creation of a back-to-back cross-connect. This back-to-back connection can be network-to-network, access-to-access, or a combination such as network-to-access. To provide an example of using DVSM mode, PE-3 in [Figure 3: Example topology](#) functions as a Layer 2 backhaul device, and PE-5 housing the PXC functions as the Layer 3 service edge. A pseudowire is extended from PE-3 to PE-5, where it is terminated in a VPRN, providing point-to-point connectivity between CE-4 and PE-5.

VLAN 100 is extended from CE-4 to PE-3, where it is indexed into an Epipe service. The SAP is service-delimiting; therefore, the VLAN is removed before frames are encapsulated into the pseudowire. The Epipe then has a single non-redundant spoke-SDP to PE-5 with VC-ID 11. The service configuration on PE-3 is as follows:

```

# on PE-3:
configure {
  service {
    epipe "Epipe 11" {
      admin-state enable
      service-id 11
      customer "1"
      spoke-sdp 35:11 {
      }
      sap 1/1/3:100 {
      }
    }
  }
  sdp 35 {
    admin-state enable
    delivery-type mpls
    ldp true
    far-end {
      ip-address 192.0.2.5
    }
  }
}

```

At PE-5, the configuration of the corresponding end of the Epipe service is shown in the following output. This service consists of a single spoke-SDP toward PE-3 with VC-ID 11 to match the VC-ID advertised by PE-3, and a single SAP toward the PXC port. The syntax takes the form "pxc-n.l:vlan" where "n" is the PXC identifier, "l" is the sub-port letter (in this case .a), and "vlan" represents the VLAN identifier of the SAP.

As shown in the following output, the Epipe service uses PXC 1, which is the non-redundant PXC port. This is only an example; it could similarly use the redundant PXC port, in which case the SAP syntax would be the conventional LAG syntax (for example, lag-1:100, lag-2:100). Also note that although VLAN 100 is used both at PE-3's Epipe SAP and PE-5's Epipe PXC SAP, there is no correlation or dependence between the two. Both VLAN tags are service-delimiting and are subsequently stripped before the Ethernet frame is encapsulated into the pseudowire payload, so any valid VLAN value could be used at either point. The service configuration on PE-5 is as follows:

```
# on PE-5:
configure {
  service {
    epipe "Epipe 11" {
      admin-state enable
      service-id 11
      customer "1"
      spoke-sdp 53:11 {
      }
      sap pxc-1.a:100 {
      }
    }
    sdp 53 {
      admin-state enable
      delivery-type mpls
      ldp true
      far-end {
        ip-address 192.0.2.3
      }
    }
  }
}
```

The VPRN configuration at the corresponding side of the PXC port is shown in the following output. The VPRN has two interfaces: the first is toward a directly connected test port used to verify IP connectivity, and the second ("to-CE-4") is toward CE-4 and has a SAP with a PXC syntax. The PXC syntax represents the same PXC and VLAN identifiers as the preceding Epipe configuration, but the PXC sub-port is .b, to represent the "other side" of the PXC logical loopback. Therefore, the VLAN values must match to create the back-to-back connection. A BGP session is configured between PE-5 and CE-4 for route exchange. VPRN 10 is configured as follows:

```
# on PE-5:
configure {
  service {
    vprn "VPRN 10" {
      description "VPRN 10 using PXC DVSM"
      admin-state enable
      service-id 10
      customer "1"
      autonomous-system 64496
      bgp-ipvprn {
        mpls {
          admin-state enable
          route-distinguisher "64496:10"
          vrf-import {
            policy ["vrf10-import"]
          }
          vrf-export {

```

```

        policy ["vrf10-export"]
        }
        auto-bind-tunnel {
            resolution any
        }
    }
}
bgp {
    ebgp-default-reject-policy {
        import false
    }
    group "EBGP" {
        peer-as 64511
        family {
            ipv4 true
        }
    }
    neighbor "192.168.45.1" {
        split-horizon true
        group "EBGP"
        as-override true
        export {
            policy ["vrf-ebgp-export"]
        }
    }
}
interface "Test-Port-C" {
    ipv4 {
        primary {
            address 172.31.105.1
            prefix-length 24
        }
    }
    sap 1/1/3:100 {
    }
}
interface "to-CE-4" {
    ipv4 {
        primary {
            address 192.168.45.2
            prefix-length 30
        }
    }
    sap pxc-1.b:100 {
    }
}

```

PXC port dimensioning

When the VPRN service at PE-5 is in **admin-state enable**, the EBGP session to CE-4 is established. The relevant routes are exchanged between CE-4 and PE-5 and traffic can be exchanged between test ports B (connected to CE-4) and C (connected to PE-5). Initially, traffic is sent from test port B toward port C at a rate of 100 packets/s. Traffic is intentionally sent in only one direction (in this example) to emphasize a point regarding PXC port dimensioning and capacity planning, as follows.

The PXC in use by the Epipe/VPRN service is PXC 1, which uses physical port 1/2/1. The following output—in classic CLI—shows a snapshot of a monitor command against the physical port. Although traffic is only being sent in a single direction (test port B behind CE-4 toward test port C connected to PE-5), the

input/output rate of packets per second is the same at 100 packets/s. This is because the physical port consists of two PXC sub-ports that are looped.

```
*A:PE-5# monitor port 1/2/1 rate interval 3

=====
Monitor statistics for Port 1/2/1
=====
-----
Input                               Output
-----
---snip---
-----
At time t = 3 sec (Mode: Rate)
-----
Octets                               51600                               51600
Packets                              100                                100
Errors                                0                                  0
Bits                                 412800                             412800
Utilization (% of port capacity)    ~0.00                              ~0.00
```

In this example, traffic is output from pxc-1.a when traffic is sent from the Epipe SAP into the PXC port, and traffic is input at pxc-1.b when traffic is received by the VPRN SAP from the PXC port. Because both upstream/ingress traffic and downstream/egress traffic will be seen as output packets using the available capacity of the physical port, this needs to be considered when capacity is being planned.

QoS continuity

The application of ingress/egress SAP QoS policies is fundamentally the same for a PXC-based SAP as it is for a conventional SAP. However, there is a difference with regard to how ingress Forwarding Class (FC) mappings are maintained throughout the PXC in DVSM mode. On a conventional SAP, ingress packets are classified and mapped to an FC. That FC mapping is maintained (as part of the fabric header) when the packet transits through the system and is ultimately used to define the egress queue and egress marking, such as MPLS EXP bits or dot1p bits.

However, the PXC sub-ports are subtly different. Consider SAP ingress traffic entering the VPRN at PE-5 from the locally connected test port C destined toward test port B at CE-4. At the ingress to PE-5, this traffic is mapped to FC Expedited Forwarding (EF) and forwarded into the PXC port through SAP pxc-1.b:100. When the traffic is forwarded out of the (PXC) SAP, the fabric header is removed as if it were a conventional SAP, and therefore, the information conveying the FC mapping is lost. When the traffic arrives at the opposing PXC sub-port SAP (in this case, pxc-1.a:100), a further FC classification is undertaken, and without some non-default configuration, traffic will be classified as FC Best Effort (BE). Therefore, it is a requirement to use non-default ingress/egress QoS policies through the PXC port in order to maintain FC continuity. A relatively simple way to do to this is through the use of dot1p markings.

To illustrate how this FC continuity is achieved, and in general how QoS is applied to PXC ports, an example of the relevant policies applied to PE-5's egress traffic toward CE-4 is used.

The first of the following outputs provides an example of the SAP-egress QoS policy applied at the VPRN PXC SAP (pxc-1.b:100). There are three classes in use: BE, Assured-Forwarding (AF), and EF. These FCs are remapped to queues 1, 2, and 3, respectively, and each queue is mapped to a parent H-QoS scheduler. Because the FCs must be maintained through the PXC loop, dot1p markings are used to distinguish between them. FC EF uses dot1p 5, FC AF uses dot1p 3, and FC BE uses dot1p 1. The SAP egress QoS policy is configured on PE-5 as follows:

```
# on PE-5:
configure {
```

```

qos {
  sap-egress "SAP egress 2" {
    policy-id 2
    queue 1 {
      scheduler-parent {
        scheduler-name "aggregate-rate"
        level 2
        weight 10
      }
    }
    queue 2 {
      queue-type best-effort
      rate {
        pir 5000
        cir max
      }
      scheduler-parent {
        scheduler-name "aggregate-rate"
        level 2
        weight 40
        cir-level 2
      }
    }
    queue 3 {
      queue-type expedited
      rate {
        pir 2000
        cir 2000
      }
      scheduler-parent {
        scheduler-name "aggregate-rate"
        cir-level 3
      }
    }
  }
  fc be {
    queue 1
    dot1p {
      in-profile 1
      out-profile 1
      exceed-profile 1
    }
  }
  fc af {
    queue 2
    dot1p {
      in-profile 3
      out-profile 3
      exceed-profile 3
    }
  }
  fc ef {
    queue 3
    dot1p {
      in-profile 5
      out-profile 5
      exceed-profile 5
    }
  }
}

```

The configuration of the Tier 1 scheduler "aggregate-rate" referenced by the child queues in the preceding SAP-egress QoS policy is shown in the following output. The scheduler in turn references a **port-scheduler-policy** using the command **port-parent**. Parenting to a port-scheduler is optional, but allows

for inclusion of Preamble and Inter-Frame Gap (IFG) in the QoS scheduling algorithm, which is otherwise not included by a conventional H-QoS scheduler. The **port-scheduler-policy** "port-scheduler" is not referenced directly by the Tier 1 scheduler, but rather the port-scheduler is inherited by any child queues on the port to which the port-scheduler is applied. In this case, the **port-scheduler-policy** "port-scheduler" is applied to the PXC sub-port pxc-1.b as follows:

```
# on PE-5:
configure {
  qos {
    scheduler-policy "egress-hqos-scheduler" {
      tier 1 {
        scheduler "aggregate-rate" {
          rate {
            pir 1
          }
          port-parent {
          }
        }
      }
    }
    port-scheduler-policy "port-scheduler" {
    }
  }
  port pxc-1.b {
    admin-state enable
    ethernet {
      egress {
        port-scheduler-policy {
          policy-name "port-scheduler"
        }
      }
    }
  }
}
```

Finally, the SAP-egress QoS policy is applied to the PXC sub-port SAP within the VPRN interface context. The H-QoS scheduler is also attached and an override of the rate configured. In summary, the SAP-egress QoS policy configuration looks exactly like that used on a conventional SAP, other than the dot1p markings used for FC continuity, which may not always be used or required.

```
# on PE-5:
configure {
  service {
    vprn "VPRN 10" {
      interface "to-CE-4" {
        admin-state disable
        ipv4 {
          primary {
            address 192.168.45.2
            prefix-length 30
          }
        }
      }
      sap pxc-1.b:100 {
        admin-state disable
        egress {
          qos {
            sap-egress {
              policy-name "SAP egress 2"
            }
            scheduler-policy {
              policy-name "egress-hqos-scheduler"
              overrides {
                scheduler "aggregate-rate" {

```



```

spoke-sdp 53:11 {
}
sap pxc-1.a:100 {
  ingress {
    qos {
      sap-ingress {
        policy-name "SAP ingress 11"
      }
    }
  }
}

```

The preceding configuration shows the required QoS policies for downstream traffic (VPRN egress to Epipe ingress). Corresponding QoS policies must also be configured for upstream traffic (Epipe egress to VPRN ingress). For brevity, they are not shown here.

AS mode

AS mode creates an FPE context that is used to provide information to the system about which PXC ports or LAGs are paired, so that the configuration process can be simplified by automatic provisioning of cross-connects. To illustrate the use of AS mode, the redundant PXC (formed of LAG 1 and 2) configured earlier in this chapter is used. However, redundancy is not a requirement. Non-redundant PXC ports can also be used with AS mode.

For AS mode, a similar setup to the DVSM example is used, with Epipe termination into a VPRN. This provides a generic view of the applicability of AS mode, but also allows a direct comparison between the DVSM and AS mode approaches. Again, PE-3 in [Figure 3: Example topology](#) functions as a Layer 2 backhaul device and PE-5 hosts the PXC functions as the Layer 3 service edge. A pseudowire is extended from PE-3 to PE-5 where it will be terminated in a VPRN, providing point-to-point connectivity between CE-4 and PE-5.

The following output illustrates the configuration of the Epipe service at PE-3. CE-4 uses Q-in-Q encapsulation on the PE-CE link to PE-3 with SVLAN tag 100 and CVLAN tag 1024. At PE-3, it is indexed into an Epipe service using a q.* SAP to make the CVLAN tag transparent (part of the payload). As the spoke-SDP toward PE-5 is also configured with **force-vc-forwarding vlan**, both SVLAN and CVLAN tags will be encapsulated in the pseudowire payload.

```

# on PE-3:
configure {
  service {
    epipe "Epipe 13" {
      admin-state enable
      service-id 13
      customer "1"
      spoke-sdp 35:13 {
        force-vc-forwarding vlan
      }
      sap 1/1/3:100.* {      # SAP to VPRN 12 on CE-4
      }
    }
  }
}

```

As in the previous configuration example, LAG 1 and LAG 2 are used for PXC redundancy. LAG 1 has the PXC sub-ports pxc-2.a and pxc-3.a as member links, while LAG 2 has the PXC sub-ports pxc-2.b and pxc-3.b as member links. For AS mode, the next requirement is to configure the FPE construct and assign the paired LAG instances to that FPE. When entering the **fwd-path-ext** context, the **sdp-id-range** must be configured before any **fpe** instances can be created. The **sdp-id-range** allocates a block of SDP identifiers

to be used for the automatic cross-connects between service applications and the FPE. Up to 128 SDP identifiers can be allocated in the range 1 to 17407.

After the **sdp-id-range** is configured, the **fpe** instance is created and the user enters the **fpe** context. The **path** command is used to assign redundant or non-redundant PXC objects to the FPE. In the case of a non-redundant FPE, the **path** command would refer to a **pxc** instance. In the case of a redundant FPE, the **path** syntax requires that each of the paired LAG instances is assigned to cross-connect "a" or cross-connect "b".

Each FPE has two fundamental components, known as the transit side and the terminating side. The transit side is the side where additional traffic preprocessing is carried out, such as header removal or manipulation. It can be considered as the side closest to the network. The terminating side is the side where the preprocessed traffic is terminated in a service. When an FPE is used, the system automatically assigns cross-connect "a" to the transit side, and cross-connect "b" to the terminating side.

In the following example, the command **path>xc-lag-a "lag-1"** assigns LAG 1 to cross-connect "a" and the command **path>xc-lag-b "lag-2"** assigns LAG 2 to cross-connect "b". This means that LAG 1 is the transit side while LAG 2 is the terminating side.

The application of the FPE also needs to be configured. In this example, **pw-port true** is configured to allow for support of pseudowire-SAP (including Enhanced Subscriber Management (ESM) over pseudowire). The other available options (for example, **vlan-termination**) are beyond the scope of this chapter.

```
# on PE-5:
configure {
  fwd-path-ext {
    sdp-id-range {
      start 17280
      end 17407
    }
  }
  fpe 1 {
    path {
      xc-lag-a "lag-1"
      xc-lag-b "lag-2"
    }
    application {
      pw-port true
    }
  }
}
```

After the LAG instance is assigned to the FPE, it can no longer be used for other general purposes, such as IP interfaces and/or SAPs. Any attempt to do so is blocked in MD-CLI. The operational state of the FPE can be verified as shown in the following output. It is also useful to be able to identify the services and pw-ports that are mapped to an FPE. This can be obtained using the **show fwd-path-ext fpe <number> associations** command.

```
[/]
A:admin@PE-5# show fwd-path-ext fpe 1

=====
FPE Id: 1
=====
Description      : (Not Specified)
Path             : lag-1, lag-2
Pw Port          : Enabled           Oper    : up
Sub Mgmt Extension : Disabled         Oper    : N/A
Vxlan Termination : Disabled         Oper    : down
Segment-Routing V6 : Disabled
```

The next step is to configure a pseudowire-port (pw-port) that will be used for terminating services. The creation of the **pw-port** creates a new context in which the only required configuration is to define the encapsulation type as dot1q or qinq. In this instance, the **pw-port** will support **encap-type qinq**.

```
# on PE-5:
configure {
  pw-port 1 {
    encap-type qinq
  }
}
```

The operational state of the pw-port is captured as a reference at this point, so that a comparison can be made later in the configuration process.

```
[/]
A:admin@PE-5# show pw-port 1

=====
PW Port Information
=====
PW Port   Encap           SDP:VC-Id       IfIndex
-----
1         qinq           N/A             1526726657
=====
```

At PE-5, the requirement now is to link the spoke-SDP from PE-3 to the configured pw-port (pw-port 1) via the FPE. To do this, an Epipe service must be used that is configured for multi-segment pseudowire working, configured with the option **vc-switching true**. The Epipe service consists of a single spoke-SDP toward PE-3 with a VC-ID matching that signaled by PE-3 (VC-ID 13). The pw-port 1 refers to **fpe-id 1** in "Epipe 13". This command essentially creates an internal cross-connect between the Epipe service and the pw-port via the configured FPE object. The configuration is as follows:

```
# on PE-5:
configure {
  pw-port 1 {
    encap-type qinq
    epipe "Epipe 13" {
      fpe-id 1
    }
  }
  service {
    epipe "Epipe 13" {
      admin-state enable
      service-id 13
      customer "1"
      vc-switching true
      spoke-sdp 53:13 {
      }
    }
  }
}
```

The following output shows the SDPs belonging to the preceding vc-switched Epipe service configured. The first SDP with identifier 53:13 is the pseudowire toward PE-3 with VC-ID 13. The second SDP has identifier 17280:1 allocated from the preconfigured **sdp-id-range**, and has a type of Fpe. In the configuration of **fpe 1**, the **path** command assigned LAG 1 to cross-connect "a" (**xc-lag-a**) and LAG 2 to cross-connect "b" (**xc-lag-b**). Also, cross-connect "b" is always automatically assigned to the terminate side of the FPE. Therefore, the Far End address is shown as fpe_1.b, in order to terminate the service.

```
[/]
A:admin@PE-5# show service id 13 sdp
```

```

=====
Services: Service Destination Points
=====
SdpId          Type      Far End addr  Adm   Opr      I.Lbl  E.Lbl
-----
53:13          Spok     192.0.2.3    Up    Up        524279 524283
17280:1        Fpe      fpe_1.b      Up    Up        524281 524280
-----
Number of SDPs : 2
=====

```

With the vc-switching Epipe service configured and operational, the state of the pw-port can again be shown in the following output. Before the configuration of the vc-switching Epipe, the pw-port had no SDP identifier or VC-ID. Now both entries exist; automatically created by the system when **pw-port 1>epipe "Epipe 13">fpe-id 1** was configured. The SDP identifier of 17281 is allocated from the preconfigured **sdp-id-range**.

```

[/]
A:admin@PE-5# show pw-port 1

=====
PW Port Information
=====
PW Port  Encap      SDP:VC-Id      IfIndex
-----
1         qinq        17281:100001   1526726657
=====

```

The output for SDP 17281 shows that the Far End is fpe_1.a (transit), the Delivery (Del) is MPLS, the LSP type is FPE (F), and that no signaling (Sig) is used for this internal SDP, as follows:

```

[/]
A:admin@PE-5# show service sdp 17281

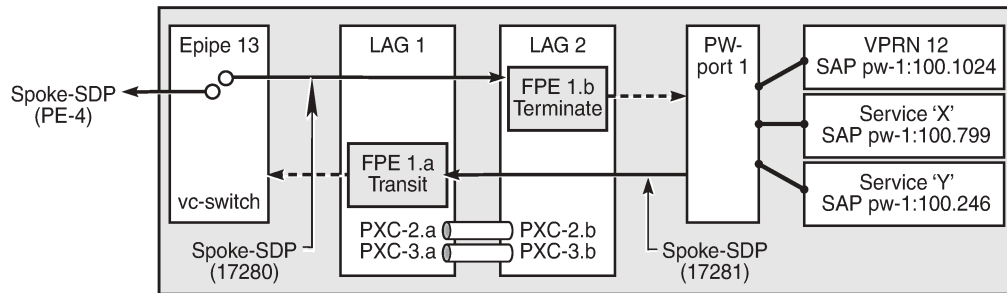
=====
Service Destination Point (Sdp Id : 17281)
=====
SdpId  AdmMTU  OprMTU  Far End      Adm  Opr      Del   LSP  Sig
-----
17281  0        8678    fpe_1.a      Up   Up        MPLS  F    None
=====

```

In SR OS, the combination of SDP ID and VC-ID is always associated with a service. When using AS mode, the system automatically creates an internal VPLS service with ID 2147383649 and a name of `_tmns_InternalVplsService`. This VPLS includes all internal SDPs dynamically created for binding pw-ports to the transit side of the corresponding FPE. The VPLS is an internal construct that does not affect forwarding.

Figure 6: AS mode with redundant FPE shows the components of the FPE from vc-switching Epipe to pw-port.

Figure 6: AS mode with redundant FPE



26266

Next, bind the VPRN service to the pw-port with the relevant VLAN delimiters. CE-4 is using SVLAN tag 100 and CVLAN tag 1024 and both VLANs are encapsulated inside the pseudowire as payload. The following VPRN configuration has two interfaces: the first is toward a directly connected test port used to verify IP connectivity, and the second is toward CE-4 and has a SAP with a pw-port syntax. The SAP pw-1:100.1024 represents pw-port 1 with Q-in-Q encapsulation using SVLAN tag 100 and CVLAN tag 1024 as service delimiters. A BGP session is configured between PE-5 and CE-4 for route exchange. The remainder of the VPRN parameters are generic and are not explained here.

```
# on PE-5:
configure {
  service {
    vprn "VPRN 12 using PXC AS" {
      admin-state enable
      service-id 12
      customer "1"
      autonomous-system 64496
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "64496:12"
          vrf-target {
            community "target:64496:12"
          }
          auto-bind-tunnel {
            resolution any
          }
        }
      }
    }
  }
  bgp {
    group "EBGP" {
      peer-as 64511
      family {
        ipv4 true
      }
    }
    neighbor "192.168.45.1" {
      split-horizon true
      group "EBGP"
      as-override true
      ebgp-default-reject-policy {
        import false
      }
      export {
        policy ["vrf-ebgp-export"]
      }
    }
  }
}
```

```

    }
  }
  interface "Test-Port-C" {
    ipv4 {
      primary {
        address 172.31.105.1
        prefix-length 24
      }
    }
    sap 1/1/3:100 {
    }
  }
  interface "to-CE-4" {
    ipv4 {
      primary {
        address 192.168.45.2
        prefix-length 30
      }
    }
    sap pw-1:100.1024 {
    }
  }
}

```

FPE port dimensioning

After the VPRN service at PE-5 is configured with **admin-state enable**, the EBGP session to CE-4 is established. The relevant routes are exchanged between CE-4 and PE-5 and traffic can be exchanged between test ports B (behind CE-4) and C (connected to PE-5). Initially, traffic is sent unidirectionally from test port C (connected to PE-5) toward port B (connected to CE-4) at a rate of 100 packets/s. To provide a level of entropy for the generated traffic, 100 destination IP addresses are used in the range 172.31.104.2 through 172.31.104.101, and 100 source IP addresses are used in the range 172.31.105.2 through 172.31.105.101.

The following output—in classic CLI—shows a snapshot of a monitor command against LAG 2 (xc-lag-b, or terminating side) incorporating both physical ports. First, note that the input and output rate of packets per second are equal at 100 packets/s, which is not intuitive for a unidirectional traffic flow. This is because the LAG statistics are essentially a copy of the physical port statistics and the physical port consists of two PXC sub-ports that are looped. Logically, this unidirectional traffic flow is forwarded in a single upstream direction from pxc-2.a/pxc-3.a to pxc-2.b/pxc-3.b. Physically, the unidirectional traffic is transmitted by ports 1/2/2 and 1/2/3, then received by the same ports through the loop. Second, note that traffic is load-balanced over both member links (PXC sub-ports) of the LAG. This is because conventional LAG load-balancing mechanisms are used for the FPE LAG, which in the case of a VPRN SAP-to-network relies on source/destination IP address (with optional Layer 4, which is not currently configured).

```
*A:PE-5# monitor lag 2 rate interval 3
```

```
=====
Monitor statistics for LAG ID 2
=====
```

Port-id	Input packets Input bytes Input errors [Input util %]	Output packets Output bytes Output errors [Output util %]
-----snip-----		

```
At time t = 9 sec (Mode: Rate)
```

```
-----
1/2/2!      41                                41

```

	22041		22041	
1/2/3!	0	0.00	0	0.00
	59		59	
	32159		32159	
	0	0.00	0	0.00

Totals	100		100	
	54200		54200	
	0	0.00	0	0.00

! indicates that the port is assigned to a port-xc.

Traffic is then generated unidirectionally upstream from test port B (connected to CE-4) toward port C (connected to PE-5) at a rate of 100 packets/s. Again, to provide a level of entropy for the generated traffic, 100 destination IP addresses are used in the range 172.31.105.2 through 172.31.105.101, and 100 source IP addresses are used in the range 172.31.104.2 through 172.31.104.101. The input/output rates of packets per second are the same, as previously explained. Again, traffic is load-balanced over both member links (PXC sub-ports). This is because hashing of traffic through a vc-switched Epipe service uses source/destination IP information (and optional Layer 4 information, which is not currently configured).

```
*A:PE-5# monitor lag 2 rate interval 3
```

```
=====
```

```
Monitor statistics for LAG ID 2
```

```
=====
```

Port-id	Input packets		Output packets	
	Input bytes		Output bytes	
	Input errors [Input util %]		Output errors [Output util %]	

---snip---				

At time t = 9 sec (Mode: Rate)				

1/2/2!	44		44	
	23848		23848	
	0	0.00	0	0.00
1/2/3!	56		56	
	30352		30352	
	0	0.00	0	0.00

Totals	100		100	
	54200		54200	
	0	0.00	0	0.00

! indicates that the port is assigned to a port-xc.

QoS continuity

When using AS mode, the FPE construct creates internal cross-connects between the vc-switching Epipe and the pw-port. These internal cross-connects function as MPLS tunnels that transit through internal network interfaces on the PXC sub-ports. The internal network interfaces use the default network policy 1 for egress marking and ingress classification/FC mapping. Like all default QoS policies, this network policy cannot be modified (or deleted). Also, it is not possible to use a non-default network policy, because there is no router interface to which the non-default policy can be attached.

The internal cross-connects also use the default network-queue policy named "default". While this policy also cannot be modified, it is possible to configure and apply a non-default network-queue policy (including a port-scheduler-policy, if required) at PXC sub-port level. An example of how this would be applied is

shown in the following output. Where redundant PXC ports are used in an LAG instance, the queue-policy is applied to all member links of the LAG, for consistency.

```
# on PE-5:
configure
  port pxc-2.a {
    admin-state enable
    ethernet {
      network {
        egress {
          queue-policy "non-default"
        }
      }
    }
  }
  port pxc-3.a {
    admin-state enable
    ethernet {
      network {
        egress {
          queue-policy "non-default"
        }
      }
    }
  }
}
```

To demonstrate QoS continuity through the FPE, the following is established:

- **Downstream:** Traffic is generated from test port C (connected to PE-5) toward test port B (connected to CE-4) with DiffServ marking EF at a rate of 100 packets/s. At PE-5 SAP ingress, this traffic is mapped into FC EF.
- **Upstream:** Traffic is generated from test port B (connected to CE-4) toward test port C (connected to PE-5) with DiffServ marking EF at a rate of 100 packets/s. At PE-3, a SAP-ingress QoS policy is used to map the traffic into FC EF.
- The default network QoS policy 1 is used on all network interfaces at PE-3 and PE-5. On egress, this policy marks FC EF as MPLS EXP 5. On ingress, MPLS EXP 5 is mapped to FC EF.
- The default network queue-policy "default" is used on all network interfaces at PE-3 and PE-5. This maps FC EF traffic to queue 6 at ingress and egress.

First, QoS continuity for downstream traffic is validated. The following output shows the relatively simple SAP-egress QoS policy that is applied to the egress of the VPRN interface (pw-port) toward CE-4. No classification of traffic and mapping to FCs are present in the policy, because the classification and mapping have already taken place on the SAP ingress at PE-5 (the SAP facing the test port C).

```
# on PE-5:
configure {
  qos {
    sap-egress "SAP egress 12" {
      policy-id 12
      queue 1 {
        scheduler-parent {
          scheduler-name "aggregate-rate"
          level 2
          weight 10
        }
      }
    }
    queue 2 {
      queue-type best-effort
      rate {
        pir 5000
      }
    }
  }
}
```

```

        cir max
    }
    scheduler-parent {
        scheduler-name "aggregate-rate"
        level 2
        weight 40
        cir-level 2
    }
}
queue 3 {
    queue-type expedited
    rate {
        pir 2000
        cir 2000
    }
    scheduler-parent {
        scheduler-name "aggregate-rate"
        cir-level 3
    }
}
fc be {
    queue 1
}
fc af {
    queue 2
}
fc ef {
    queue 3
}
}

```

The configuration of the Tier 1 scheduler "aggregate-rate" referenced by the child queues in the preceding SAP-egress QoS policy is as follows. The Tier 1 scheduler references a **port-scheduler-policy** using the command **port-parent**. Parenting to a port-scheduler is optional, but allows for inclusion of Preamble and IFG in the QoS scheduling algorithm, which otherwise are not included. The Tier 1 scheduler does not directly reference the port-scheduler-policy by name, but rather inherits any port-scheduler configured on the port to which the child queues are mapped. In this example, the port-scheduler-policy "port-scheduler" is applied to PXC sub-ports pxc-2.b and pxc-3.b (terminating side).

```

# on PE-5:
configure {
    qos {
        scheduler-policy "egress-hqos-scheduler" {
            tier 1 {
                scheduler "aggregate-rate" {
                    rate {
                        pir 1
                    }
                    port-parent {
                    }
                }
            }
        }
        port-scheduler-policy "port-scheduler" {
        }
    }
    port pxc-2.b {
        admin-state enable
        ethernet {
            egress {
                port-scheduler-policy {
                    policy-name "port-scheduler"
                }
            }
        }
    }
}

```



```

    }
  }
}
port pxc-3.b {
  admin-state enable
  ethernet {
    egress {
      port-scheduler-policy {
        policy-name "port-scheduler"
      }
    }
  }
}
}

```

Finally, the SAP-egress QoS policy is applied to the pw-port SAP within the VPRN. The egress H-QoS scheduler is also attached and an override of the rate is configured.

```

# on PE-5:
configure {
  service {
    vprn "VPRN 12 using PXC AS" {
      interface "to-CE-4" {
        sap pw-1:100.1024 {
          egress {
            qos {
              sap-egress {
                policy-name "SAP egress 12"
              }
              scheduler-policy {
                policy-name "egress-hqos-scheduler"
                overrides {
                  scheduler "aggregate-rate" {
                    rate {
                      pir 25000
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}

```

When traffic is generated downstream from test port C toward CE-4 in FC EF at a rate of 100 packets/s, the first point of verification is the VPRN pw-port SAP egress. The following output—in classic CLI—is a **monitor** of the SAP showing that traffic is correctly mapped to queue 3.

```

*A:PE-5# monitor service id 12 sap pw-1:100.1024 rate
=====
Monitor statistics for Service 12 SAP pw-1:100.1024
=====
---snip---
-----
At time t = 11 sec (Mode: Rate)
-----
---snip---
-----
Sap per Queue Stats
-----
                Packets                Octets                % Port
                Util.
-----
---snip---

```

```
Egress Queue 3
For. In/InplusProf   : 0           0           0.00
For. Out/ExcProf     : 100        51600        0.04
Dro. In/InplusProf   : 0           0           0.00
Dro. Out/ExcProf     : 0           0           0.00
```

Monitoring of network interfaces does not show queue statistics (and is not supported on PXC sub-ports), but a verification of the sub-port statistics on the transit side (LAG 1) shows that packets are incrementing in ingress queue 6 on both sub-ports, as follows:

```
[/]
A:admin@PE-5# show port pxc-2.a detail | match "Ingress Queue 6" post-lines 4
Ingress Queue 6          Packets          Octets
  In Profile forwarded   :    711          382518
  In Profile dropped     :     0              0
  Out Profile forwarded  :     0              0
  Out Profile dropped    :     0              0
```

```
[/]
A:admin@PE-5# show port pxc-3.a detail | match "Ingress Queue 6" post-lines 4
Ingress Queue 6          Packets          Octets
  In Profile forwarded   :    404          217352
  In Profile dropped     :     0              0
  Out Profile forwarded  :     0              0
  Out Profile dropped    :     0              0
```

The last point of verification is the network egress interface toward PE-3. Again, a check at the physical port level shows that packets are incrementing in egress queue 6. Therefore, we can conclude that QoS/FC continuity is maintained in the downstream direction.

```
[/]
A:admin@PE-5# show port 1/1/2 detail | match "Egress Queue 6" post-lines 4
Egress Queue 6          Packets          Octets
  In/Inplus Prof fwded  :   2394          1297548
  In/Inplus Prof dropped:     0              0
  Out/Exc Prof fwded    :     0              0
  Out/Exc Prof dropped  :     0              0
```

Next, the upstream QoS continuity is verified. PE-3 is marking traffic generated by test port B to FC EF, which in turn is marked as MPLS EXP 5 by PE-3's default network QoS policy. The following output taken at PE-5 shows that packets are incrementing in ingress queue 6 of the network interface toward PE-3 and confirms that traffic is correctly marked as FC EF at ingress.

```
[/]
A:admin@PE-5# show port 1/1/2 detail | match "Ingress Queue 6" post-lines 4
Ingress Queue 6          Packets          Octets
  In Profile forwarded   :   3458          1874236
  In Profile dropped     :     0              0
  Out Profile forwarded  :     0              0
  Out Profile dropped    :     0              0
```

The next point of verification is the egress side of the PXC sub-ports (pxc-2.a and pxc-3.a) forming the transit side (LAG 1). The sub-port statistics verify that packets are incrementing in egress queue 6 of both sub-ports (as traffic is being load-balanced).

```
[/]
A:admin@PE-5# show port pxc-2.a detail | match "Egress Queue 6" post-lines 4
Egress Queue 6          Packets          Octets
```

```
In/Inplus Prof fwded : 12441 6693258
In/Inplus Prof dropped: 0 0
Out/Exc Prof fwded : 0 0
Out/Exc Prof dropped : 0 0
```

```
[/]
A:admin@PE-5# show port pxc-3.a detail | match "Egress Queue 6" post-lines 4
Egress Queue 6          Packets          Octets
  In/Inplus Prof fwded : 12893          6936434
  In/Inplus Prof dropped: 0              0
  Out/Exc Prof fwded   : 0              0
  Out/Exc Prof dropped : 0              0
```

PXC sub-ports operate in hybrid mode. When the upstream traffic arrives on the PXC sub-ports that form the terminating side of the FPE (pxc-2.b and pxc-3.b), it is mapped to the pw-port SAP-ingress queues, bypassing the ingress network QoS policy and associated ingress network queues. As a result, the MPLS EXP-to-FC mapping cannot be fulfilled and traffic requires reclassification and remapping to the correct FC by the SAP-ingress QoS policy. The following output shows the SAP-ingress QoS policy applied to the pw-port SAP within the VPRN. Because the EXP-to-FC mapping could not be completed, FC reclassification is required in order to map traffic to its original FC before transiting the FPE. In this example, DSCP is used. Also, FC EF is mapped to queue 3.

```
#on PE-5:
configure {
  qos {
    sap-ingress "SAP ingress 12" {
      policy-id 12
      queue 1 {
        scheduler-parent {
          scheduler-name "aggregate-rate"
          level 2
          weight 10
        }
      }
      queue 2 {
        queue-type best-effort
        scheduler-parent {
          scheduler-name "aggregate-rate"
          level 2
          weight 40
          cir-level 2
        }
        rate {
          pir 5000
          cir max
        }
      }
      queue 3 {
        queue-type expedited
        scheduler-parent {
          scheduler-name "aggregate-rate"
          cir-level 3
        }
        rate {
          pir 2000
          cir 2000
        }
      }
      queue 11 {
        rate {
          cir max
        }
      }
    }
  }
}
```

```

    }
  }
  dscp be {
    fc "be"
  }
  dscp af31 {
    fc "af"
  }
  dscp ef {
    fc "ef"
  }
  fc "af" {
    queue 2
  }
  fc "be" {
    queue 1
  }
  fc "ef" {
    queue 3
  }
}

```

For completeness, the configuration of the Tier 1 scheduler "aggregate-rate" referenced by the child queues in the preceding SAP-ingress QoS policy is as follows. Unlike the egress counterpart, there is no parenting to a port-scheduler because this is an egress function only.

```

# on PE-5:
configure {
  qos {
    scheduler-policy "ingress-hqos-scheduler" {
      tier 1 {
        scheduler "aggregate-rate" {
          rate {
            pir 1
          }
        }
      }
    }
  }
}

```

The SAP-ingress QoS policy is applied to the pw-port SAP within the VPRN, together with the ingress H-QoS scheduler. An override of the scheduler rate is also applied.

```

# on PE-5:
configure {
  service {
    vprn "VPRN 12 using PXC AS" {
      interface "to-CE-4" {
        sap pw-1:100.1024 {
          ingress {
            qos {
              sap-ingress {
                policy-name "SAP ingress 12"
              }
              scheduler-policy {
                policy-name "ingress-hqos-scheduler"
                overrides {
                  scheduler "aggregate-rate" {
                    rate {
                      pir 25000
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}

```

```
}

```

With the SAP-ingress policy applied, a monitor output of the SAP in the following output—in classic CLI—verifies that the packets are being received in queue 3 at a rate of 100 packets/s. This verifies the FC continuity in the upstream direction, noting that reclassification and remapping of FC is required at SAP ingress.

```
*A:PE-5# monitor service id 12 sap pw-1:100.1024 rate
=====
Monitor statistics for Service 12 SAP pw-1:100.1024
=====
---snip---
-----
Sap Statistics
-----
---snip---
---snip---
Packets                               Octets
---snip---
Ingress Queue 3 (Unicast) (Priority)
Off. HiPrio      : 0                      0          0.00
Off. LowPrio    : 100                    51647      0.04
Dro. HiPrio     : 0                      0          0.00
Dro. LowPrio    : 0                      0          0.00
For. InProf     : 0                      0          0.00
For. OutProf    : 100                    51647      0.04
```

OAM continuity

The FPE pw-port functionality may be used by redundant routers to provide resilient service termination for a Layer 2 backhaul node implementing a mechanism such as active/standby pseudowire. In SR OS, an active/standby pseudowire is modeled as an Epipe or VPLS service with an endpoint object containing two spoke-SDPs. This form of redundancy relies on the propagation of the Pseudowire Status TLV within an LDP Notification message to convey the operational status of the pseudowires and thereby indicate which one of the pseudowires is active and which one is standby.

The FPE construct uses the concept of a multi-segment pseudowire, implementing Switching-PE (S-PE) functionality to instantiate dynamic cross-connects through the FPE. To verify that LDP status signaling is maintained through this S-PE function, the following is established:

- The Epipe service at PE-3 used for Layer 2 backhaul to the FPE is modified to include an **endpoint** object referenced by two spoke-SDPs.
- The first spoke-SDP has a far end of PE-2 and is configured as **precedence primary**, so becomes the active pseudowire.
- The second spoke-SDP has a far end of PE-5 and is configured with the default precedence 4, so becomes the standby pseudowire.
- Because the endpoint object is configured for **standby-signaling-master**, PE-3 will signal a status of standby toward PE-5.

For completeness, the configuration of the Epipe service at PE-3 is as follows:

```
# on PE-3:
configure {
  service {
```

```

epipe "Epipe 13" {
  admin-state enable
  service-id 13
  customer "1"
  endpoint "redundant-Layer3" {
    standby-signaling master
  }
  spoke-sdp 32:13 {
    endpoint {
      name "redundant-Layer3"
      precedence primary
    }
  }
  spoke-sdp 35:13 {
    endpoint {
      name "redundant-Layer3"
    }
  }
  sap 1/1/3:100.* {
  }
}

```

As shown in the following output, PE-3 has the spoke-SDP to PE-5 (sdp 35:13) as administratively and operationally up, but is signaling a status of standby (pwFwdingStandby).

```

[/]
A:admin@PE-3# show service id 13 sdp 35:13 detail
Admin State      : Up                Oper State      : Up
Local Pw Bits    : pwFwdingStandby
Peer Pw Bits     : None
Admin State      : Disabled          Oper State      : Disabled

```

At PE-5, the signaled status is acknowledged at the far end of the pseudowire in the Peer Pw Bits field.

```

[/]
A:admin@PE-5# show service id 13 sdp 53:13 detail
Admin State      : Up                Oper State      : Up
Local Pw Bits    : None
Peer Pw Bits     : pwFwdingStandby
Admin State      : Disabled          Oper State      : Disabled

```

Typically, an S-PE would propagate the status TLV received from one pseudowire segment into the opposing pseudowire segment in order to provide end-to-end status signaling. However, when using FPE, the SR OS Service Manager process correlates between a pseudowire and its corresponding pw-port SAPs, so can take the necessary actions based upon the operational state of each. Therefore, it is not necessary for the S-PE to propagate the status TLV from one segment to another. This is illustrated in the following output at PE-5, which shows the second segment of the multi-segment pseudowire toward the terminating side fpe_1.b. As described, the status bits are not copied between single segments and all local/peer pseudowire bits remain unset.

```

[/]
A:admin@PE-5# show service id 13 sdp 17280:1 detail
Admin State      : Up                Oper State      : Up
Local Pw Bits    : None
Peer Pw Bits     : None
Admin State      : Disabled          Oper State      : Disabled

```

The pw-port 1 used throughout in this example is internally bound to SDP 17281:

```
[/]
A:admin@PE-5# show pw-port 1

=====
PW Port Information
=====
PW Port   Encap          SDP:VC-Id          IfIndex
-----
1         qinq           17281:100001      1526726657
=====
```

The following output shows that SDP 17281 is operationally down with the flag "stitchingSvcTxDown":

```
[/]
A:admin@PE-5# show service sdp 17281 detail | match "SDP: 17281 Pw-port: 1" post-lines 10
SDP: 17281 Pw-port: 1
-----
VC-Id           : 100001           Admin Status      : up
Encap           : qinq             Oper Status       : down
VC Type         : ether
Dot1Q Ethertype : 0x8100          QinQ Ethertype    : 0x8100
Control Word    : Not Preferred
Entropy Label   : Disabled

Admin Ingress label : 524280          Admin Egress label : 524281
Oper Flags        : stitchingSvcTxDown
```

At service level, the first of the following two outputs shows the state of the SAP bound to pw-port 1. As shown, the operational state is down with an indication that this is due to the port being operationally down. The second output shows that this SAP status is propagated to IP interface level because the interface "to-CE-4" is also shown as operationally down.

```
[/]
A:admin@PE-5# show service id 12 sap pw-1:100.1024 detail | match 'Admin State|Flags'
Admin State      : Up           Oper State        : Down
Flags           : PortOperDown
```

```
[/]
A:admin@PE-5# show router 12 interface "to-CE-4"

=====
Interface Table (Service: 12)
=====
Interface-Name      Adm    Opr(v4/v6)  Mode    Port/SapId
IP-Address          PfxState
-----
to-CE-4             Up     Down/Down   VPRN    pw-1:100.1024
192.168.45.2/30    n/a
-----
Interfaces : 1
=====
```

To verify a failover, the state of the active/standby pseudowire is transitioned by failing the active pseudowire between PE-3 and PE-2. This causes PE-3 to declare the pseudowire to PE-5 active, which clears the standby status bits. This action causes the SDP (17281) bound to pw-port 1 to become

operationally up, followed by pw-port 1 and its associated SAPs, followed by the VPRN IP interface "to-CE-4".

```
208 2021/05/25 11:34:33.485 CEST MINOR: SVCMGR #2103 Base
"Status of service 2147483649 (customer 1) changed to administrative state: up, operational
state: up"

209 2021/05/25 11:34:33.485 CEST MINOR: SVCMGR #2313 Base
"Status of SDP Bind 53:13 in service 13 (customer 1) peer PW status bits changed to none"

210 2021/05/25 11:34:33.486 CEST MAJOR: SVCMGR #2210 Base
"Processing of an access port state change event is finished and the status of all affected
SAPs on port pw-1 has been updated."

211 2021/05/25 11:34:33.486 CEST WARNING: SNMP #2005 vprn12 to-CE-4
"Interface to-CE-4 is operational"
```

This example in the AS mode section illustrated how notification of a downstream failure is propagated through the components of the PXC in AS mode and reflected in the status of the pw-port (and its associated services). Also, if a pw-port fails due to a PXC failure (for example, the physical port fails), it is just as important that the operational state is propagated externally. In the case of pseudowire backhaul (as in the example), this would be achieved by setting the LDP pseudowire status bits to psnIngressFault and psnEgressFault toward the far end.

Conclusion

This chapter demonstrates the principles of PXC configuration. The PXC can be used to provide a relatively simple back-to-back cross-connect operation in DVSM mode, or it can be used in AS mode to provide an integrated path through the FPE with automated cross-connects used to simplify the provisioning process. In both DVSM mode and AS mode, the PXC can be configured as redundant or non-redundant. A relatively simple use-case of terminating an Epipe into a VPRN has been demonstrated for both modes.

There are a large number of use-cases where frame/packet preprocessing is required before service termination. The workaround for these use-cases has previously been a physical external loop, but can now be resolved logically and internally through use of the PXC.

Router Configuration

This section provides configuration information for the following topics:

- [6PE Next-Hop Resolution](#)
- [Aggregate Route Indirect Next-Hop Option](#)
- [Bi-Directional Forwarding Detection](#)
- [LFA Policies Using OSPF as IGP](#)
- [PBR/PBF Redundancy](#)
- [Weighted ECMP for 6PE over RSVP-TE LSPs](#)

6PE Next-Hop Resolution

This chapter provides information about 6PE next hop resolution.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

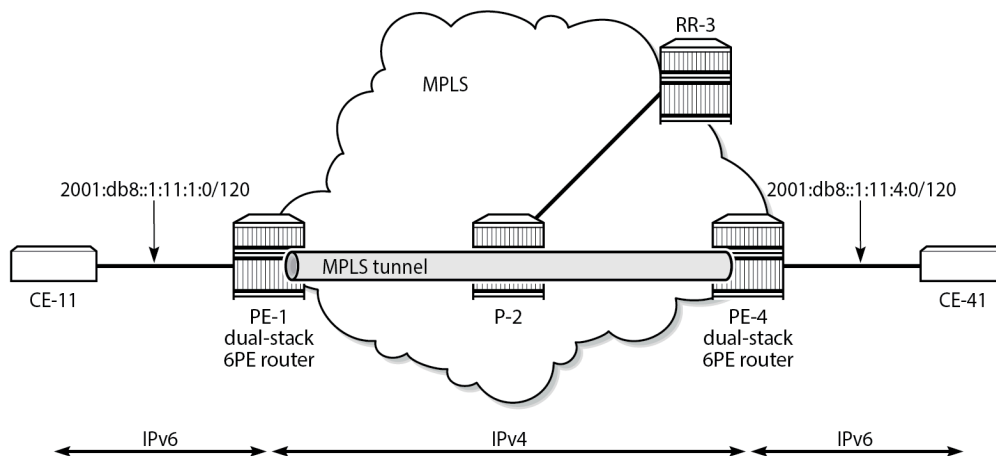
This chapter was initially written based on SR OS Release 14.0.R7, but the MD-CLI in the current edition corresponds to SR OS Release 23.7.R1.

In Releases earlier than 14.0.R1, only label distribution protocol label switched paths (LDP LSPs) could be used to resolve IPv6 provider edge (6PE) next hops. Additional options for 6PE next hop resolution are supported in SR OS Release 14.0.R1, and later. In this chapter, examples are shown with 6PE next hop resolution to different kinds of MPLS tunnels, such as LDP, RSVP-TE, SR-ISIS, and BGP tunnels.

Overview

IPv6 provider edge (6PE) enables IPv6 communication between IPv6 domains over an IPv4 multi-protocol label switching (MPLS) cloud. IPv6 packets are forwarded in an MPLS tunnel from one dual-stack 6PE router to another, as shown in [Figure 7: IPv6 provider edge \(6PE\)](#).

Figure 7: IPv6 provider edge (6PE)



26333

The 6PE route next hop resolution is configured using the following command:

```
*[ex:/configure router "Base" bgp next-hop-resolution labeled-routes transport-tunnel family
label-ipv6]
A:admin@PE-1# resolution ?

resolution <keyword>
<keyword> - (none|filter|any)
Default   - filter

Resolution mode for binding BGP routes to tunnel types
```

With 6PE next hop resolution set to **any**, the tunnels are selected based on availability and tunnel table manager (TTM) preference. The order of preference of TTM tunnels is: RSVP, SR-TE, LDP, SR-OSPF, SR-ISIS, and UDP.

For LDP to be used, it is sufficient to enable LDP on the interfaces in the MPLS network.

For RSVP-TE to be used, an RSVP-TE LSP to the 6PE next-hop destination must be available or configured. For segment routing to be used, an SR-signaled path to the 6PE next hop destination must be available or configured. For BGP labeled routes to be used, the 6PE next hop must have been learned via a BGP peering carrying labeled unicast routes and placed in the active route table.

With 6PE next hop resolution set to filter (default), a subset of protocols is required, and LDP is automatically included in the protocol list in the resolution filter. The following **info** command shows an empty list of protocols when no resolution filter has been defined; the **info detail** command shows that LDP is (implicitly) included.

```
*[ex:/configure router "Base" bgp next-hop-resolution labeled-routes transport-tunnel family
label-ipv6 resolution-filter]
A:admin@PE-1# info

*[ex:/configure router "Base" bgp next-hop-resolution labeled-routes transport-tunnel family
label-ipv6 resolution-filter]
A:admin@PE-1# info detail
  bgp false
  ldp true
  rsvp false
  sr-isis false
  sr-ospf false
  sr-te false
  udp false
  sr-policy false
  rib-api false
  mpls-fwd-policy false
  sr-ospf3 false
```

If the 6PE next hop can be resolved to an LDP tunnel, this tunnel is preferred to a BGP tunnel.

It is possible to explicitly exclude LDP from the list, as follows:

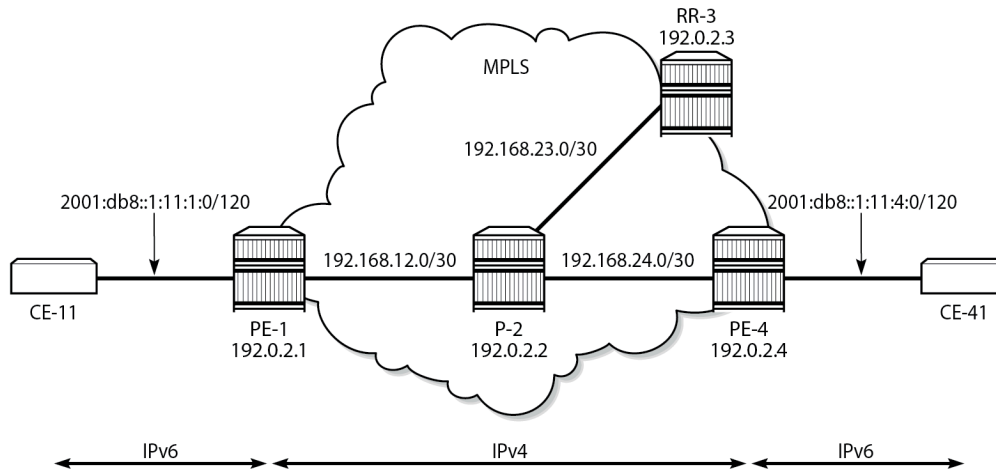
```
*[ex:/configure router "Base" bgp next-hop-resolution labeled-routes transport-tunnel family
label-ipv6 resolution-filter]
A:admin@PE-1# ldp false

*[ex:/configure router "Base" bgp next-hop-resolution labeled-routes transport-tunnel family
label-ipv6 resolution-filter]
A:admin@PE-1# info
  bgp true
  ldp false
```

Configuration

Figure 8: Example topology shows the example topology with two dual-stack 6PE routers (PE-1 and PE-4), a core router (P-2), and a route reflector (RR-3). IPv4 is used in the core network; IPv6 is used between the CEs and the PEs.

Figure 8: Example topology



26334

The initial configuration on the nodes is as follows:

- Cards, MDAs, ports
- Router interfaces
- IS-IS as IGP in the core IPv4 network (alternatively, OSPF can be used)
- LDP enabled on the interfaces between the PEs and P-2, but not toward RR-3
- MPLS and RSVP enabled on the interfaces between the PEs and P-2, but not toward RR-3

BGP configuration

BGP is configured for the label-IPv6 address family on PE-1, PE-4, and RR-3, but not on P-2. The BGP configuration on both PEs defines how the 6PE next hops will be resolved: the resolution filter contains three options (LDP, RSVP, and SR-ISIS). The BGP configuration is identical on PE-1 and PE-4.

```
# on PE-1, PE-4:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      split-horizon true
      next-hop-resolution {
        labeled-routes {
          transport-tunnel {
            family label-ipv6 {
              resolution-filter {
                # ldp true          # default
```

```

        rsvp true
        sr-isis true
    }
    # resolution filter #default
}
}
}
}
group "IBGP" {
    peer-as 64496
    export {
        policy ["export-6pe"]
    }
}
neighbor "192.0.2.3" {
    group "IBGP"
    family {
        label-ipv6 true
    }
}
}

```

The export policy "export-6pe" exports the IPv6 prefixes that are local to the PE, for example, on PE-1: 2001:db8::1:11:1:0/120, and is defined as follows:

```

# on PE-1, PE-4:
configure {
    policy-options {
        policy-statement "export-6pe" {
            entry 10 {
                from {
                    protocol {
                        name [direct]
                    }
                }
                action {
                    action-type accept
                }
            }
            default-action {
                action-type reject
            }
        }
    }
}

```

The BGP configuration on RR-3 does not include any export policy or any next-hop resolution settings, as follows:

```

# on RR-3:
configure {
    router "Base"
        autonomous-system 64496
        bgp {
            split-horizon true
            group "IBGP" {
                peer-as 64496
                cluster {
                    cluster-id 192.0.2.3
                }
            }
        }
        neighbor "192.0.2.1" {
            group "IBGP"
            family {
                label-ipv6 true
            }
        }
    }
}

```

```

    }
  }
  neighbor "192.0.2.4" {
    group "IBGP"
    family {
      label-ipv6 true
    }
  }
}

```

IES configuration

On PE-1, an IES is configured with IPv6 addresses on the interface toward CE-11, as follows:

```

# on PE-1:
configure {
  service {
    ies "IES-1" {
      admin-state enable
      description "6PE"
      service-id 1
      customer "1"
      interface "int-PE-1-CE-11" {
        sap 1/1/c3/1:1 {
        }
        ipv6 {
          address 2001:db8::1:11:1:1 {
            prefix-length 120
          }
        }
      }
    }
  }
}

```

The configuration on PE-4 is similar; the IPv6 address on interface "int-PE-4-CE-41" is different: 2001:db8::1:11:4:1/120.

A BGP labeled IPv6 tunnel, which is active in the IPv6 routing table, is established between the PEs, as follows:

```

[/]
A:admin@PE-1# show router route-table ipv6

=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]
Type      Proto    Age           Pref
Metric
-----
2001:db8::1:11:1:0/120
int-PE-1-CE-11
Local     Local    00h04m06s    0
0
2001:db8::1:11:4:0/120
192.0.2.4 (tunneled)
Remote   BGP_LABEL 00h03m56s    170
20
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====

```

CE-11 can send IPv6 packets with source address 2001:db8::1:11:1:11 to destination address 2001:db8::1:11:4:41 on CE-41, as follows:

```
[/]
A:admin@PE-1# ping 2001:db8::1:11:4:41 router-instance "CE-11" source-address
2001:db8::1:11:1:11
PING 2001:db8::1:11:4:41 56 data bytes
64 bytes from 2001:db8::1:11:4:41 icmp_seq=1 hlim=62 time=8.79ms.
64 bytes from 2001:db8::1:11:4:41 icmp_seq=2 hlim=62 time=3.69ms.
64 bytes from 2001:db8::1:11:4:41 icmp_seq=3 hlim=62 time=3.47ms.
64 bytes from 2001:db8::1:11:4:41 icmp_seq=4 hlim=62 time=3.65ms.
64 bytes from 2001:db8::1:11:4:41 icmp_seq=5 hlim=62 time=2.60ms.

---- 2001:db8::1:11:4:41 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 2.60ms, avg = 4.44ms, max = 8.79ms, stddev = 2.21ms
```

6PE next hop resolved to an LDP tunnel

On PE-1, the route for prefix 2001:db8::1:11:4:0/120 uses a tunnel to 6PE next hop 192.0.2.4, as follows:

```
[/]
A:admin@PE-1# show router route-table 2001:db8::1:11:4:0/120

=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]                Metric
-----
2001:db8::1:11:4:0/120           Remote BGP_LABEL 00h04m17s 170
  192.0.2.4 (tunneled)                20
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```

LDP is enabled on the interfaces between the PEs and P-2, which is sufficient for 6PE next hop resolution to an LDP tunnel. RSVP-TE tunnels have a higher priority, but no MPLS LSPs have been configured yet on the PEs. The tunnel table on PE-1 shows that the only tunnel to 6PE next hop 192.0.2.4 is an LDP tunnel, as follows:

```
[/]
A:admin@PE-1# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner    Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.2/32     ldp      MPLS 65537   9     192.168.12.2 10
192.0.2.4/32     ldp      MPLS 65538   9     192.168.12.2 20
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
```

E = Inactive best-external BGP route
k = RIB-API or Forwarding Policy backup hop
=====

Alternatively, the following show command can be used: the only tunnel on slot 1 (card 1) to 6PE next hop 192.0.2.4 is an LDP tunnel:

```
[/]
A:admin@PE-1# show router fp-tunnel-table 1 192.0.2.4/32

=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol      Tunnel-ID
Lbl/SID                                     NextHop      Intf/Tunnel
Lbl/SID (backup)                           NextHop      (backup)
-----
192.0.2.4/32                               LDP          -
524286                                       192.168.12.2 1/1/c1/1:1000
-----
Total Entries : 1
=====
```

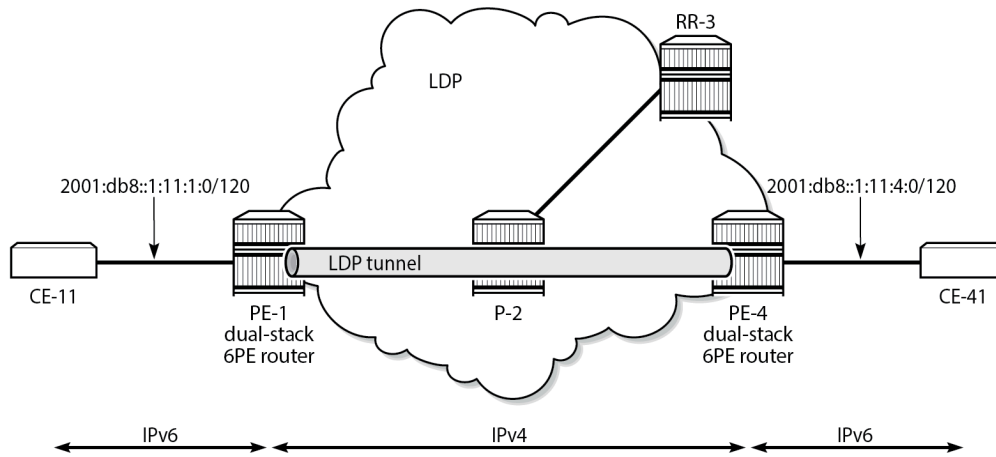
The extended route information for IPv6 prefix 2001:db8::1:11:4:0/120 shows that the 6PE next hop 192.0.2.4 is resolved to an LDP tunnel:

```
[/]
A:admin@PE-1# show router route-table 2001:db8::1:11:4:0/120 extensive

=====
Route Table (Router: Base)
=====
Dest Prefix      : 2001:db8::1:11:4:0/120
Protocol         : BGP_LABEL
Age              : 00h05m42s
Preference       : 170
Indirect Next-Hop : 192.0.2.4
Label            : 2
QoS              : Priority=n/c, FC=n/c
Source-Class     : 0
Dest-Class       : 0
ECMP-Weight      : N/A
Resolving Next-Hop : 192.0.2.4 (LDP tunnel)
Metric           : 20
ECMP-Weight      : N/A
-----
No. of Destinations: 1
=====
```

Figure 9: 6PE next hop resolved to an LDP tunnel shows that the 6PE next hop is resolved to an LDP tunnel. No other tunnels are available in the IPv4 core network.

Figure 9: 6PE next hop resolved to an LDP tunnel



26335

6PE next hop resolved to an RSVP-TE tunnel

MPLS and RSVP are enabled on the interfaces between the PEs and P-2. On both PEs, an RSVP-TE LSP is configured toward the peer PE; for example, on PE-1:

```
# on PE-1:
configure {
  router Base
    mpls {
      path "empty" {
        admin-state enable
      }
      lsp "LSP-PE-1-PE-4" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.4
        primary "empty" {
        }
      }
    }
  }
}
```

The configuration is similar on PE-4. No additional configuration is required on P-2.

The following output shows that two tunnels are available to 6PE next hop 192.0.2.4/32: an LDP tunnel and an RSVP-TE tunnel:

```
[/]
A:admin@PE-1# show router fp-tunnel-table 1 192.0.2.4/32

=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol  Tunnel-ID
Lbl/SID
```

NextHop Lbl/SID (backup) NextHop (backup)		Intf/Tunnel

192.0.2.4/32	LDP	-
524286		
192.168.12.2		1/1/c1/1:1000
192.0.2.4/32	RSVP	1
524284		
192.168.12.2		1/1/c1/1:1000

Total Entries : 2		

=====		

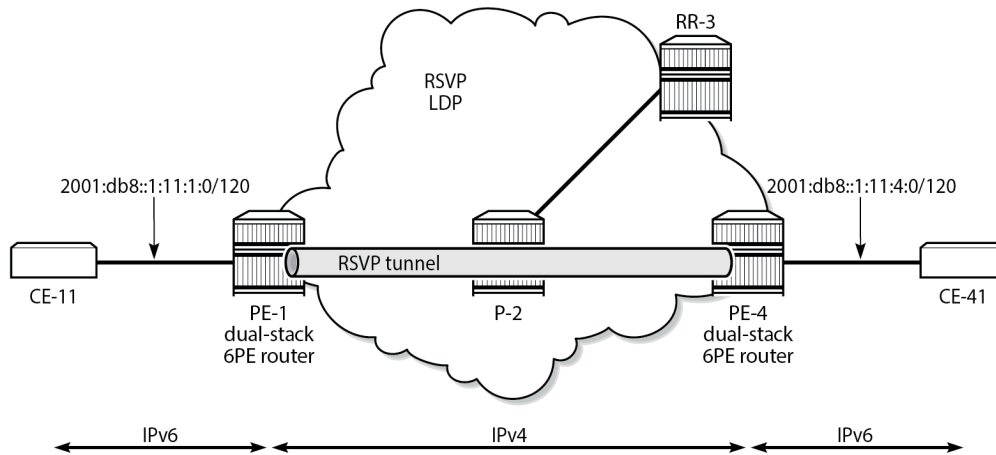
For 6PE next hop resolution, RSVP-TE tunnels are preferred to any other tunnel type in the tunnel table, so the BGP next hop 192.0.2.4 will be resolved to an RSVP-TE tunnel, as follows:

```
[/]
A:admin@PE-1# show router route-table 2001:db8::1:11:4:0/120 extensive

=====
Route Table (Router: Base)
=====
Dest Prefix          : 2001:db8::1:11:4:0/120
Protocol             : BGP_LABEL
Age                  : 00h00m43s
Preference           : 170
Indirect Next-Hop   : 192.0.2.4
Label                : 2
QoS                  : Priority=n/c, FC=n/c
Source-Class         : 0
Dest-Class           : 0
ECMP-Weight          : N/A
Resolving Next-Hop : 192.0.2.4 (RSVP tunnel:1)
Metric               : 20
ECMP-Weight          : N/A
-----
No. of Destinations: 1
=====
```

Figure 10: 6PE next hop resolved to an RSVP-TE tunnel shows that the 6PE next hop 192.0.2.4 is resolved to an RSVP-TE tunnel, even though an LDP tunnel is available too.

Figure 10: 6PE next hop resolved to an RSVP-TE tunnel



26336

6PE next hop resolved to an SR-ISIS tunnel

Segment routing is enabled for IS-IS on PE-1, P-2, and PE-4. The configuration is similar on each of these nodes; the only difference is the IPv4 node SID index on the system interface. The SR-ISIS configuration on PE-1 is as follows:

```
# on PE-1:
configure {
  router "Base" {
    mpls-labels {
      sr-labels {
        start 20000
        end 20099
      }
    }
  }
  isis 0 {
    advertise-router-capability area
    segment-routing {
      admin-state enable
      prefix-sid-range {
        start-label 20000
        max-index 99
      }
    }
    interface "system" {
      ipv4-node-sid {
        index 1
      }
    }
  }
}
```

For more information about SR-ISIS, see the [Segment Routing with IS-IS Control Plane](#) chapter.

The following output shows that three tunnels are available toward 6PE next hop 192.0.2.4/32:

```
[/]
A:admin@PE-1# show router fp-tunnel-table 1 192.0.2.4/32
```

```

=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol      Tunnel-ID
Lbl/SID                                    NextHop      Intf/Tunnel
Lbl/SID (backup)                          NextHop      (backup)
-----
192.0.2.4/32                               LDP          -
524286                                       192.168.12.2 1/1/c1/1:1000
192.0.2.4/32                               RSVP         1
524284                                       192.168.12.2 1/1/c1/1:1000
192.0.2.4/32                               SR-ISIS-0    524291
20004                                       192.168.12.2 1/1/c1/1:1000
-----
Total Entries : 3
=====

```

RSVP-TE tunnels are preferred; therefore, the 6PE next hop 192.0.2.4 is resolved to the RSVP-TE tunnel, as follows:

```

[/]
A:admin@PE-1# show router route-table 2001:db8::1:11:4:0/120 extensive

=====
Route Table (Router: Base)
=====
Dest Prefix          : 2001:db8::1:11:4:0/120
Protocol             : BGP_LABEL
Age                  : 00h02m12s
Preference           : 170
Indirect Next-Hop    : 192.0.2.4
Label                : 2
QoS                  : Priority=n/c, FC=n/c
Source-Class         : 0
Dest-Class           : 0
ECMP-Weight          : N/A
Resolving Next-Hop : 192.0.2.4 (RSVP tunnel:1)
Metric               : 20
ECMP-Weight          : N/A
-----
No. of Destinations: 1
=====

```

To verify that LDP tunnels are preferred over SR-ISIS tunnels, the RSVP-TE LSPs are disabled, as follows:

```

# on PE-1:
configure {
  router "Base"
    mpls {
      lsp "LSP-PE-1-PE-4" {
        admin-state disable
      }
    }
}

```

```
}

```

The following output shows that two tunnels are available toward 6PE next hop 192.0.2.4/32: an LDP tunnel and an SR-ISIS tunnel.

```
[/]
A:admin@PE-1# show router fp-tunnel-table 1 192.0.2.4/32

=====
IPv4 Tunnel Table Display

Legend:
Label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol      Tunnel-ID
 Lbl/SID                                     NextHop      Intf/Tunnel
 Lbl/SID (backup)                           NextHop      (backup)
-----
192.0.2.4/32                               LDP          -
 524286                                     192.168.12.2 1/1/c1/1:1000
192.0.2.4/32                               SR-ISIS-0    524291
 20004                                     192.168.12.2 1/1/c1/1:1000
-----
Total Entries : 2
=====
```

For 6PE next-hop resolution, the LDP tunnel is preferred over the SR-ISIS tunnel, as follows:

```
[/]
A:admin@PE-1# show router route-table 2001:db8::1:11:4:0/120 extensive

=====
Route Table (Router: Base)
=====
Dest Prefix      : 2001:db8::1:11:4:0/120
Protocol         : BGP_LABEL
Age              : 00h00m31s
Preference      : 170
Indirect Next-Hop : 192.0.2.4
Label           : 2
QoS              : Priority=n/c, FC=n/c
Source-Class     : 0
Dest-Class       : 0
ECMP-Weight      : N/A
Resolving Next-Hop : 192.0.2.4 (LDP tunnel)
Metric           : 20
ECMP-Weight      : N/A
-----
No. of Destinations: 1
=====
```

When LDP is disabled on interface "int-PE-1-P-2" on PE-1, the only remaining tunnel is an SR-ISIS tunnel, as follows:

```
# on PE-1:
```

```

configure {
  router "Base" {
    ldp {
      interface-parameters {
        interface "int-PE-1-P-2" {
          admin-state disable
        }
      }
    }
  }
}

[/]
A:admin@PE-1# show router fp-tunnel-table 1 192.0.2.4/32

=====
IPv4 Tunnel Table Display

Legend:
Label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol      Tunnel-ID
Lbl/SID                                     NextHop
Lbl/SID (backup)                           Intf/Tunnel
NextHop (backup)
-----
192.0.2.4/32                               SR-ISIS-0      524291
20004
  192.168.12.2                               1/1/c1/1:1000
-----
Total Entries : 1
=====

```

The 6PE next hop 192.0.2.4 is resolved to an SR-ISIS tunnel, as follows:

```

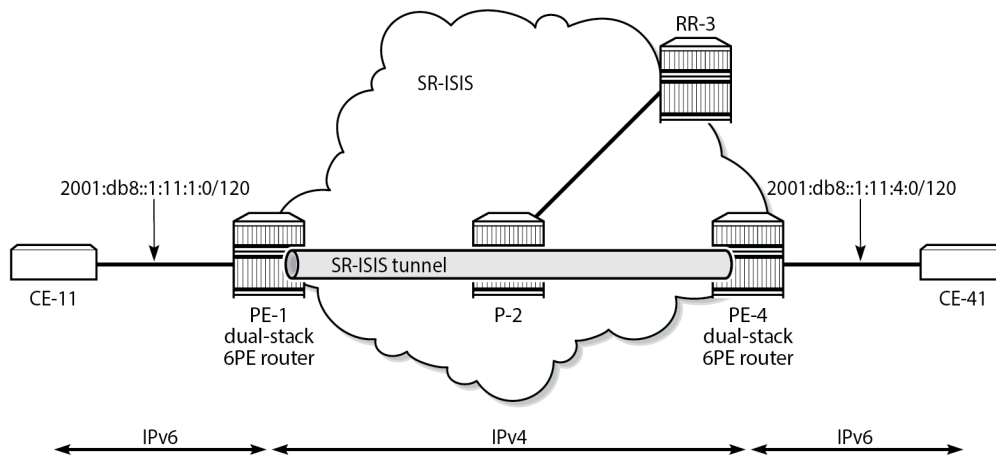
[/]
A:admin@PE-1# show router route-table 2001:db8::1:11:4:0/120 extensive

=====
Route Table (Router: Base)
=====
Dest Prefix          : 2001:db8::1:11:4:0/120
Protocol             : BGP_LABEL
Age                  : 00h00m55s
Preference           : 170
Indirect Next-Hop    : 192.0.2.4
Label                : 2
QoS                  : Priority=n/c, FC=n/c
Source-Class         : 0
Dest-Class           : 0
ECMP-Weight          : N/A
Resolving Next-Hop : 192.0.2.4 (SR-ISIS tunnel:524291)
Metric               : 20
ECMP-Weight          : N/A
-----
No. of Destinations: 1
=====

```

Figure 11: 6PE next hop resolved to an SR-ISIS tunnel shows that the 6PE next hop 192.0.2.4 is resolved to an SR-ISIS tunnel after the RSVP-TE LSPs are disabled and LDP is disabled on the interfaces between the PEs and P-2. No other tunnels are available.

Figure 11: 6PE next hop resolved to an SR-ISIS tunnel



26337

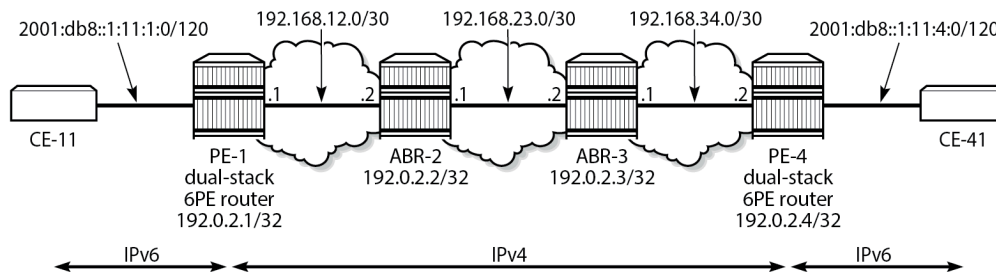
6PE next-hop resolution to a BGP IPv4 tunnel

The preceding example cannot be extended with BGP labeled IPv4 tunnels. The reason is that for BGP to work, some underlying MPLS signaling protocol is required, such as RSVP-TE or LDP. Because BGP tunnels have a very low preference, they will not be used when an LDP or RSVP-TE tunnel is available to the 6PE next hop.

This section shows a seamless MPLS example where 6PE next hops are resolved to BGP labeled IPv4 routes, because no LDP tunnel is available to the 6PE next hop in a different IGP topology (in this example, LDP is configured, not RSVP-TE). For a description of this seamless MPLS implementation, see the [Seamless MPLS: Isolated IGP/LDP Domains and Labeled BGP](#) chapter.

Figure 12: Example topology for seamless MPLS shows the example topology for seamless MPLS with two aggregation networks and one core network.

Figure 12: Example topology for seamless MPLS



26338

Different IS-IS instances are configured: IS-IS instance 0 is configured in the core, whereas IS-IS instance 1 is configured in the aggregation networks. On the area border routers (ABRs) ABR-2 and ABR-3, two instances of IS-IS are configured: IS-IS instance 0 for the core and IS-IS instance 1 for the aggregation network. PE-1 and PE-4 will only learn routes to destinations within their respective aggregation networks; ABRs learn routes within one aggregation network and the core network. LDP is configured on all

interfaces, but PE-1 will not have an LDP binding for prefix 192.0.2.4/32, as shown in the following output. Therefore, 6PE next hop 192.0.2.4 cannot be resolved to an LDP tunnel.

```
[/]
A:admin@PE-1# show router ldp bindings active prefixes ipv4

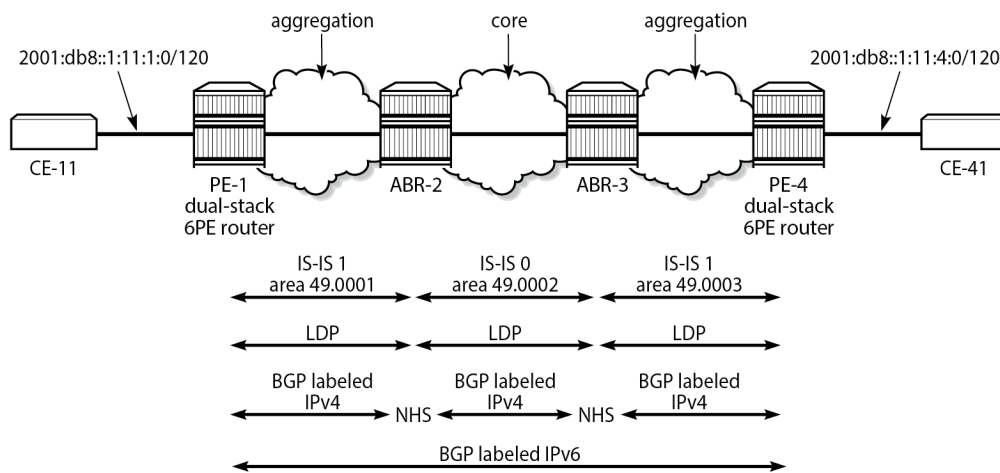
=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.1/32                          Pop
524287                                 --
--                                     --

192.0.2.2/32                          Push
--                                     524287
192.168.12.2                          1/1/c1/1:1000

-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

Figure 13: Configured protocols for seamless MPLS shows the configured protocols for this example: IS-IS instances, LDP, BGP labeled IPv4 with the ABRs as route reflector with **next-hop-self** (NHS) option, and BGP labeled IPv6 peering between PE-1 and PE-4.

Figure 13: Configured protocols for seamless MPLS



26339

The following initial configuration on ABR-2 includes two IS-IS instances in different areas. IS-IS instance 0 with area ID 49.0002 is configured in the core network; IS-IS instance 1 with area ID 49.0001 is configured in the aggregation network between PE-1 and ABR-2. LDP is configured on each router interface.

```
# on ABR-2:
configure {
  router "Base" {
    interface "int-ABR-2-ABR-3" {
      port 1/1/c3/1:1000
      ipv4 {
        primary {
          address 192.168.23.1
          prefix-length 30
        }
      }
    }
    interface "int-ABR-2-PE-1" {
      port 1/1/c2/1:1000
      ipv4 {
        primary {
          address 192.168.12.2
          prefix-length 30
        }
      }
    }
    interface "system" {
      ipv4 {
        primary {
          address 192.0.2.2
          prefix-length 32
        }
      }
    }
  }
  isis 0 {
    admin-state enable
    level-capability 2
    area-address [49.0002]
    interface "int-ABR-2-ABR-3" {
      interface-type point-to-point
    }
  }
}
```

```

    }
    interface "system" {
    }
}
isis 1 {
  admin-state enable
  level-capability 2
  area-address [49.0001]
  interface "int-ABR-2-PE-1" {
    interface-type point-to-point
  }
  interface "system" {
  }
}
ldp {
  interface-parameters {
    interface "int-ABR-2-ABR-3" {
      ipv4 {
      }
    }
    interface "int-ABR-2-PE-1" {
      ipv4 {
      }
    }
  }
}
}

```

The configuration is similar on the other nodes. Only the ABRs have two IS-IS instances configured; the PEs only have one IS-IS instance.

BGP needs to be configured for the label-IPv4 and label-IPv6 address families:

- The label-IPv4 address family is used with the ABRs as RR in the aggregation network. Each ABR is configured with the **next-hop-self** option. BGP label-IPv4 peering is between the ABRs without RR.
- The label-IPv6 address family is used between PE-1 and PE-4. The BGP session can only be established after the BGP labeled IPv4 routes have been exchanged between PE-1 and PE-4.

BGP is configured on PE-1 as follows:

```

# on PE-1:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      split-horizon true
      next-hop-resolution {
        labeled-routes {
          transport-tunnel {
            family label-ipv6 {
              resolution-filter {
                bgp true
                # ldp true   # LDP is by default included
              }
            }
          }
        }
      }
    }
  }
  group "IBGPv4" {
    peer-as 64496
    export {
      policy ["export-sys"]
    }
  }
}

```

```

    group "IBGPv6" {
        peer-as 64496
        export {
            policy ["export-6pe"]
        }
    }
    neighbor "192.0.2.2" {
        group "IBGPv4"
        family {
            label-ipv4 true
        }
    }
    neighbor "192.0.2.4" {
        group "IBGPv6"
        family {
            label-ipv6 true
        }
    }
}

```

The configuration is similar on PE-4, but the neighbor IP addresses are different.

The resolution filter will include LDP as well as BGP, because it is added automatically. However, no LDP tunnel will be available from PE-1 to PE-4, or vice versa; therefore, BGP labeled IPv4 will be used.

The "export-sys" policy exports the IPv4 system address of the PE and is defined as follows:

```

# on PE-1, PE-4:
configure {
    policy-options {
        prefix-list "system" {
            prefix 192.0.2.0/24 type longer {
            }
        }
    }
    policy-statement "export-sys" {
        entry 10 {
            from {
                prefix-list ["system"]
                protocol {
                    name [direct]
                }
            }
            action {
                action-type accept
            }
        }
        default-action {
            action-type reject
        }
    }
}

```

The "export-6pe" policy exports the local labeled IPv6 routes and is the same in the preceding examples:

```

# on PE-1, PE-4:
configure {
    policy-options {
        policy-statement "export-6pe" {
            entry 10 {
                from {
                    protocol {
                        name [direct]
                    }
                }
            }
        }
    }
}

```

```
        action {
            action-type accept
        }
    }
    default-action {
        action-type reject
    }
}
```

The BGP configuration on ABR-2 has two different groups for BGP labeled IPv4 peering: one toward the aggregation network—with the ABR as RR—and one toward the core, as follows:

```
# on ABR-2:
configure {
    router "Base" {
        autonomous-system 64496
        bgp {
            advertise-inactive true
            split-horizon true
            group "IBGPv4-agg" {
                next-hop-self true
                peer-as 64496
                cluster {
                    cluster-id 192.0.2.2
                }
            }
            group "IBGPv4-core" {
                next-hop-self true
                peer-as 64496
            }
            neighbor "192.0.2.1" {
                group "IBGPv4-agg"
                family {
                    label-ipv4 true
                }
            }
            neighbor "192.0.2.3" {
                group "IBGPv4-core"
                family {
                    label-ipv4 true
                }
            }
        }
    }
}
```

The configuration is similar on ABR-3, but the neighbor IP addresses and the cluster ID are different.

The ABRs are configured with the **next-hop-self** option for both groups. The 6PE next hop 192.0.2.4 will have next hop ABR-2 on PE-1, which can be resolved to an LDP tunnel. On ABR-2, 6PE next hop 192.0.2.4 will have ABR-3 as next hop, which can be resolved to an LDP tunnel. On ABR-3, the 6PE next hop 192.0.2.4 can be resolved to an LDP tunnel (no active BGP route to 192.0.2.4/32 on ABR-3 because the route via IS-IS is preferred).

The **advertise-inactive** option is required for ABR-2 to export a BGP route for prefix 192.0.2.1/32, which is not active on ABR-2, because an IS-IS route is available for this prefix and IS-IS routes are preferred over BGP routes.

The IES configuration is the same as in the preceding example.

When the labeled IPv4 routes are exchanged between PE-1 and PE-4, the BGP labeled session using IPv6 peering can be established between PE-1 and PE-4, as follows:

```
[/]
```

```
A:admin@PE-1# show router bgp summary all

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId          AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
                   PktSent OutQ
-----
192.0.2.2
Def. Inst          64496      12   0 00h03m34s 1/1/1 (Lbl-IPv4)
                   12   0
192.0.2.4
Def. Inst          64496       8   0 00h01m47s 1/1/1 (Lbl-IPv6)
                   8   0
-----
```

For IPv6 prefix 2001:db8::1:11:4:0/120 on PE-1, 6PE next hop 192.0.2.4 is resolved to a BGP tunnel, as follows:

```
[/]
A:admin@PE-1# show router route-table 2001:db8::1:11:4:0/120 extensive

=====
Route Table (Router: Base)
=====
Dest Prefix          : 2001:db8::1:11:4:0/120
Protocol             : BGP_LABEL
Age                  : 00h01m40s
Preference           : 170
Indirect Next-Hop    : 192.0.2.4
Label                : 2
QoS                  : Priority=n/c, FC=n/c
Source-Class         : 0
Dest-Class           : 0
ECMP-Weight          : N/A
Resolving Next-Hop : 192.0.2.4 (BGP tunnel)
Metric               : 1000
ECMP-Weight          : N/A
-----
No. of Destinations: 1
=====
```

The BGP labeled IPv4 route to 192.0.2.4 has different next hops in different nodes, because both ABRs set the **next-hop-self** option. On PE-1, the BGP labeled IPv4 route for prefix 192.0.2.4 has next hop 192.0.2.2 and uses an LDP tunnel to reach ABR-2 within the aggregation network, as follows:

```
[/]
A:admin@PE-1# show router fp-tunnel-table 1 192.0.2.4/32

=====
IPv4 Tunnel Table Display
=====
Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                     Protocol      Tunnel-ID
-----
```

```

Lbl/SID
  NextHop
Lbl/SID (backup)
  NextHop (backup)
-----
192.0.2.4/32          BGP          -
524282
  192.0.2.2          LDP
-----
Total Entries : 1
=====

```

On ABR-2, the BGP labeled route to 192.0.2.4/32 has next hop 192.0.2.3 and uses an LDP tunnel in the core network to reach ABR-3, as follows:

```

[/]
A:admin@ABR-2# show router fp-tunnel-table 1 192.0.2.4/32

=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination          Protocol          Tunnel-ID
Lbl/SID
  NextHop
Lbl/SID (backup)
  NextHop (backup)
-----
192.0.2.4/32          BGP          -
524282
  192.0.2.3          LDP
-----
Total Entries : 1
=====

```

On ABR-3, no BGP labeled IPv4 route is active for prefix 192.0.2.4 because IS-IS routes are preferred to BGP routes. An LDP tunnel is used toward PE-4 in the aggregation network, as follows:

```

[/]
A:admin@ABR-3# show router fp-tunnel-table 1 192.0.2.4/32

=====
IPv4 Tunnel Table Display

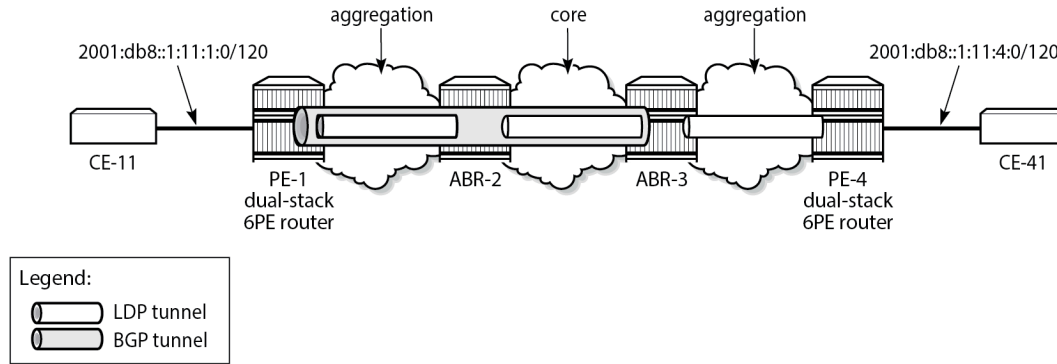
Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination          Protocol          Tunnel-ID
Lbl/SID
  NextHop
Lbl/SID (backup)
  NextHop (backup)
-----
192.0.2.4/32          LDP          -
524287
  192.168.34.2          1/1/c1/1:1000
-----

```

```
-----
Total Entries : 1
-----
=====
```

Figure 14: BGP labeled IPv4 tunnel for 192.0.2.4/32 using LDP tunnels shows the BGP and LDP tunnels used for 6PE next hop 192.0.2.4/32.

Figure 14: BGP labeled IPv4 tunnel for 192.0.2.4/32 using LDP tunnels



26340

Conclusion

The 6PE next hops can be resolved to different types of MPLS tunnels, each with a different preference.

Aggregate Route Indirect Next-Hop Option

This chapter provides information about aggregate routes with indirect next-hop option.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

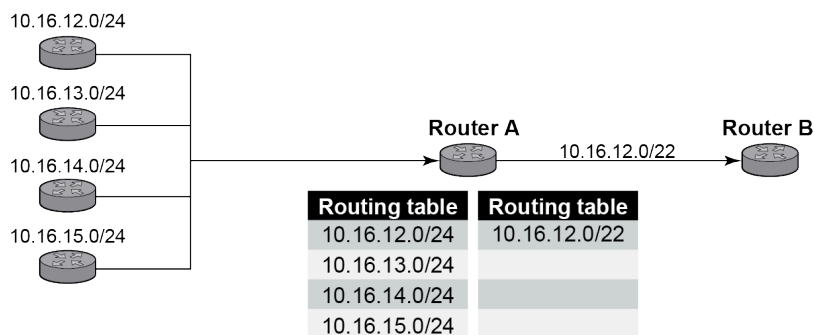
Applicability

This chapter was initially written based on SR OS Release 11.0.R1. The MD-CLI in the current edition corresponds to SR OS Release 22.10.R1.

Overview

In SR OS nodes, IPv4 and IPv6 aggregate routes can be configured. A configured aggregate route that has the best preference for the prefix is activated, and therefore, added to the routing table, when it has at least one contributing route; the aggregate route is removed from the routing table when there are no longer any contributing routes. A contributing route is any route installed in the forwarding table that is a more specific match of the aggregate. For example, the route 10.16.12.0/24 is a contributing route to the aggregate route 10.16.12.0/22, but for this same aggregate, the routes 10.16.0.0/16 and 10.0.0.0/8 are not contributing routes.

Figure 15: Aggregate routes



al_0294

In [Figure 15: Aggregate routes](#), Router A can advertise all four routes or one aggregate route. By aggregating the four routes, fewer updates are sent on the link between routers A and B, router B needs to maintain a smaller routing table resulting in better convergence and router B saves on computational resources by evaluating fewer entries in its routing table.

It is possible to configure an indirect hop for aggregate routes. The indirect next hop specifies where packets will be forwarded if they match the aggregate route, but not a more specific route in the IP forwarding table.

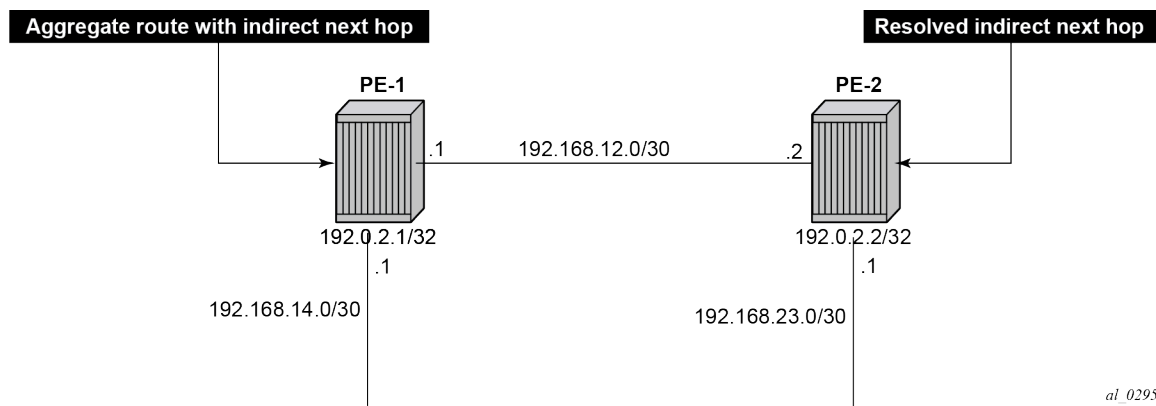
Different network operators have different requirements on how to forward a packet that matches an aggregate route but not any of the more specific routes in the forwarding table that activated the aggregate. In general, there are three different options:

1. The packet can be forwarded according to the next-most specific route, ignoring the aggregate route. This can lead to routing loops in some topologies.
2. The packet can be discarded.
3. The packet can be forwarded toward an indirect next-hop address that is configured by the operator. The indirect next-hop could be the address of a threat management server that analyzes the packets it receives for security threats. This option requires the aggregate route to be installed in the forwarding table with a resolved next-hop interface determined from a route lookup of the indirect next-hop address.

Configuration

The example topology with two PEs is shown in [Figure 16: Example topology](#).

Figure 16: Example topology



al_0295

Initial configuration

The nodes have the following basic configuration:

- cards, MDAs
- ports
- router interfaces

The router interfaces on PE-1 are configured as follows:

```
# on PE-1:
configure {
  router "Base" {
    interface "int-PE-1-PE-2" {
      port 1/1/c1/1:1000
      ipv4 {
        primary {
          address 192.168.12.1
          prefix-length 30
        }
      }
    }
  }
}
```

```

    }
  }
}
interface "int-PE-1-PE-4" {
  port 1/1/c2/1:1000
  ipv4 {
    primary {
      address 192.168.14.1
      prefix-length 30
    }
  }
}
interface "system" {
  ipv4 {
    primary {
      address 192.0.2.1
      prefix-length 32
    }
  }
}
}

```

The configuration on PE-2 is similar. The IP addresses are shown in [Figure 16: Example topology](#). In this example, static routes are configured. There is no need for an IGP, but it could be configured.

Aggregate route with indirect next hop option

This feature adds the **indirect** keyword and an associated IP address parameter to the **aggregate** command in the configuration contexts of the base router and of VPRN services.

The aggregate route configuration command in the base router context is as follows:

```

*[ex:/configure router "Base" aggregates aggregate 10.16.12.0/22]
A:admin@PE-1# ?

Immutable fields      - indirect

aggregator            + Enter the aggregator context
apply-groups          - Apply a configuration group at this level
apply-groups-exclude - Exclude a configuration group at this level
as-set                - Use AS_SET path segment type for the aggregate route
community             - Community name that is added to the aggregate route
description            - Text description
discard-component-    - Advertise aggregate with aggregate route community set
  communities
local-preference      - Local preference used when aggregate route is exported
policy                - Policy name for the aggregated route
summary-only          - Advertise the aggregate route only
tunnel-group          - Tunnel group from which to associate the MC IPsec state

Choice: next-hop
blackhole              +- Enable the blackhole context
indirect             :- Address of the indirect next hop

```

Parameters:

- **indirect** — This indicates that the aggregate route has an indirect address. The indirect option is mutually exclusive with the black-hole option.

- <ip-address> — Installing an aggregate route with an indirect next-hop is supported for both IPv4 and IPv6 prefixes. However, if the aggregate prefix is IPv6, the indirect next-hop must be an IPv6 address and if the aggregate prefix is IPv4, the indirect next-hop must be an IPv4 address.

If an indirect next-hop is not resolved, the aggregate route will show up as black-hole.

The aggregate route 10.16.12.0/22 is configured as follows:

```
# on PE-1:
configure {
  router "Base" {
    aggregates {
      aggregate 10.16.12.0/22 {
        community ["64496:64498"]
        indirect 192.168.11.11
      }
    }
  }
}
```

This creates an aggregate route, but there are no contributing routes that are more specific defined yet. Therefore, the aggregate route remains inactive:

```
[/]
A:admin@PE-1# show router aggregate

=====
Legend: G - generate-icmp enabled
=====
Aggregates (Router: Base)
=====
Prefix                               Aggr IP-Address  Aggr AS
  Summary                             AS Set          State
  NextHop                             Community       NextHopType
-----
10.16.12.0/22                         0.0.0.0         0
  False                               False           Inactive
  192.168.11.11                       64496:64498    Indirect
-----
No. of Aggregates: 1
=====
```

The inactive aggregate route does not appear in the routing table:

```
[/]
A:admin@PE-1# show router route-table

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                   Type  Proto  Age      Pref
  Next Hop[Interface Name]           Metric
-----
192.0.2.1/32                         Local Local  00h02m31s 0
  system                             0
192.168.12.0/30                      Local Local  00h02m31s 0
  int-PE-1-PE-2                       0
192.168.14.0/30                      Local Local  00h02m31s 0
  int-PE-1-PE-4                       0
-----
No. of Routes: 3
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
```

Configure contributing routes to activate the aggregate route

The aggregate route remains inactive as long as there is no contributing route which is more specific than the aggregate route. The following contributing routes are statically configured on PE-1:

```
# on PE-1:
configure {
  router "Base" {
    static-routes {
      route 10.16.12.0/24 route-type unicast {
        next-hop "192.168.14.2" {
          admin-state enable
        }
      }
      route 10.16.13.0/24 route-type unicast {
        next-hop "192.168.14.2" {
          admin-state enable
        }
      }
      route 10.16.14.0/24 route-type unicast {
        next-hop "192.168.14.2" {
          admin-state enable
        }
      }
      route 10.16.15.0/24 route-type unicast {
        next-hop "192.168.14.2" {
          admin-state enable
        }
      }
    }
  }
}
```

As a result, the aggregate route becomes active:

```
[/]
A:admin@PE-1# show router aggregate

=====
Legend: G - generate-icmp enabled
=====
Aggregates (Router: Base)
=====
```

Prefix	Aggr IP-Address	Aggr AS
Summary	AS Set	State
NextHop	Community	NextHopType
10.16.12.0/22	0.0.0.0	0
False	False	Active
192.168.11.11	64496:64498	Indirect

```
-----
No. of Aggregates: 1
=====
```

The active aggregate route is added to the route table, as well as the contributing routes:

```
[/]
A:admin@PE-1# show router route-table

=====
```

```
Route Table (Router: Base)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]
-----
10.16.12.0/22
Black Hole
10.16.12.0/24
192.168.14.2
10.16.13.0/24
192.168.14.2
10.16.14.0/24
192.168.14.2
10.16.15.0/24
192.168.14.2
192.0.2.1/32
system
192.168.12.0/30
int-PE-1-PE-2
192.168.14.0/30
int-PE-1-PE-4
-----
No. of Routes: 8
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
* indicates that the corresponding row element may have been truncated.
```

Dest Prefix[Flags] Next Hop[Interface Name]	Type	Proto	Age Metric	Pref
10.16.12.0/22 Black Hole	Blackh*	Aggr	00h00m29s 0	130
10.16.12.0/24 192.168.14.2	Remote	Static	00h00m29s 1	5
10.16.13.0/24 192.168.14.2	Remote	Static	00h00m29s 1	5
10.16.14.0/24 192.168.14.2	Remote	Static	00h00m29s 1	5
10.16.15.0/24 192.168.14.2	Remote	Static	00h00m29s 1	5
192.0.2.1/32 system	Local	Local	00h03m20s 0	0
192.168.12.0/30 int-PE-1-PE-2	Local	Local	00h03m20s 0	0
192.168.14.0/30 int-PE-1-PE-4	Local	Local	00h03m20s 0	0

The aggregate route is black-holed because the next hop is not resolved. There is no route to 192.168.11.0/24.

Configure resolving route to indirect next hop

A static route is configured on PE-1 to the indirect next hop, as follows:

```
# on PE-1:
configure {
  router "Base" {
    static-routes {
      route 192.168.11.0/24 route-type unicast {
        next-hop "192.168.12.2" {
          admin-state enable
        }
      }
    }
  }
}
```

In the route table, the aggregate route is no longer black-holed. The next hop for the indirect next hop is 192.168.12.2 (PE-2).

```
[/]
A:admin@PE-1# show router route-table

Route Table (Router: Base)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]
-----
10.16.12.0/22
Remote Aggr
00h00m03s 130
```

Dest Prefix[Flags] Next Hop[Interface Name]	Type	Proto	Age Metric	Pref
10.16.12.0/22	Remote	Aggr	00h00m03s	130

```

192.168.12.2
10.16.12.0/24 Remote Static 00h00m55s 5
    192.168.14.2 1
10.16.13.0/24 Remote Static 00h00m55s 5
    192.168.14.2 1
10.16.14.0/24 Remote Static 00h00m55s 5
    192.168.14.2 1
10.16.15.0/24 Remote Static 00h00m55s 5
    192.168.14.2 1
192.0.2.1/32 Local Local 00h03m45s 0
    system 0
192.168.11.0/24 Remote Static 00h00m03s 5
    192.168.12.2 1
192.168.12.0/30 Local Local 00h03m45s 0
    int-PE-1-PE-2 0
192.168.14.0/30 Local Local 00h03m45s 0
    int-PE-1-PE-4 0
-----
No. of Routes: 9
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

In this example, PE-2 is the resolved indirect next hop and it has a route for prefix 10.16.12.0/22:

```

# on PE-2:
configure {
  router "Base" {
    static-routes {
      route 10.16.12.0/22 route-type unicast {
        next-hop "192.168.23.2" {
          admin-state enable
        }
      }
    }
  }
}

```

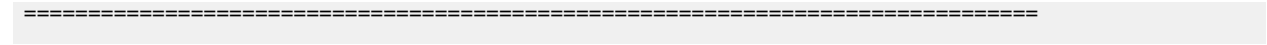
The route table on PE-2 looks as follows:

```

[/]
A:admin@PE-2# show router route-table

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type  Proto  Age      Pref
  Next Hop[Interface Name]                       Metric
-----
10.16.12.0/22                                     Remote Static 00h00m00s 5
    192.168.23.2                                  1
192.0.2.2/32                                     Local  Local  00h04m03s 0
    system                                         0
192.168.12.0/30                                  Local  Local  00h04m03s 0
    int-PE-2-PE-1                                 0
192.168.23.0/30                                  Local  Local  00h04m03s 0
    int-PE-2-PE-3                                 0
-----
No. of Routes: 4
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested

```



Conclusion

Aggregate routes offer several advantages, the key being reduction in the routing table size and overcoming routing loops, among other things. Aggregate routes with indirect next hop option helps in faster network convergence by decreasing the number of route table changes. This example shows how to configure aggregate routes with indirect next hop option.

Bi-Directional Forwarding Detection

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was originally written for SR OS Release 8.0.R4. The MD-CLI in the current edition corresponds to SR OS Release 23.3.R1.

Overview

Bi-directional forwarding detection (BFD) is a lightweight protocol that provides rapid path failure detection between two systems. It has been published as a series of RFCs: RFC 5880, RFC 5881, RFC 5882, RFC 5883, and RFC 5884.

If a system running BFD stops receiving BFD messages on an interface, it will determine that there has been a failure in the path and notify other protocols associated with the interface. BFD is useful in situations where two nodes are interconnected through either an optical dense wavelength division multiplexing (DWDM) or Ethernet network. In both cases, the physical network has numerous extra devices which are not part of the Layer 3 network and therefore, the Layer 3 nodes are incapable of detecting failures which occur in the physical network on spans to which the Layer 3 devices are not directly connected.

BFD protocol provides rapid link continuity checking between network devices, and the state of BFD can be propagated to IP routing protocols to drastically reduce convergence time in cases where a physical network error occurs in a transport network.

RFC 5880 defines two modes of operation for BFD:

- Asynchronous mode (supported) — Uses periodic BFD control messages to test the path between systems. If a number (configured as **multiplier**) of BFD hello packets are not received, the session is considered down.
- Demand mode (not supported)

In addition to the two operational modes, an echo function is defined. SR OS routers only support response sending, which is looping back received BFD messages to the original sender.

BFD is running between two peers and supported for scenarios such as:

- BFD for IS-IS
- BFD for OSPF
- BFD for PIM
- BFD for static routes
- BFD for RSVP

- BFD for I-LDP
- BFD for T-LDP
- BFD for MPLS-TP
- BFD for OSPF CE-PE adjacencies
- BFD for VRRP
- BFD for SRRP
- BFD for IPSec

Many of these BFD scenarios are described in this chapter.

Configuration

BFD packets are processed both locally on the IOM CPU and centrally on the CPM.

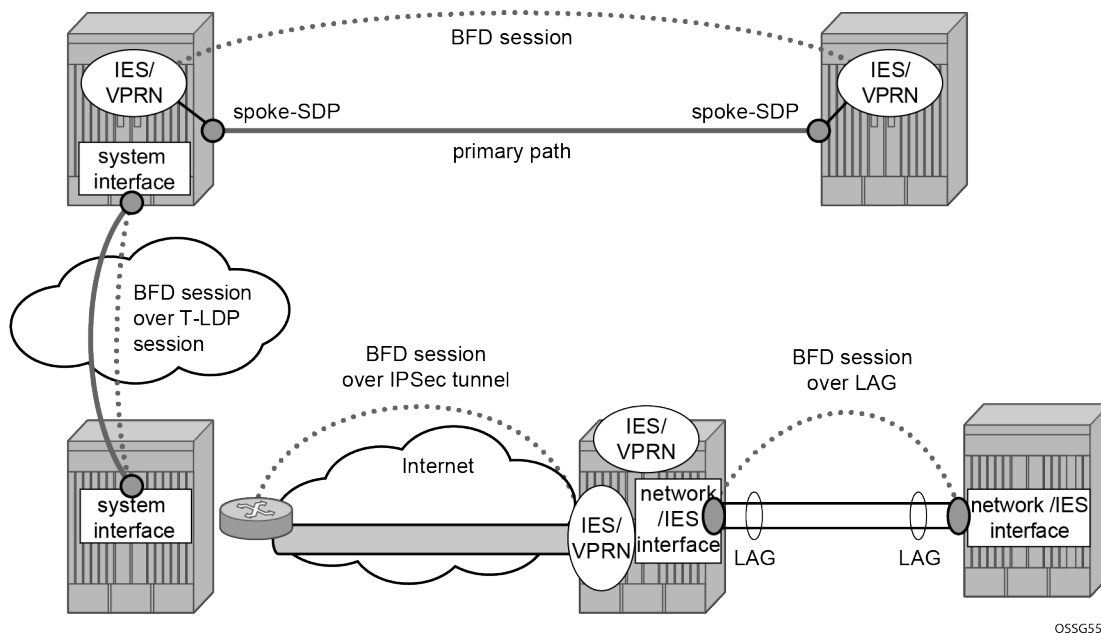
The CPM is able to centrally generate the BFD packets at a subsecond interval as low as 10 ms. The BFD state machine is implemented in software. BFD packet generation can be selectively delegated to CPM hardware as needed. This is applicable when subsecond operations or exceeding the IOM scaling limits is required.

The following applications require BFD to run centrally on the SF/CPM and a centralized session will be created independently of the type explicitly declared by the user:

- BFD for IES/VP RN over spoke SDP
- BFD for LAG and VSM interfaces
- Protocol associations using loopback and system interfaces (for example, BFD for T-LDP)
- BFD for IPSec sessions
- BFD sessions associated with multi-hop peering (BGP)

[Figure 17: BFD centralized sessions](#) shows the most relevant scenarios where centralized BFD sessions are used.

Figure 17: BFD centralized sessions



On the other end, when the two peers are directly connected, the BFD session is local by default, but the user can choose what session type (local or centralized) to implement.

As general rule, the following steps are required to configure and enable a BFD session when peers are directly connected:

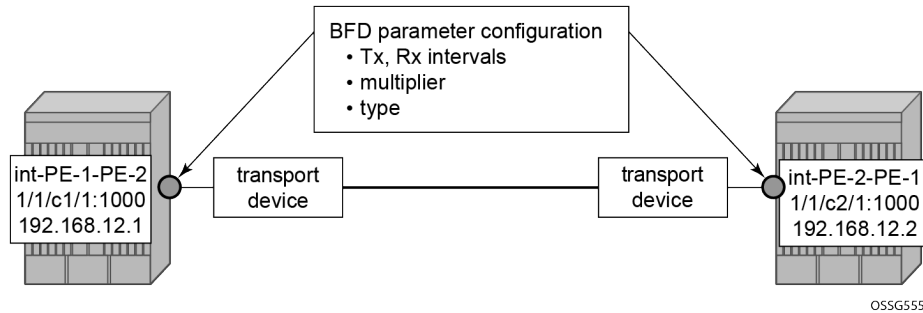
1. configure BFD parameters on the peering interfaces
2. check that the Layer 3 protocol, that is to be bound to BFD, is up and running
3. enable BFD under the Layer 3 protocol interface.

Because most of the following procedures share the same first step, it is described only once in the next section and then referred to in subsequent sections.

BFD base parameter configuration and troubleshooting

The reference topology for the generic configuration of BFD over two local peers is shown in [Figure 18: BFD interface configuration](#).

Figure 18: BFD interface configuration



The user needs to configure base level BFD on interfaces between the peers PE-1 and PE-2.

```
# on PE-1:
configure {
  router "Base" {
    interface "int-PE-1-PE-2"
      port 1/1/c1/1:1000
      ipv4 {
        bfd {
          admin-state enable
        }
        primary {
          address 192.168.12.1
          prefix-length 30
        }
      }
  }
}
```

```
# on PE-2:
configure {
  router "Base" {
    interface "int-PE-2-PE-1" {
      port 1/1/c2/1:1000
      ipv4 {
        bfd {
          admin-state enable
        }
        primary {
          address 192.168.12.2
          prefix-length 30
        }
      }
    }
  }
}
```

The default values for the BFD parameters are:

- transmit interval 100 ms
- receive interval 100 ms
- multiplier 3

```
*[ex:/configure router "Base" interface "int-PE-1-PE-2" ipv4 bfd]
A:admin@PE-1# info detail
  admin-state enable
  transmit-interval 100
  receive 100
  multiplier 3
## echo-receive
```

```
type auto
```

The following **show** commands are used to verify the BFD configuration on the router interfaces on PE-1 and PE-2.

On PE-1:

```
[/]
A:admin@PE-1# show router bfd interface

=====
BFD Interface
=====
Interface name          Tx Interval    Rx Interval    Multiplier
-----
int-PE-1-PE-2          100            100            3
-----
No. of BFD Interfaces: 1
=====
```

On PE-2:

```
[/]
A:admin@PE-2# show router bfd interface

=====
BFD Interface
=====
Interface name          Tx Interval    Rx Interval    Multiplier
-----
int-PE-2-PE-1          100            100            3
-----
No. of BFD Interfaces: 1
=====
```



Note: BFD is an asynchronous protocol, so it is possible to configure different transmit and receive intervals on the two peers. This is because BFD transmit and receive interval values are signaled in the BFD packets while establishing the BFD session.

The configurable BFD parameters are the following:

```
*[ex:/configure router "Base" interface "int-PE-1-PE-2" ipv4 bfd]
A:admin@PE-1# ?

admin-state      - Administrative state of BFD sessions
echo-receive     - Minimum echo interval over this interface
multiplier       - Number of consecutive BFD messages missed from the peer
receive         - BFD receive interval over this interface
transmit-interval - BFD transmit interval over this interface
type            - Local termination point for the BFD session
```

By default, the BFD type is auto, but it is possible to force the BFD session to be centrally managed by the CPM hardware: **type cpm-np**.

Regarding the echo function, it is possible to set the minimum echo receive interval, in milliseconds, for the BFD session.

The base BFD configuration on the router interfaces is not sufficient for a BGP session to come up:

```
*A:PE-1# show router bfd session

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id          State      Tx Pkts   Rx Pkts
  Rem Addr/Info/SdpId:VcId  Multipl   Tx Intvl  Rx Intvl
  Protocols          Type      LAG Port  LAG ID
  Loc Addr                               LAG name
-----
No Matching Entries Found
=====
```

Configuring the BFD parameters on the interface does not enable BFD sessions. BFD can be enabled afterward, for instance, in IS-IS.



Note: If a BFD session is active on an interface, it is possible to modify the BFD intervals and the multiplier on the interface, but not the BFD type. To change the BFD type, the BFD session must be disabled manually, which causes the upper layer protocols bound to it to be brought down as well.

If a BFD session is active on the interface, an attempt to modify the BFD type triggers the following error message:

```
[ex:/configure router "Base" interface "int-PE-1-PE-2" ipv4 bfd]
A:admin@PE-1# type cpm-np

*[ex:/configure router "Base" interface "int-PE-1-PE-2" ipv4 bfd]
A:admin@PE-1# commit
INFO: BFD #1001: configure router "Base" interface "int-PE-1-PE-2" - Inconsistent value - BFD
sessions active on this interface. Cannot change BfdType on this interface
```

Forcing a centralized session in the case of directly connected peers can be useful when:

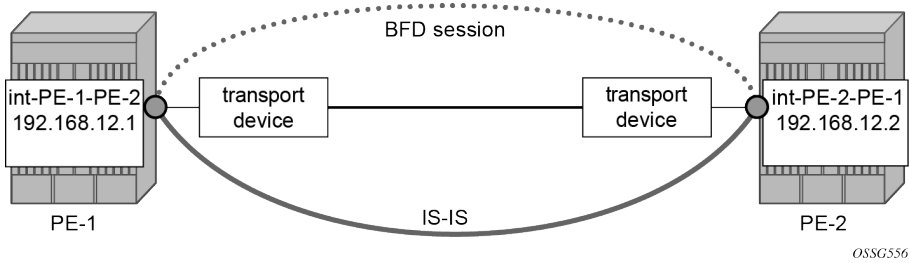
- lower Tx and Rx intervals are desired (down to 10 ms instead of 100 ms supported by local sessions)
- no more local (IOM) sessions are available
- the maximum limit of 500 packets per second per IOM has been reached

The instructions illustrated in following paragraphs are required to complete the configuration and enable BFD.

BFD for IS-IS

The goal of this section is to configure BFD on a network interlink between two SR OS nodes that are IS-IS peers. [Figure 19: BFD for ISIS](#) shows the used topology.

Figure 19: BFD for ISIS



For the base BFD configuration, see the [BFD base parameter configuration and troubleshooting](#) section.

On PE-1, BFD is applied to the IS-IS interface between PE-1 and PE-2:

```
# on PE-1:
configure {
  router "Base" {
    isis 0 {
      interface "int-PE-1-PE-2" {
        bfd-liveness {
          ipv4 {
          }
        }
      }
    }
  }
}
```

When BFD is only applied on PE-1 and not on PE-2, the BFD session on PE-1 remains down, as follows:

```
[/]
A:admin@PE-1# show router bfd session

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
```

Session Id	State	Tx Pkts	Rx Pkts
Rem Addr/Info/SdpId:VcId	Multipl	Tx Intvl	Rx Intvl
Protocols	Type	LAG Port	LAG ID
Loc Addr			LAG name
int-PE-1-PE-2	Down	11	0
192.168.12.2	3	1000	100
isis	iom	N/A	N/A
192.168.12.1			

```
-----
No. of BFD sessions: 1
=====
```

On PE-2, BFD is enabled on the interface to PE-1, as follows:

```
# on PE-2:
configure {
  router "Base" {
    isis 0 {
      interface "int-PE-2-PE-1" {
        bfd-liveness {
        }
      }
    }
  }
}
```

```

        ipv4 {
        }
    }

```

The following command verifies that the local IOM BFD session is operational between PE-1 and PE-2.

On PE-1:

```

[/]
A:admin@PE-1# show router bfd session
=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id                State      Tx Pkts  Rx Pkts
Rem Addr/Info/SdpId:VcId  Multipl   Tx Intvl Rx Intvl
Protocols                 Type      LAG Port  LAG ID
Loc Addr                  LAG name
-----
int-PE-1-PE-2            Up        231      179
192.168.12.2             3         100      100
isis                     iom       N/A      N/A
192.168.12.1
-----
No. of BFD sessions: 1
=====

```

On PE-2:

```

[/]
A:admin@PE-2# show router bfd session
=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id                State      Tx Pkts  Rx Pkts
Rem Addr/Info/SdpId:VcId  Multipl   Tx Intvl Rx Intvl
Protocols                 Type      LAG Port  LAG ID
Loc Addr                  LAG name
-----
int-PE-2-PE-1            Up        152      151
192.168.12.1             3         100      100
isis                     iom       N/A      N/A
192.168.12.2
-----
No. of BFD sessions: 1
=====

```

If the command shows that the BFD session is down, troubleshoot it by first checking that the protocol that is bound to it is up: for instance, check the IS-IS adjacency, as follows:

```

[/]
A:admin@PE-1# show router isis adjacency "int-PE-1-PE-2"

```

```

=====
Rtr Base ISIS Instance 0 Adjacency
=====
System ID          Usage State Hold Interface          MT-ID
-----
PE-2              L1L2 Up    22  int-PE-1-PE-2          0
-----
Adjacencies : 1
=====

```

If the IS-IS adjacency is up, then check whether a BFD resource limit has been reached (maximum number of (local or centralized) sessions or maximum number of packets per second per IOM).

If the overloaded limit is the maximum supported number of sessions, the cause is shown in log 99 (maxSessionsPerSlot).

In this case, when one of the running sessions is manually removed or goes down, then the additional configured session will come up. If the IOM limit is reached, it is possible to bring up the session by changing the session type to centralized.

To check if the IOM CPU is able to start more local BFD sessions, execute a **show router bfd session summary** command:

```

[/]
A:admin@PE-1# show router bfd session summary

```

```

=====
BFD Session Summary
=====
Termination      Session Count
-----
central          0
cpm-np           0
iom, slot 1      1
iom, slot 2      0
iom, slot 3      0
iom, slot 4      0
iom, slot 5      0
iom, slot 6      0
Total            1
=====

```

The **show router bfd session src <ip-address> detail** command can help debugging the BFD session. The sent and received counters are not supported for cpm-np type sessions.

```

[/]
A:admin@PE-1# show router bfd session src 192.168.12.1 detail

```

```

=====
BFD Session
=====
Remote Address : 192.168.12.2
Local Address  : 192.168.12.1
Admin State    : Up                               Oper State    : Up
Protocols      : isis
Rx Interval    : 100                               Tx Interval   : 100
Multiplier    : 3                                 Echo Interval : 0
Recd Msgs     : 681                               Sent Msgs     : 718
Up Time       : 0d 00:00:53                       Up Transitions : 1
Last Down Time : 0d 00:00:32                       Down Transitions : 0
Version Mismatch : 0

```


Forwarding Information

```

Local Discr      : 1
Local Diag       : 0 (None)
Local Min Tx     : 100
Last Sent        : 04/20/2023 16:17:48
Type             : iom
Remote Discr     : 1
Remote Diag      : 0 (None)
Remote Min Tx    : 100
Remote C-flag    : 1
Last Recv       : 04/20/2023 16:17:48

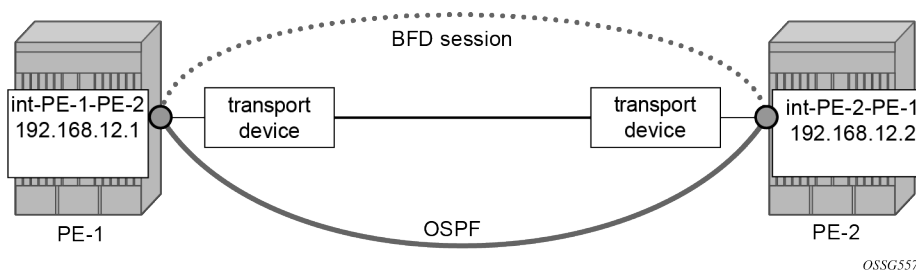
Local State      : Up
Local Mode       : Async
Local Mult       : 3
Local Min Rx     : 100

Remote State     : Up
Remote Mode      : Async
Remote Mult      : 3
Remote Min Rx    : 100
    
```

BFD for OSPF

The goal of this section is to configure BFD on a network interlink between two SR OS nodes that are OSPF peers. [Figure 20: BFD for OSPF](#) shows the topology for this scenario.

Figure 20: BFD for OSPF



The base BFD configuration is described in the section [BFD base parameter configuration and troubleshooting](#).

In this section, BFD is applied on the OSPF interfaces, as follows:

```

# on PE-1:
configure {
  router "Base" {
    ospf 0 {
      admin-state enable
      traffic-engineering true
      area 0.0.0.0 {
        interface "int-PE-1-PE-2" {
          interface-type point-to-point
          bfd-liveness {
          }
        }
        interface "system" {
        }
      }
    }
  }
}
    
```

```

# on PE-2:
configure {
  router "Base" {
    ospf 0 {
      admin-state enable
    }
  }
}
    
```

```

traffic-engineering true
area 0.0.0.0 {
    interface "int-PE-2-PE-1" {
        interface-type point-to-point
        bfd-liveness {
        }
    }
    interface "system" {
    }
}

```

The following commands verify that the BFD session for OSPF is operational between PE-1 and PE-2.

On PE-1:

```

[/]
A:admin@PE-1# show router bfd session
=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id          State      Tx Pkts   Rx Pkts
Rem Addr/Info/SdpId:VcId  Multipl   Tx Intvl  Rx Intvl
Protocols           Type      LAG Port   LAG ID
Loc Addr                LAG name
-----
int-PE-1-PE-2      Up        102       101
192.168.12.2       3         100       100
ospf2              iom       N/A       N/A
192.168.12.1
-----
No. of BFD sessions: 1
=====

```

On PE-2:

```

[/]
A:admin@PE-2# show router bfd session
=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id          State      Tx Pkts   Rx Pkts
Rem Addr/Info/SdpId:VcId  Multipl   Tx Intvl  Rx Intvl
Protocols           Type      LAG Port   LAG ID
Loc Addr                LAG name
-----
int-PE-2-PE-1      Up        69        69
192.168.12.1       3         100       100
ospf2              iom       N/A       N/A
192.168.12.2
-----
No. of BFD sessions: 1
=====

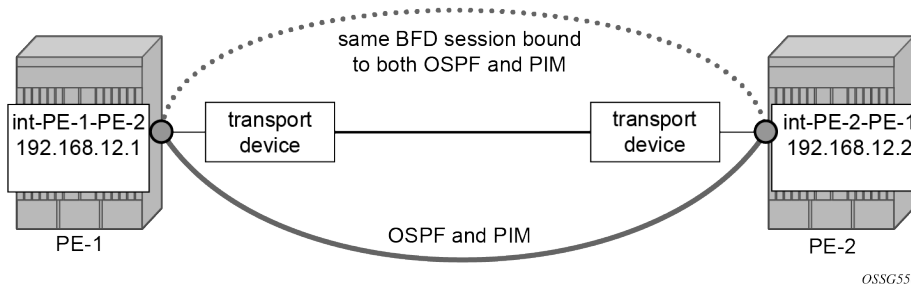
```

BFD for PIM

The PIM implementation uses an interior gateway protocol (IGP) in order to determine its reverse path forwarding (RPF) tree, so the BFD configuration to support PIM requires the BFD configuration of both the IGP protocol and the PIM protocol. In this example, the IGP protocol is OSPF and that the initial configuration is as described in the section [BFD for OSPF](#).

[Figure 21: BFD for OSPF and PIM](#) shows the topology. BFD is configured and enabled for PIM on the same interfaces that are configured with BFD for OSPF.

Figure 21: BFD for OSPF and PIM



The following commands enable BFD on the PIM interfaces on PE-1 and PE-2.

```
# on PE-1:
configure {
  router "Base" {
    pim {
      admin-state enable
      interface "int-PE-1-PE-2" {
        bfd-liveness {
          ipv4 true
        }
      }
    }
  }
}
```

```
# on PE-2:
configure {
  router "Base" {
    pim {
      admin-state enable
      interface "int-PE-2-PE-1" {
        bfd-liveness {
          ipv4 true
        }
      }
    }
  }
}
```

The following commands show that the BFD session is operational for OSPF and PIM between PE-1 and PE-2.

```
[/]
A:admin@PE-1# show router bfd session

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path   pp = Protecting path
=====
```

```

BFD Session
=====
Session Id          State      Tx Pkts  Rx Pkts
Rem Addr/Info/SdpId:VcId  Multipl  Tx Intvl  Rx Intvl
Protocols          Type      LAG Port  LAG ID
Loc Addr                    LAG name
-----
int-PE-1-PE-2      Up        661      660
192.168.12.2      3         100      100
ospf2 pim         iom       N/A      N/A
192.168.12.1
-----
No. of BFD sessions: 1
=====

```

```

[/]
A:admin@PE-2# show router bfd session

Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path

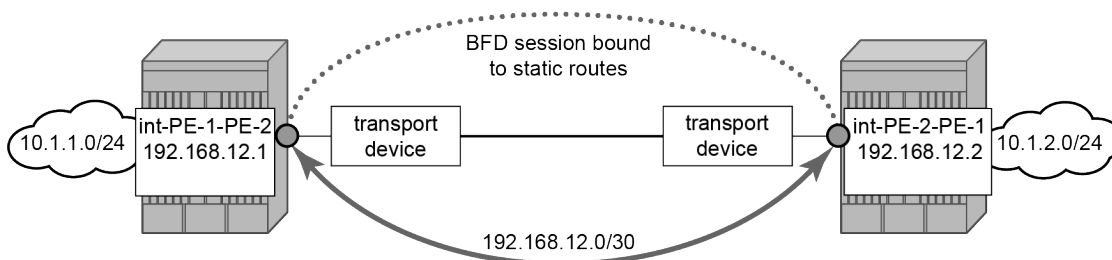
BFD Session
=====
Session Id          State      Tx Pkts  Rx Pkts
Rem Addr/Info/SdpId:VcId  Multipl  Tx Intvl  Rx Intvl
Protocols          Type      LAG Port  LAG ID
Loc Addr                    LAG name
-----
int-PE-2-PE-1      Up        637      637
192.168.12.1      3         100      100
ospf2 pim         iom       N/A      N/A
192.168.12.2
-----
No. of BFD sessions: 1
=====

```

BFD for static routes

In this section, BFD is applied to static routes between PE-1 and PE-2. [Figure 22: BFD for static routes](#) shows the topology.

Figure 22: BFD for static routes



OSSG559

The base level BFD is already configured on PE-1 and PE-2, as described in the [BFD base parameter configuration and troubleshooting](#) section.

The following commands configure static routes toward the remote networks in PE-1 and PE-2 using the BFD interfaces as next hop. BFD is enabled on the the next hop interfaces.



Note: BFD cannot be enabled if the next hop is indirect or the **black-hole** keyword is specified.

```
# on PE-1:
configure {
  router "Base" {
    static-routes {
      route 10.1.2.0/24 route-type unicast {
        next-hop "192.168.12.2" {
          admin-state enable
          bfd-liveness true
        }
      }
    }
  }
}
```

```
# on PE-2:
configure {
  router "Base" {
    static-routes {
      route 10.1.1.0/24 route-type unicast {
        next-hop "192.168.12.1" {
          admin-state enable
          bfd-liveness true
        }
      }
    }
  }
}
```

The following commands show the static routes populated in the routing tables on PE-1 and PE-2.

```
*A:PE-1# show router route-table protocol static

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type   Proto   Age           Pref
  Next Hop[Interface Name]                       Metric
-----
10.1.2.0/24                                       Remote Static   00h00m04s    5
  192.168.12.2                                     1
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

```
*A:PE-2# show router route-table protocol static

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type   Proto   Age           Pref
  Next Hop[Interface Name]                       Metric
-----
10.1.1.0/24                                       Remote Static   00h00m03s    5
  192.168.12.1                                     1
-----
```

```

-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

The following commands show the BFD session status on PE-1 and PE-2.

```

[/]
A:admin@PE-1# show router bfd session

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id                               State      Tx Pkts   Rx Pkts
Rem Addr/Info/SdpId:VcId                 Multipl   Tx Intvl  Rx Intvl
Protocols                                 Type      LAG Port  LAG ID
Loc Addr                                  LAG name
-----
int-PE-1-PE-2                             Up        431       427
192.168.12.2                               3         100       100
static                                    iom      N/A       N/A
192.168.12.1
-----
No. of BFD sessions: 1
=====

```

```

[/]
A:admin@PE-2# show router bfd session

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id                               State      Tx Pkts   Rx Pkts
Rem Addr/Info/SdpId:VcId                 Multipl   Tx Intvl  Rx Intvl
Protocols                                 Type      LAG Port  LAG ID
Loc Addr                                  LAG name
-----
int-PE-2-PE-1                             Up        399       398
192.168.12.1                               3         100       100
static                                    iom      N/A       N/A
192.168.12.2
-----
No. of BFD sessions: 1
=====

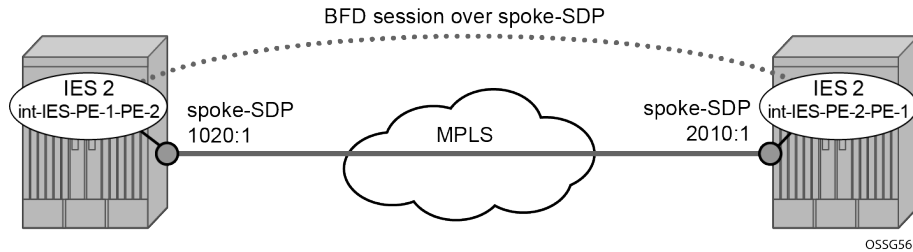
```

BFD for IES

The goal of this section is to configure BFD for an IES service over a spoke SDP.

The IES service is configured on PE-1 and PE-2, and their interfaces are connected by spoke SDPs. [Figure 23: BFD for IES over spoke SDP](#) shows the topology.

Figure 23: BFD for IES over spoke SDP



In this scenario, BFD is run between the IES interfaces independent of the SDP or LSP paths.

The following commands on PE-1 and PE-2 configure an IES service and add the IES interfaces to the OSPF area domain. BFD is not configured yet.

```
# on PE-1:
configure {
  service {
    sdp 1020 {
      admin-state enable
      delivery-type mpls
      sr-isis true
      far-end {
        ip-address 192.0.2.2
      }
    }
  }
  ies "IES-2" {
    admin-state enable
    service-id 2
    customer "1"
    interface "int-IES-PE-1-PE-2" {
      spoke-sdp 1020:1 {
      }
      ipv4 {
        primary {
          address 192.168.12.5
          prefix-length 30
        }
      }
    }
  }
}
router "Base" {
  ospf 0 {
    area 0.0.0.0 {
      interface "int-IES-PE-1-PE-2"
    }
  }
}
```

```
# on PE-2:
configure {
  service {
    sdp 2010 {
      admin-state enable
      delivery-type mpls
```

```

        sr-isis true
        far-end {
            ip-address 192.0.2.1
        }
    }
    ies "IES-2" {
        admin-state enable
        service-id 2
        customer "1"
        interface "int-IES-PE-2-PE-1" {
            spoke-sdp 2010:1 {
            }
            ipv4 {
                primary {
                    address 192.168.12.6
                    prefix-length 30
                }
            }
        }
    }
}
router "Base" {
    ospf 0 {
        area 0.0.0.0 {
            interface "int-IES-PE-2-PE-1" {
            }
        }
    }
}

```

The following commands verify that OSPF and the services are up on both routers.

On PE-1:

```

[/]
A:admin@PE-1# show service id 2 base
=====
Service Basic Information
=====
Service Id       : 2                Vpn Id          : 0
Service Type    : IES
MACSec enabled  : no
Name            : IES-2
Description     : (Not Specified)
Customer Id     : 1                Creation Origin  : manual
Last Status Change: 04/20/2023 16:24:00
Last Mgmt Change  : 04/20/2023 16:23:45
Admin State     : Up              Oper State      : Up
SAP Count       : 0              SDP Bind Count  : 1

-----
Service Access & Destination Points
-----
Identifier                               Type      AdmMTU  OprMTU  Adm  Opr
-----
sdp:1020:1 S(192.0.2.2)                  Spok     0       8910   Up   Up
=====

```

```

[/]
A:admin@PE-1# show router ospf neighbor
=====
Rtr Base OSPFv2 Instance 0 Neighbors
=====

```



```

=====
Interface-Name          Rtr Id      State      Pri  RetxQ  TTL
Area-Id
-----
int-PE-1-PE-2          192.0.2.2   Full       1    0       39
 0.0.0.0
int-IES-PE-1-PE-2     192.0.2.2   Full       1    0       30
 0.0.0.0
-----
No. of Neighbors: 2
=====

```

On PE-2:

```

[/]
A:admin@PE-2# show service id 2 base

=====
Service Basic Information
=====
Service Id       : 2                Vpn Id          : 0
Service Type    : IES
MACSec enabled  : no
Name            : IES-2
Description     : (Not Specified)
Customer Id     : 1                Creation Origin  : manual
Last Status Change: 04/20/2023 16:23:59
Last Mgmt Change : 04/20/2023 16:23:53
Admin State     : Up                Oper State      : Up
SAP Count       : 0                SDP Bind Count  : 1

-----
Service Access & Destination Points
-----
Identifier              Type      AdmMTU  OprMTU  Adm  Opr
-----
sdp:2010:1 S(192.0.2.1) Spok      0       8910   Up   Up
=====

```

```

[/]
A:admin@PE-2# show router ospf neighbor

=====
Rtr Base OSPFv2 Instance 0 Neighbors
=====
Interface-Name          Rtr Id      State      Pri  RetxQ  TTL
Area-Id
-----
int-PE-2-PE-1          192.0.2.1   Full       1    0       32
 0.0.0.0
int-IES-PE-2-PE-1     192.0.2.1   Full       1    0       34
 0.0.0.0
-----
No. of Neighbors: 2
=====

```

The following commands on PE-1 and PE-2 configure BFD on the IES interfaces and enable BFD on the OSPF interfaces.

```

# on PE-1:
configure {

```

```

service {
  ies "IES-2" {
    interface "int-IES-PE-1-PE-2" {
      ipv4 {
        bfd {
          admin-state enable
        }
      }
    }
  }
}
router "Base" {
  ospf 0 {
    area 0.0.0.0 {
      interface "int-IES-PE-1-PE-2" {
        bfd-liveness {
        }
      }
    }
  }
}

```

```

# on PE-2:
configure {
  service {
    ies "IES-2" {
      interface "int-IES-PE-2-PE-1" {
        ipv4 {
          bfd {
            admin-state enable
          }
        }
      }
      info
    }
  }
  router "Base" {
    ospf 0 {
      area 0.0.0.0 {
        interface "int-IES-PE-2-PE-1" {
          bfd-liveness {
          }
        }
      }
    }
  }
}

```

A centralized BFD session is created for BFD over spoke SDP even if a physical link exists between the two nodes. This centralized BFD session is created because the spoke SDP is terminated at the CPM. This is also the case for BFD running over LAG bundles.

The *central* type is used when BFD packets are completely generated and processed by software on the CPM. The *cpm-np* type is used when BFD packets are generated and processed with hardware assistance on the CPM. The following output shows that BFD session type is **cpm-np**.

```

[/]
A:admin@PE-1# show router bfd session

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path   pp = Protecting path
=====
BFD Session
=====
Session Id                               State      Tx Pkts   Rx Pkts

```

Rem Addr/Info/SdpId:VcId Protocols Loc Addr	Multipl Type	Tx Intvl LAG Port	Rx Intvl LAG ID LAG name
int-IES-PE-1-PE-2 192.168.12.6 ospf2 192.168.12.5	Up 3 cpm-np	N/A 1000 N/A	N/A 1000 N/A
No. of BFD sessions: 1			

```
[/]
A:admin@PE-2# show router bfd session

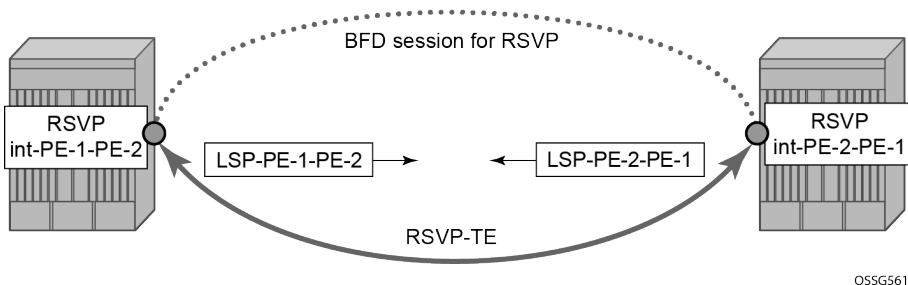
=====
Legend:
Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
wp = Working path   pp = Protecting path
=====
BFD Session
=====
Session Id          State      Tx Pkts  Rx Pkts
Rem Addr/Info/SdpId Multipl   Tx Intvl Rx Intvl
Protocols           Type     LAG Port  LAG ID
Loc Addr                               LAG name
-----
int-IES-PE-2-PE-1  Up        N/A      N/A
192.168.12.5      3         1000    1000
ospf2              cpm-np    N/A      N/A
192.168.12.6
-----
No. of BFD sessions: 1
=====
```

The transmitted and received packet counters are not included in the preceding **show** commands. BFD sessions of the **cpm-np** type are handled by hardware. The hardware does not have transmitted or received packet counters. In contrast, IOM BFD sessions are handled by the CPU of the IOM, so the packets are counted. Likewise, BFD sessions of type central are handled by the CPU of the CPM and the packets are counted.

BFD for RSVP

The goal of this section is to configure BFD between two RSVP interfaces configured in two SR OS nodes. [Figure 24: BFD for RSVP](#) shows the topology for this scenario.

Figure 24: BFD for RSVP



BFD is configured on the interfaces between PE-1 and PE-2 as described in [BFD base parameter configuration and troubleshooting](#).

The following commands on PE-1 and PE-2 configure the paths, the LSPs, and the interfaces within MPLS and RSVP.

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      admin-state enable
      interface "int-PE-1-PE-2" {
        admin-state enable
      }
      interface "system" {
        admin-state enable
      }
      path "empty" {
        admin-state enable
      }
      lsp "LSP-PE-1-PE-2" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.2
        path-computation-method local-cspf
        primary "empty" {
        }
      }
    }
  }
  rsvp {
    admin-state enable
    interface "int-PE-1-PE-2" {
    }
  }
}
```

```
# on PE-2:
configure {
  router "Base" {
    mpls {
      admin-state enable
      interface "int-PE-2-PE-1" {
        admin-state enable
      }
      interface "system" {
        admin-state enable
      }
      path "empty" {
        admin-state enable
      }
      lsp "LSP-PE-2-PE-1" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.1
        path-computation-method local-cspf
        primary "empty" {
        }
      }
    }
  }
  rsvp {
    admin-state enable
    interface "int-PE-2-PE-1" {
    }
  }
}
```

```
}

```

The following command on PE-1 verifies that the RSVP sessions are up.

```
[/]
A:admin@PE-1# show router rsvp session

=====
RSVP Sessions
=====
RSVP Session Name
  From           To           Tunnel ID   LSP ID      State
-----
LSP-PE-2-PE-1::empty
192.0.2.2       192.0.2.1   1           38912       Up
LSP-PE-1-PE-2::empty
192.0.2.1       192.0.2.2   1           12800       Up
-----
Sessions : 2
=====
```

The following commands on PE-1 and PE-2 enable BFD on the RSVP interfaces.

```
# on PE-1:
configure {
  router "Base" {
    rsvp {
      interface "int-PE-1-PE-2"
        bfd-liveness true
    }
  }
}
```

```
# on PE-2:
configure {
  router "Base" {
    rsvp {
      interface "int-PE-2-PE-1"
        bfd-liveness true
    }
  }
}
```

The following commands verify that the BFD session is operational between PE-1 and PE-2.

On PE-1:

```
[/]
A:admin@PE-1# show router bfd session

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id           State      Tx Pkts  Rx Pkts
  Rem Addr/Info/SdpId:VcId  Multipl  Tx Intvl  Rx Intvl
  Protocols              Type     LAG Port   LAG ID
  Loc Addr                LAG name
-----
int-PE-1-PE-2           Up         315       284
```

```

192.168.12.2          3          100          100
rsvp                 iom          N/A          N/A
192.168.12.1
-----
No. of BFD sessions: 1
=====

```

On PE-2:

```

[/]
A:admin@PE-2# show router bfd session

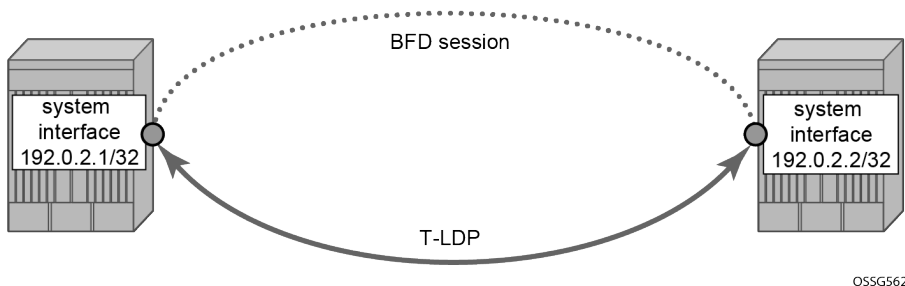
=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path   pp = Protecting path
=====
BFD Session
=====
Session Id          State      Tx Pkts   Rx Pkts
Rem Addr/Info/SdpId:VcId  Multipl  Tx Intvl  Rx Intvl
Protocols           Type     LAG Port   LAG ID
Loc Addr
-----
int-PE-2-PE-1      Up        270       270
192.168.12.1       3        100       100
rsvp               iom      N/A       N/A
192.168.12.2
-----
No. of BFD sessions: 1
=====

```

BFD for T-LDP

BFD tracking of an LDP session associated with a T-LDP adjacency allows for faster detection of the liveness of the session by registering the transport address of an LDP session with a BFD session. [Figure 25: BFD for T-LDP](#) shows the topology.

Figure 25: BFD for T-LDP



The parameters used for the BFD session are configured under the loopback interface corresponding to the LSR-ID. By default, the LSR-ID matches the system interface address.

```

# on PE-1, PE-2:
configure {
  router "Base" {
    interface "system" {

```

```

    ipv4 {
        bfd {
            admin-state enable
            transmit-interval 3000
            receive 3000
        }
    }

```

The loopback interface can be used to source BFD sessions to many peers in the network.

When using BFD over other links with the ability to reroute, such as spoke-SDPs, the interval and multiplier values configuring BFD should be set to allow sufficient time for the underlying network to reconverge before the associated BFD session expires. A general rule of thumb should be that the expiration time (interval * multiplier) is three times the convergence time for the IGP network between the two endpoints of the BFD session.

On PE-1 and PE-2, the following T-LDP session is established with BFD enabled.

```

# on PE-1:
configure {
    router "Base" {
        ldp {
            targeted-session {
                peer 192.0.2.2
                admin-state enable
                bfd-liveness true
            }
        }
    }
}

```

```

# on PE-2:
configure {
    router "Base" {
        ldp {
            targeted-session {
                peer 192.0.2.1
                admin-state enable
                bfd-liveness true
            }
        }
    }
}

```

By enabling BFD for a selected targeted session, the state of that session is tied to the state of the underlying BFD session between the two nodes.

The following commands on PE-1 and PE-2 verify that the T-LDP session is up.

On PE-1:

```

[/]
A:admin@PE-1# show router ldp session ipv4

=====
LDP IPv4 Sessions
=====
Peer LDP Id      Adj Type  State      Msg Sent  Msg Recv  Up Time
-----
192.0.2.2:0     Targeted Established 238       237       0d 00:20:39
-----
No. of IPv4 Sessions: 1
=====

```

On PE-2:

```
[/]
A:admin@PE-2# show router ldp session ipv4

=====
LDP IPv4 Sessions
=====
Peer LDP Id          Adj Type  State          Msg Sent  Msg Recv  Up Time
-----
192.0.2.1:0         Targeted  Established    235       237       0d 00:20:33
-----
No. of IPv4 Sessions: 1
=====
```

The following commands on PE-1 and PE-2 show that the BFD session is up.

On PE-1:

```
[/]
A:admin@PE-1# show router bfd session

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id          State      Tx Pkts      Rx Pkts
Rem Addr/Info/SdpId:VcId  Multipl   Tx Intvl    Rx Intvl
Protocols           Type      LAG Port     LAG ID
Loc Addr                               LAG name
-----
system              Up        N/A          N/A
192.0.2.2           3        3000        3000
Ldp                  cpm-np   N/A          N/A
192.0.2.1
-----
No. of BFD sessions: 1
=====
```

On PE-2:

```
[/]
A:admin@PE-2# show router bfd session

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id          State      Tx Pkts      Rx Pkts
Rem Addr/Info/SdpId:VcId  Multipl   Tx Intvl    Rx Intvl
Protocols           Type      LAG Port     LAG ID
Loc Addr                               LAG name
-----
system              Up        N/A          N/A
192.0.2.1           3        3000        3000
Ldp                  cpm-np   N/A          N/A
-----
```



```

192.0.2.2
-----
No. of BFD sessions: 1
=====

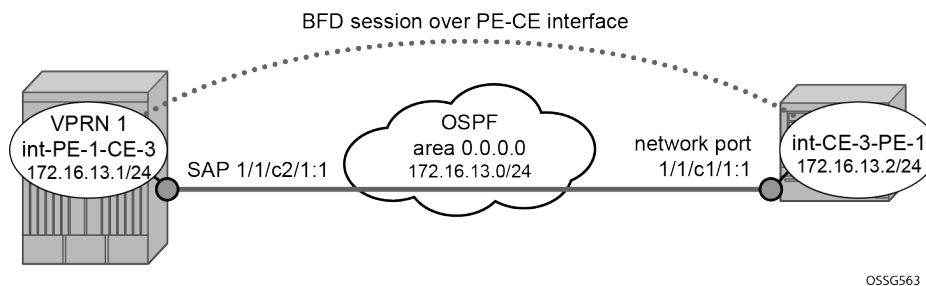
```

When the T-LDP session comes up, a centralized **cpm-np** BFD session is always created even if the local interface has a direct link to the peer.

BFD for OSPF PE-CE adjacencies

BFD for OSPF PE-CE adjacencies extends BFD support to OSPF within a **vprn** context when OSPF is used as the PE-CE protocol. [Figure 26: BFD for OSPF PE-CE interfaces](#) shows the topology used in this section.

Figure 26: BFD for OSPF PE-CE interfaces



OSSG563

On PE-1, the following VPRN configuration includes service interface int-PE-1-CE-1 with BFD parameters.

```

# on PE-1:
configure {
  service {
    vprn "VPRN-1" {
      admin-state enable
      service-id 1
      customer "1"
      interface "int-PE-1-CE-3" {
        ipv4 {
          bfd {
            admin-state enable
          }
          primary {
            address 172.16.13.1
            prefix-length 24
          }
        }
        sap 1/1/c2/1:1 {
        }
      }
    }
    ospf 0 {
      admin-state enable
      area 0.0.0.0 {
        interface "int-PE-1-CE-3" {
          bfd-liveness {
          }
        }
      }
    }
  }
}

```

On CE-3, the following configures the router interface int-CE-3-PE-1 with BFD parameters. BFD is enabled on this interfaces that is added to the OSPF area 0.0.0.0 domain.

```
# on CE-3:
configure {
  router "Base" {
    interface "int-CE-3-PE-1" {
      port 1/1/c1/1:1
      ipv4 {
        bfd {
          admin-state enable
        }
        primary {
          address 172.16.13.2
          prefix-length 24
        }
      }
    }
    interface "system" {
      ipv4 {
        primary {
          address 192.0.2.3
          prefix-length 32
        }
      }
    }
    ospf 0 {
      admin-state enable
      area 0.0.0.0 {
        interface "int-CE-3-PE-1" {
          bfd-liveness {
          }
        }
      }
    }
  }
}
```

The following command shows that the OSPF adjacency is up.

On PE-1:

```
[/]
A:admin@PE-1# show router 1 ospf neighbor

=====
Rtr vprn1 OSPFv2 Instance 0 Neighbors
=====
Interface-Name          Rtr Id      State      Pri  RetxQ  TTL
Area-Id
-----
int-PE-1-CE-3          192.0.2.3   Full       1    0      31
0.0.0.0
-----
No. of Neighbors: 1
=====
```

On CE-3:

```
[/]
A:admin@CE-3# show router ospf neighbor

=====
Rtr Base OSPFv2 Instance 0 Neighbors
```

```

=====
Interface-Name          Rtr Id      State      Pri  RetxQ  TTL
Area-Id
-----
int-CE-3-PE-1          192.0.2.1  Full       1    0      38
0.0.0.0
-----
No. of Neighbors: 1
=====

```

The following commands show that the BFD session is up in both PE-1 and CE-3.

```

[/]
A:admin@PE-1# show router 1 bfd session

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id          State      Tx Pkts  Rx Pkts
Rem Addr/Info/SdpId:VcId  Multipl  Tx Intvl  Rx Intvl
Protocols          Type     LAG Port  LAG ID
Loc Addr                    LAG name
-----
int-PE-1-CE-3      Up        1788     1782
172.16.13.2        3         100      100
ospf2              iom       N/A      N/A
172.16.13.1
-----
No. of BFD sessions: 1
=====

```

```

[/]
A:admin@CE-3# show router bfd session

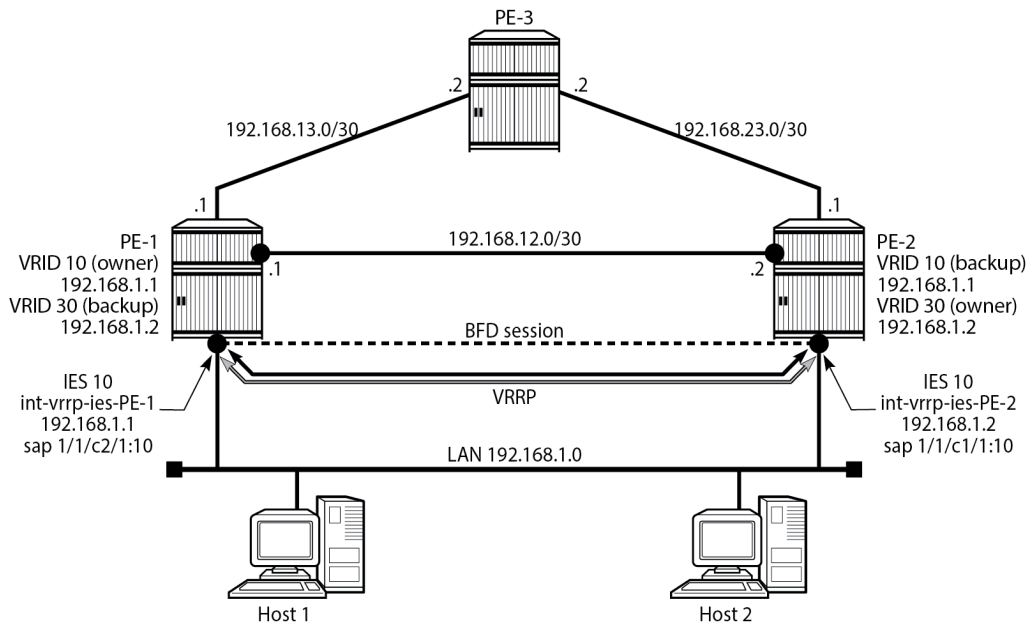
=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id          State      Tx Pkts  Rx Pkts
Rem Addr/Info/SdpId:VcId  Multipl  Tx Intvl  Rx Intvl
Protocols          Type     LAG Port  LAG ID
Loc Addr                    LAG name
-----
int-CE-3-PE-1      Up        1996     1999
172.16.13.1        3         100      100
ospf2              iom       N/A      N/A
172.16.13.2
-----
No. of BFD sessions: 1
=====

```

BFD for VRRP

This feature assigns a BFD session to provide a heart-beat mechanism for the VRRP instance. There can be only one BFD session assigned to any VRRP instance, but there can be multiple VRRP sessions using the same BFD session. [Figure 27: BFD for VRRP](#) shows the topology for this section.

Figure 27: BFD for VRRP



25511

Host 1 and host 2 are connected to LAN subnet 192.168.1.0/24. PE-1 and PE-2 are connected to the LAN subnet by IES or VPRN services. In the following example, IES 10 is created on PE-1 and PE-2 and BFD parameters are configured on the IES interface.

```
# on PE-1:
configure {
  service {
    ies "IES-10" {
      admin-state enable
      service-id 10
      customer "1"
      interface "int-vrrp-ies-PE-1" {
        mac 00:00:5e:00:53:01
        sap 1/1/c2/1:10 {
        }
      }
      ipv4 {
        bfd {
          admin-state enable
          multiplier 10
        }
        primary {
          address 192.168.1.1
          prefix-length 24
        }
      }
    }
  }
}
```

```
# on PE-2:
configure {
```

```

service {
  ies "IES-10" {
    admin-state enable
    service-id 10
    customer "1"
    interface "int-vrrp-ies-PE-2" {
      mac 00:00:5e:00:53:02
      sap 1/1/cl/1:10 {
      }
      ipv4 {
        bfd {
          admin-state enable
          multiplier 10
        }
        primary {
          address 192.168.1.2
          prefix-length 24
        }
      }
    }
  }
}

```

The following command on PE-1 verifies that the IES service "IES-10" is operational:

```

[/]
A:admin@PE-1# show service service-using ies
=====
Services [ies]
=====
ServiceId   Type      Adm  Opr  CustomerId Service Name
-----
2           IES       Up   Up   1          IES-2
10         IES       Up   Up  1          IES-10
2147483648  IES       Up   Down 1          _tmnx_InternalIesService
-----
Matching Services : 3
-----
=====

```

The following command on PE-1 verifies the connectivity to the remote interface IP address 192.168.1.2:

```

[/]
A:admin@PE-1# ping 192.168.1.2 interval 0.1 output-format summary
PING 192.168.1.2 56 data bytes
!!!!
---- 192.168.1.2 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 3.97ms, avg = 4.03ms, max = 4.12ms, stddev = 0.068ms

```

On PE-1 and PE-2, VRRP is enabled on the IES interface that connects to the 192.168.1.0/24 subnet. In this section, the configurations are shown for the VRRP owner mode for primary but any other scenario for VRRP can be configured (non owner mode for primary). In the following example, two VRRP instances are created on the 192.168.1.0/24 subnet:

```

VRID = 10  Owner   = PE-1
          Backup = PE-2
          VRRP IP = 192.168.1.1
VRID = 30  Owner   = PE-2
          Backup = PE-1
          VRRP IP = 192.168.1.2

```

Host 1 is configured with default gateway 192.168.1.1, and host 2 is configured with default gateway 192.168.1.2.

VRRP 10 and VRRP 30 are configured on the IES interface connected to the 192.168.1.0/24 subnet. To bind the VRRP instances with a BFD session, BFD liveness is enabled on this IES interface for VRRP 10 and VRRP 30. The configuration on PE-1 is as follows:

```
# on PE-1:
configure {
  service {
    ies "IES-10" {
      admin-state enable
      service-id 10
      customer "1"
      interface "int-vrrp-ies-PE-1" {
        mac 00:00:5e:00:53:01
        sap 1/1/c2/1:10 {
        }
        ipv4 {
          bfd {
            admin-state enable
            multiplier 10
          }
          primary {
            address 192.168.1.1
            prefix-length 24
          }
          vrrp 10 {
            backup [192.168.1.1]
            owner true
            bfd-liveness {
              dest-ip 192.168.1.2
              service-name "IES-10"
              interface-name "int-vrrp-ies-PE-1"
            }
          }
          vrrp 30 {
            backup [192.168.1.2]
            ping-reply true
            telnet-reply true
            ssh-reply true
            bfd-liveness {
              dest-ip 192.168.1.2
              service-name "IES-10"
              interface-name "int-vrrp-ies-PE-1"
            }
          }
        }
      }
    }
  }
}
```

The configuration on PE-2 is as follows:

```
# on PE-2:
configure {
  service {
    ies "IES-10" {
      admin-state enable
      service-id 10
      customer "1"
      interface "int-vrrp-ies-PE-2" {
        mac 00:00:5e:00:53:02
        sap 1/1/c1/1:10 {
        }
      }
    }
  }
}
```

```

    ipv4 {
        bfd {
            admin-state enable
            multiplier 10
        }
        primary {
            address 192.168.1.2
            prefix-length 24
        }
        vrrp 10 {
            backup [192.168.1.1]
            ping-reply true
            telnet-reply true
            ssh-reply true
            bfd-liveness {
                dest-ip 192.168.1.1
                service-name "IES-10"
                interface-name "int-vrrp-ies-PE-2"
            }
        }
        vrrp 30 {
            backup [192.168.1.2]
            owner true
            bfd-liveness {
                dest-ip 192.168.1.1
                service-name "IES-10"
                interface-name "int-vrrp-ies-PE-2"
            }
        }
    }
}

```

The parameters used for the BFD are set by the BFD command under the IP interface. Unlike the previous scenarios, the user can configure **bfd-liveness** for VRRP, enabling the BFD session, even if the specified interface has not been configured with BFD parameters (**ipv4>bfd>admin-state enable**).

If the BFD parameters have not been configured yet, the BFD session will be initiated only after configuring the BFD parameters (**ipv4>bfd>admin-state enable**).

```

# on PE-1:
configure {
    service {
        ies "IES-10" {
            interface "int-vrrp-ies-PE-1" {
                ipv4 {
                    bfd {
                        admin-state enable
                        # transmit-interval 100      # default
                        # receive 100                # default
                        multiplier 10
                    }
                }
            }
        }
    }
}

```

```

# on PE-2:
configure {
    service {
        ies "IES-10" {
            interface "int-vrrp-ies-PE-2" {
                ipv4 {
                    bfd {
                        admin-state enable
                        # transmit-interval 100      # default
                        # receive 100                # default
                        multiplier 10
                    }
                }
            }
        }
    }
}

```

```
}

```

The following command on PE-1 shows that the BFD session is up:

```
[/]
A:admin@PE-1# show router bfd session src 192.168.1.1 detail

=====
BFD Session
=====
Remote Address : 192.168.1.2
Local Address  : 192.168.1.1
Admin State  : Up                                Oper State    : Up
Protocols      : vrrp
Rx Interval    : 100                                Tx Interval   : 100
Multiplier    : 10                                 Echo Interval : 0
Recd Msgs     : 2171                               Sent Msgs    : 2185
Up Time       : 0d 00:02:48                         Up Transitions : 1
Last Down Time : 0d 00:00:23                       Down Transitions : 0
                                                    Version Mismatch : 0

Forwarding Information

Local Discr    : 8                                Local State   : Up
Local Diag     : 0 (None)                        Local Mode    : Async
Local Min Tx   : 100                             Local Mult    : 10
Last Sent     : 04/20/2023 16:36:33             Local Min Rx  : 100
Type          : iom
Remote Discr   : 7                                Remote State  : Up
Remote Diag    : 0 (None)                        Remote Mode   : Async
Remote Min Tx : 100                             Remote Mult   : 10
Remote C-flag  : 1
Last Recv     : 04/20/2023 16:36:33             Remote Min Rx : 100
=====
=====
```

This session is shared by all the VRRP instances configured between the specified interfaces.

When BFD is configured in a VRRP instance, the following command gives details of BFD related to every instance:

```
[/]
A:admin@PE-1# show router vrrp instance interface "int-vrrp-ies-PE-1"

=====
VRRP Instances for interface "int-vrrp-ies-PE-1"
=====
-----
VRID 10
-----
Owner          : Yes                                VRRP State    : Master
Primary IP of Master: 192.168.1.1 (Self)
Primary IP     : 192.168.1.1                        Standby-Forwarding: Disabled
VRRP Backup Addr : 192.168.1.1
Admin State     : Up                                Oper State    : Up
Up Time        : 04/20/2023 16:33:22             Virt MAC Addr  : 00:00:5e:00:01:0a
Auth Type      : None
Config Mesg Intvl : 1                             In-Use Mesg Intvl : 1
Base Priority   : 255                              In-Use Priority  : 255
Init Delay     : 0                                 Init Timer Expires: 0.000 sec
Creation State  : Active
```



```

-----
BFD Interface
-----
Service ID       : None
Service Name     : IES-10
Interface Name   : int-vrrp-ies-PE-1
Src IP           : 192.168.1.1
Dst IP           : 192.168.1.2
Session Oper State : connected
-----

Master Information
-----
Primary IP of Master: 192.168.1.1 (Self)
Addr List Mismatch : No           Master Priority : 255
Master Since       : 04/20/2023 16:33:22
-----

Masters Seen (Last 32)
-----
Primary IP of Master  Last Seen           Addr List Mismatch  Msg Count
-----
192.168.1.1          04/20/2023 16:33:22  No                   0
-----

Statistics
-----
Become Master       : 1           Master Changes     : 1
Adv Sent            : 199          Adv Received       : 0
Pri Zero Pkts Sent : 0           Pri Zero Pkts Rcvd: 0
Preempt Events     : 0           Preempted Events  : 0
Mesg Intvl Discards : 0          Mesg Intvl Errors : 0
Addr List Discards : 0           Addr List Errors  : 0
Auth Type Mismatch : 0           Auth Failures     : 0
Invalid Auth Type  : 0           Invalid Pkt Type  : 0
IP TTL Errors      : 0           Pkt Length Errors : 0
Total Discards     : 0
-----

VRID 30
-----
Owner              : No           VRRP State         : Backup
Primary IP of Master: 192.168.1.2 (Other)
Primary IP         : 192.168.1.1   Standby-Forwarding: Disabled
VRRP Backup Addr  : 192.168.1.2
Admin State        : Up            Oper State          : Up
Up Time           : 04/20/2023 16:33:22 Virt MAC Addr      : 00:00:5e:00:01:1e
Auth Type         : None
Config Mesg Intvl : 1             In-Use Mesg Intvl : 1
Master Inherit Intvl: No
Base Priority      : 100           In-Use Priority    : 100
Policy ID         : n/a           Preempt Mode       : Yes
Ping Reply        : Yes           Telnet Reply       : Yes
Ntp Reply         : No
SSH Reply         : Yes           Traceroute Reply  : No
Init Delay        : 0             Init Timer Expires: 0.000 sec
Creation State    : Active
-----

BFD Interface
-----
Service ID       : None
Service Name     : IES-10
Interface Name   : int-vrrp-ies-PE-1

```

```

Src IP           : 192.168.1.1
Dst IP           : 192.168.1.2
Session Oper State : connected

-----
Master Information
-----
Primary IP of Master: 192.168.1.2 (Other)
Addr List Mismatch : No           Master Priority   : 255
Master Since       : 04/20/2023 16:33:38
Master Down Interval: 3.609 sec (Expires in 2.600 sec)

-----
Masters Seen (Last 32)
-----
Primary IP of Master  Last Seen           Addr List Mismatch  Msg Count
-----
192.168.1.1           04/20/2023 16:33:25  No                   0
192.168.1.2           04/20/2023 16:36:39  No                   183

-----
Statistics
-----
Become Master       : 1           Master Changes     : 2
Adv Sent            : 13          Adv Received       : 183
Pri Zero Pkts Sent : 0           Pri Zero Pkts Rcvd: 0
Preempt Events     : 0           Preempted Events  : 1
Mesg Intvl Discards : 0          Mesg Intvl Errors : 0
Addr List Discards : 0           Addr List Errors  : 0
Auth Type Mismatch : 0           Auth Failures     : 0
Invalid Auth Type  : 0           Invalid Pkt Type  : 0
IP TTL Errors      : 0           Pkt Length Errors : 0
Total Discards     : 0

=====

```

For troubleshooting, a configuration error is introduced for VRRP 10 in service "IES-10" on PE-1. In this example, the misconfiguration is that the IES service name "IES-10" is not declared in the **bfd-enable** command for VRRP 10:

```

# on PE-1:
configure {
  service {
    ies "IES-10" {
      interface "int-vrrp-ies-PE-1" {
        ipv4 {
          vrrp 10 {
            bfd-liveness {
              delete service-name
            }
          }
        }
      }
    }
  }
}

```

In this case, the BFD session between the two IP interfaces is operationally up but the command **show router vrrp instance interface <interface-name>** on PE-1 gives the following output regarding BFD for VRID 10:

```

[/]
A:admin@PE-1# show router vrrp instance interface "int-vrrp-ies-PE-1"

=====
VRRP Instances for interface "int-vrrp-ies-PE-1"
=====
-----

```

```

VRID 10
-----
Owner                : Yes                VRRP State          : Master
Primary IP of Master: 192.168.1.1 (Self)
Primary IP           : 192.168.1.1       Standby-Forwarding: Disabled
VRRP Backup Addr    : 192.168.1.1
Admin State        : Up                Oper State         : Up
Up Time              : 04/20/2023 16:33:22 Virt MAC Addr       : 00:00:5e:00:01:0a
Auth Type            : None
Config Mesg Intvl   : 1                  In-Use Mesg Intvl  : 1
Base Priority        : 255                 In-Use Priority     : 255
Init Delay          : 0                    Init Timer Expires : 0.000 sec
Creation State       : Active
-----
BFD Interface
-----
Service ID        : None
Interface Name   : int-vrrp-ies-PE-1
Src IP              :
Dst IP              : 192.168.1.2
Session Oper State : notConfigured
-----
---snip---

```

The session operational state and the service ID indicate that the service ID is not configured. To fix this, enable BFD with service name "IES-10" for VRRP instance 10:

```

# on PE-1:
configure {
  service {
    ies "IES-10" {
      interface "int-vrrp-ies-PE-1" {
        ipv4 {
          vrrp 10 {
            bfd-liveness {
              service-name "IES-10"
            }
          }
        }
      }
    }
  }
}

```

Conclusion

BFD is a light-weight protocol which provides rapid path failure detection between two systems. BFD is useful in situations where the physical network has numerous intervening devices which are not part of the Layer 3 network.

BFD is linked to a protocol state. For a BFD session to be established, the prerequisite condition is that the protocol to which the BFD is linked must be operationally active. Once the BFD session is established, the state of the protocol to which BFD is tied to is then determined based on the BFD session's state. This means that if the BFD session goes down, the corresponding protocol will be brought down.

In this chapter, several scenarios where BFD could be implemented have been described, including the configuration, show output, and troubleshooting hints.

LFA Policies Using OSPF as IGP

This chapter provides information about LFA policies using OSPF as IGP.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 12.0.R4, but the MD-CLI in the current edition corresponds to SR OS Release 23.3.R3.

Overview

Loopfree alternate (LFA) is a local control plane feature. When multiple LFAs exist, RFC 5286 chooses the LFA providing the best coverage of the failure cases. In general, this means that node LFA has preference above link LFA. In some deployments, however, this can lead to suboptimal LFA. For example, an aggregation router (typically using lower bandwidth links) protecting a core node or link (typically using high bandwidth links) is potentially undesirable.

For this reason, the operator wants to have more control in the LFA next hop selection algorithm. This is achieved by the introduction of LFA shortest path first (SPF) policies.

LFA policies can work in combination with IP fast reroute (FRR) and LDP FRR.

Implementation

The SR OS LFA policy implementation is built around the concept of **route-next-hop-policy** templates which are applied to IP interfaces. A route next hop policy template specifies criteria that influence the selection of an LFA backup next hop for either:

- a set of prefixes in a prefix list or
- a set of prefixes which resolve to a specific primary next hop

See RFC 7916 for further information. Two powerful methods which can be used as criteria inside a route next hop policy template are IP admin groups and IP shared risk link groups (SRLGs). IP admin group and IP SRLG criteria are applied before running the LFA next hop algorithm. IP admin groups and SRLGs work in a similar way as the MPLS admin groups and SRLGs.

For example, when one or more IP admin groups or SRLGs are applied to an IP interface, the same MPLS admin group and SRLG rules apply:

- IP interfaces which do not include one or more of the admin groups defined in the **include** statements are pruned before computing the LFA next hop.

- IP interfaces which belong to admin groups which have been explicitly excluded using the **exclude** statement are pruned before computing the LFA next hop.
- IP interfaces which belong to the SRLGs used by the primary next hop of a prefix are pruned before computing the LFA next hop.

For more information about MPLS admin groups, see chapter [RSVP Point-to-Point LSPs](#); for SRLGs, see chapter [Shared Risk Link Groups for RSVP-Based LSPs](#).

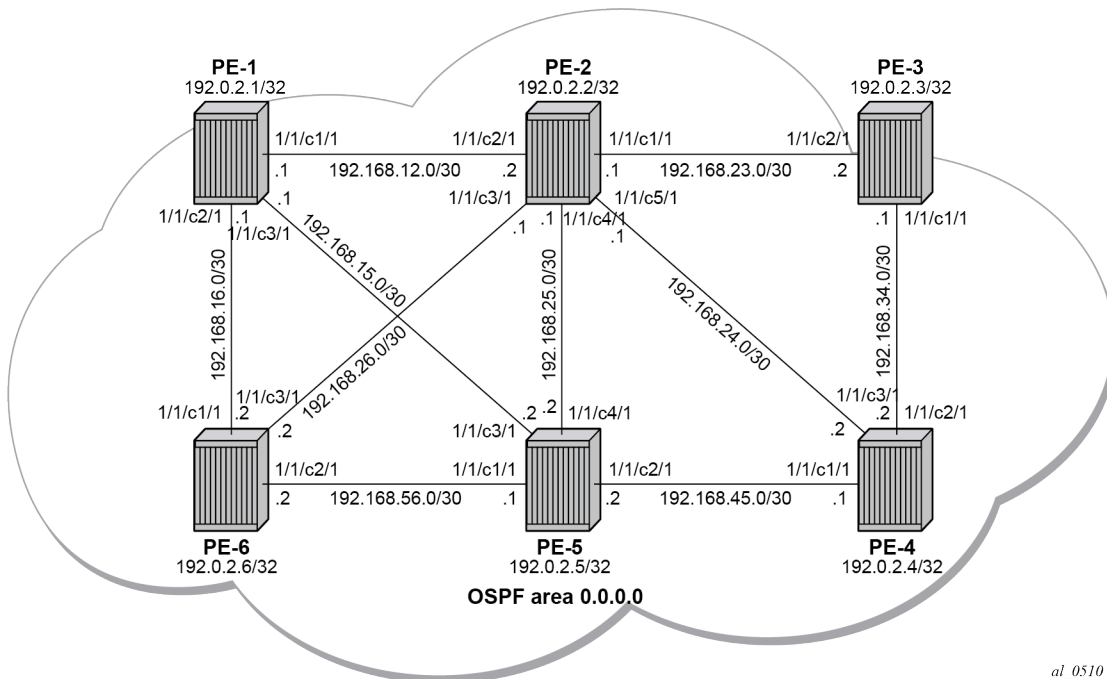
In the SR OS implementation, IP admin groups and SRLGs are locally significant, meaning they are not advertised by the IGP. Only the admin groups and SRLGs bound to an MPLS interface are advertised in TE link TLVs and sub-TLVs when the traffic engineering option is enabled in the IGP protocol. IES and VPRN interfaces do not have their attributes advertised in TE TLVs.

Other selection criteria which can be configured inside a route next hop template are protection type preference and next hop type preference. More details on these parameters are provided later in this chapter.

Configuration

[Example topology](#) shows the topology with six SR OS nodes. PE-2 will act as the point of local repair (PLR).

Figure 28: Example topology



1. Configure an IP/MPLS network with LDP FRR enabled on PE-2.

Because the focus is not on how to set up an IP/MPLS network, only summary bullets are provided.

- The system and IP interface addresses are configured according to [Figure 28: Example topology](#).
- OSPF area 0.0.0.0 is selected as the interior gateway protocol (IGP) to distribute routing information between all PEs. All OSPF interfaces are set up as type point-to-point to avoid running the

designated router/backup designated router (DR/BDR) election process. All links have an OSPF metric cost of 10, except for interface "int-PE-2-PE-5" on PE-2, which is configured with a metric of 20.

- Link LDP is enabled on all interfaces, which establishes a full mesh of LDP LSPs between all PE system interfaces. As an example, the tunnel table on PE-2 contains LDP tunnels to all the other PEs, as follows. The LDP LSP metric follows the IGP cost.

```
[/]
A:admin@PE-2# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner    Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.1/32         ldp     MPLS  65537    9     192.168.12.1  1
192.0.2.3/32         ldp     MPLS  65538    9     192.168.23.2  1
192.0.2.4/32         ldp     MPLS  65539    9     192.168.24.2  1
192.0.2.5/32         ldp     MPLS  65540    9     192.168.12.1  2
192.0.2.6/32         ldp     MPLS  65541    9     192.168.26.2  1
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====
```

- Enable LDP FRR on PE-2. This is a two-fold configuration command: the IGP needs to be triggered to do LFA next hop computation, and FRR needs to be enabled within the **ldp** context. First, LFA is enabled in OSPF on PE-2:

```
# on PE-2:
configure {
  router "Base" {
    ospf 0 {
      loopfree-alternate {
      }
    }
  }
}

[/]
A:admin@PE-2# show router ospf status | match LFA
LFA : Enabled
Remote-LFA : Disabled
Max PQ Cost (Remote-LFA) : 65535
Remote-LFA (node-protect) : Disabled
TI-LFA : Disabled
TI-LFA (node-protect) : Disabled
Mhp-LFA (IP-FRR) : Disabled
Mhp-LFA (SR) : Disabled
```

Remote LFA and topology-independent LFA (TI-LFA) can be enabled for segment routing, but this is beyond the scope of this chapter.

Second, LDP FRR is enabled on PE-2:

```
# on PE-2:
configure {
  router "Base" {
    ldp {
      fast-reroute {
      }
    }
  }
}
```

```

}

[/]
A:admin@PE-2# show router ldp status | match FRR
FRR           : Enabled           Mcast Upstream FRR   : Disabled
Mcast Upst ASBR FRR: Disabled

```

Multicast upstream FRR is for multicast LDP and is beyond the scope of this chapter.

After issuing these two CLI commands, the software precomputes both a primary and a backup next hop label forwarding entry (NHLFE) for each LDP forwarding equivalence class (FEC) in the network and downloads them into the IOM/IMM. The primary NHLFE corresponds to the label of the FEC received from the primary next hop as per standard LDP resolution of the FEC prefix in the routing table manager (RTM). The backup NHLFE corresponds to the label received for the same FEC from an LFA next hop. The **show router route-table alternative** command adds an LFA flag to the associated alternative next hop for a specific destination prefix. Other useful IGP related show commands are **show router ospf lfa-coverage** and **show router ospf routes alternative detail**.

```

[/]
A:admin@PE-2# show router route-table alternative
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type  Proto  Age           Pref
Next Hop[Interface Name]                       Metric
Alt-NextHop                                     Alt-
Metric
-----
192.0.2.1/32                                     Remote OSPF   00h02m41s    10
192.168.12.1                                     1
192.168.26.2 (LFA)                             2
192.0.2.2/32                                     Local  Local   00h02m42s    0
system                                           0
192.0.2.3/32                                     Remote OSPF   00h02m32s    10
192.168.23.2                                     1
192.168.24.2 (LFA)                             2
192.0.2.4/32                                     Remote OSPF   00h02m27s    10
192.168.24.2                                     1
192.168.23.2 (LFA)                             2
192.0.2.5/32                                     Remote OSPF   00h02m15s    10
192.168.12.1                                     2
192.168.24.2 (LFA)                             2
192.0.2.6/32                                     Remote OSPF   00h02m06s    10
192.168.26.2                                     1
192.168.12.1 (LFA)                             2
192.168.12.0/30                                  Local  Local   00h02m42s    0
int-PE-2-PE-1                                   0
192.168.15.0/30                                  Remote OSPF   00h02m41s    10
192.168.12.1                                     2
192.168.26.2 (LFA)                             3
192.168.16.0/30                                  Remote OSPF   00h02m41s    10
192.168.12.1                                     2
192.168.26.2 (LFA)                             3
192.168.23.0/30                                  Local  Local   00h02m42s    0
int-PE-2-PE-3                                   0
192.168.24.0/30                                  Local  Local   00h02m42s    0
int-PE-2-PE-4                                   0
192.168.25.0/30                                  Local  Local   00h02m42s    0
int-PE-2-PE-5                                   0
192.168.26.0/30                                  Local  Local   00h02m42s    0
int-PE-2-PE-6                                   0

```

```

192.168.34.0/30          Remote OSPF      00h02m32s  10
  192.168.23.2          2
  192.168.24.2 (LFA)   3
192.168.45.0/30        Remote OSPF      00h02m27s  10
  192.168.24.2          2
  192.168.23.2 (LFA)   3
192.168.56.0/30        Remote OSPF      00h02m06s  10
  192.168.26.2          2
  192.168.12.1 (LFA)   3
-----
No. of Routes: 16
Flags: n = Number of times nexthop is repeated
      Backup = BGP backup route
      LFA = Loop-Free Alternate nexthop
      S = Sticky ECMP requested
=====

```

Displaying the label forwarding information base (LFIB) on PE-2 shows the available alternate next hops that are displayed with the BU flag.

```

[/]
A:admin@PE-2# show router ldp bindings active prefixes ipv4

=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
  (IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static          (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop   (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding

=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix          Op
IngLbl          EgrLbl
EgrNextHop      EgrIf/LspId
-----
192.0.2.1/32    Push
  --           524287
192.168.12.1    1/1/c2/1:1000

192.0.2.1/32    Push
  --           524286BU
192.168.26.2    1/1/c3/1:1000

192.0.2.1/32    Swap
524286          524287
192.168.12.1    1/1/c2/1:1000

192.0.2.1/32    Swap
524286          524286BU
192.168.26.2    1/1/c3/1:1000

192.0.2.2/32    Pop
524287          --

```



```
--
192.0.2.3/32
--
192.168.23.2
192.0.2.3/32
--
192.168.24.2
192.0.2.3/32
524285
192.168.23.2
192.0.2.3/32
524285
192.168.24.2
192.0.2.4/32
--
192.168.24.2
192.0.2.4/32
--
192.168.23.2
192.0.2.4/32
524284
192.168.24.2
192.0.2.4/32
524284
192.168.23.2
192.0.2.5/32
--
192.168.12.1
192.0.2.5/32
--
192.168.24.2
192.0.2.5/32
524283
192.168.12.1
192.0.2.5/32
524283
192.168.24.2
192.0.2.6/32
--
192.168.26.2
192.0.2.6/32
--
192.168.12.1
192.0.2.6/32
524282
192.168.26.2
192.0.2.6/32
524282
```

```
--
Push
524287
1/1/c1/1:1000
Push
524286BU
1/1/c5/1:1000
Swap
524287
1/1/c1/1:1000
Swap
524286BU
1/1/c5/1:1000
Push
524287
1/1/c5/1:1000
Push
524284BU
1/1/c1/1:1000
Swap
524287
1/1/c5/1:1000
Swap
524284BU
1/1/c1/1:1000
Push
524283
1/1/c2/1:1000
Push
524283BU
1/1/c5/1:1000
Swap
524283
1/1/c2/1:1000
Swap
524283BU
1/1/c5/1:1000
Push
524287
1/1/c3/1:1000
Push
524282BU
1/1/c2/1:1000
Swap
524287
1/1/c3/1:1000
Swap
524282BU
```

```
192.168.12.1 1/1/c2/1:1000
```

```
-----  
No. of IPv4 Prefix Active Bindings: 21  
=====
```

Finally, a synchronization timer is enabled between the IGP and LDP protocol when LDP FRR is enabled. From the moment that the interface for the previous primary next hop is restored, the IGP may reconverge back to that interface before LDP has completed the FEC exchange with its neighbor over that interface. This may cause LDP to de-program the LFA next hop from the FEC and blackhole the traffic. In this example, a synchronization timer of 10 seconds is configured, as follows:

```
# on all PEs:
configure {
  router "Base" {
    interface <itf-name> {
      ldp-sync-timer {
        seconds 10
      }
    }
  }
}
```

When this timer is set, on restoring a failed interface, the IGP advertises this link into the network with an infinite metric for the duration of this timer. When the failed link is restored, the LDP synchronization timer is started, and LDP adjacencies are brought up over the restored link and a label exchange is completed between the peers. After the LDP synchronization timer expires, the normal metric is advertised into the network again.

At this point, everything is in place to start creating LFA policies to influence the calculated LFA next hops.

2. Create a route next hop policy template.

This is a mandatory step in the context of LFA policies. The route next hop template name is 32 characters at maximum. Creating a route next hop policy is done in the following way:

```
configure {
  routing-options {
    route-next-hop-policy {
      template <template name>
    }
  }
}
```

After a **commit** of a route next hop policy template, the IGP re-evaluates the template and schedules a new LFA SPF to recompute the LFA next hop for the prefixes associated with this template.

3. Configure admin group constraints in route next hop policy.

Admin groups are optional in the context of LFA policies. First, configure a group name and a group value for each admin group locally on the router. Admin groups are configured as follows:

```
configure {
  routing-options {
    if-attribute {
      admin-group <group-name> {
        value <number>
      }
    }
  }
}
```

Second, configure the admin group membership of the IP interfaces (network, IES, or VPRN), as follows. Maximum 32 admin groups can be assigned to an IP interface in one command. The configured IP admin group membership applies to all levels or areas the interface is participating in.

```
configure {
```

```

router "Base" {
  interface <itf-name> {
    if-attribute {
      admin-group ["group-name-1" "group-name-2" ... (up to 32 max)]
    }
  }
  configure {
    service {
      vprn <svc-name> {
        interface <itf-name> {
          if-attribute {
            admin-group ["group-name-1" "group-name-2" ... (up to 32 max)]
          }
        }
      }
    }
  }
  configure {
    service {
      ies <svc-name> {
        interface <itf-name> {
          if-attribute {
            admin-group ["group-name-1" "group-name-2" ... (up to 32 max)]
          }
        }
      }
    }
  }
}

```

Third, add the IP admin group constraints to the route next hop policy template one by one. The **include-group** statement instructs the LFA SPF selection algorithm to select a subset of LFA next hops among the links which belong to one or more of the specified admin groups. A link which does not belong to any of the admin groups is excluded. The **preference** option is used to provide a relative preference for the admin group selection. A lower preference value means that LFA SPF will first attempt to select an LFA backup next hop which is a member of the corresponding admin group. If none is found, then the admin group with the next higher preference value is evaluated. If no preference value is configured, then it is the least preferred with a default preference value of 255.

When evaluating multiple **include-group** statements having the same preference, any link which belongs to one or more of the included admin groups can be selected as an LFA next hop. There is no relative preference based on how many of those included admin groups the link is a member of.

The **exclude-group** command simply prunes all links belonging to the specified admin group before making the LFA backup next hop selection for a prefix. If the same group name is part of both include and exclude statements, the exclude statement takes precedence. In other words, the exclude statement can be viewed as having an implicit preference value of 0.

Configure the admin group constraints in the route next hop policy template with the following command:

```

configure {
  routing-options {
    route-next-hop-policy {
      template <template-name> {
        exclude-group <ip-admin-group-name>
        include-group <ip-admin-group-name> {
          preference <preference>
        }
      }
    }
  }
}

```

4. Configure SRLG constraints in route next hop policy.

SRLG constraints are optional in the context of LFA policies. First, configure a group name and group value of each SRLG group locally on the router. The penalty weight controls the likelihood of paths with links sharing SRLG values with a primary path being used by a bypass or detour LSP. The higher the penalty weight, the less desirable it is to use the link with an SRLG. SRLG constraints are configured as follows:

```

configure {
  routing-options {

```

```

if-attribute {
    srlg-group <group-name> {
        value <group-value>
        penalty-weight <penalty-weight>           # default: 0
    }
}

```

Second, configure the SRLG group membership of the IP interfaces (network, IES, or VPRN), as follows. One SRLG group can be applied to an IP interface in the **srlg-group** command but the command can be applied multiple times. The configured IP SRLG group membership is applied in all levels or areas the interface is participating in.

```

configure {
    router "Base" {
        interface <itf-name> {
            if-attribute {
                srlg-group <group-name>
            }
        }
    }
    service {
        vprn <svc-name> {
            interface <itf-name> {
                if-attribute {
                    srlg-group <group-name>
                }
            }
        }
        ies <svc-name> {
            interface <itf-name> {
                if-attribute {
                    srlg-group <group-name>
                }
            }
        }
    }
}

```

Third, add IP SRLG group constraints to the route next hop policy template, as follows. When this command is applied to a prefix, the LFA SPF attempts to select an LFA next hop which uses an outgoing interface that does not participate in any of the SRLGs of the outgoing interface used by the primary next hop.

```

configure {
    routing-options {
        route-next-hop-policy {
            template <template-name> {
                srlg true
            }
        }
    }
}

```

5. Configure the protection type in route next hop policy.

This is an optional step in the context of LFA policies. With the following command, the user can also select if link protection or node protection is preferred for IP prefixes and LDP FEC prefixes protected by a backup LFA next hop. By default, node protection is chosen. The implementation falls back to link protection if no LFA next hop is found for node protection.

```

configure {
    routing-options {
        route-next-hop-policy {
            template <template-name> {
                protection-type {link|node}
            }
        }
    }
}

```

6. Configure the next hop preference type in route next hop policy.

This is an optional step in the context of LFA policies. With the following command, the user can also select if tunnel backup next hop or IP backup next hop is preferred for IP prefixes and LDP FEC prefixes protected by a backup LFA next hop. By default, IP backup next hop is chosen. The implementation falls back to the other type (tunnel) if no LFA next hop of the preferred type is found.

```
configure {
  routing-options {
    route-next-hop-policy {
      template <template-name> {
        nh-type {ip|tunnel}
      }
    }
  }
}
```

7. Apply the route next hop policy template to an IP interface.

When the route next hop policy is applied to an IP interface with one of the following commands, all prefixes using this interface as primary next hop take the selection criteria specified in Step 3, Step 4, Step 5, and Step 6 into account.

```
configure {
  router "Base" {
    ospf <ospf-instance> {
      area <area-id> {
        interface <itf-name> {
          loopfree-alternate {
            policy-map {
              route-nh-template <template-name>
            }
          }
        }
      }
    }
  }

  configure {
    router "Base" {
      ospf3 <ospf-instance> {
        area <area-id> {
          interface <itf-name> {
            loopfree-alternate {
              policy-map {
                route-nh-template <template-name>
              }
            }
          }
        }
      }
    }
  }

  configure {
    service {
      vprn <svc-name> {
        ospf <ospf-instance> {
          area <area-id> {
            interface <itf-name> {
              loopfree-alternate {
                policy-map {
                  route-nh-template <template-name>
                }
              }
            }
          }
        }
      }
    }
  }

  configure {
    service {
      vprn <svc-name> {
        ospf3 <ospf-instance> {
          area <area-id> {
            interface <itf-name> {
              loopfree-alternate {
                policy-map {
                  route-nh-template <template-name>
                }
              }
            }
          }
        }
      }
    }
  }
}
```

LFA policy examples

All the following examples focus on providing another LFA next hop for LDP FEC prefix 192.0.2.1/32 and 192.0.2.6/32 (the system IP addresses of PE-1 and PE-6), with PE-2 being the PLR.

See [Figure 28: Example topology](#) for the example topology.

The default LFA next hop (without policy) for LDP FEC prefix 192.0.2.1/32 is 192.168.26.2 on PE-6, as follows:

```
[/]
A:admin@PE-2# show router ldp bindings active prefixes prefix 192.0.2.1/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
      (IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static          (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop    (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix          Op
IngLbl          EgrLbl
EgrNextHop      EgrIf/LspId
-----
192.0.2.1/32    Push
--             524287
192.168.12.1    1/1/c2/1:1000

192.0.2.1/32    Push
--             524285BU
192.168.26.2    1/1/c3/1:1000

192.0.2.1/32    Swap
524286          524287
192.168.12.1    1/1/c2/1:1000

192.0.2.1/32    Swap
524286          524285BU
192.168.26.2    1/1/c3/1:1000

-----
No. of IPv4 Prefix Active Bindings: 4
=====
```

The default LFA next hop for LDP FEC prefix 192.0.2.6/32 is 192.168.12.1 on PE-1, as follows:

```
[/]
A:admin@PE-2# show router ldp bindings active prefixes prefix 192.0.2.6/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
      (IPv6 LSR ID ::)
=====
```

```

=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.6/32                          Push
--                                     524287
192.168.26.2                          1/1/c3/1:1000

192.0.2.6/32                          Push
--                                     524282BU
192.168.12.1                          1/1/c2/1:1000

192.0.2.6/32                          Swap
524282                                524287
192.168.26.2                          1/1/c3/1:1000

192.0.2.6/32                          Swap
524282                                524282BU
192.168.12.1                          1/1/c2/1:1000
-----
No. of IPv4 Prefix Active Bindings: 4
=====

```

This default LFA next hop can be changed by adding specific selection criteria inside a route next hop policy template.

Example 1: LFA policy with admin group constraint

The objective is to force the LFA next hop for both LDP FEC prefixes to use the path between PE-2 and PE-5.

Define admin group "red" with value 1 and apply it to the IP interfaces "int-PE-2-PE-1" and "int-PE-2-PE-6":

```

# on PE-2:
configure {
  routing-options {
    if-attribute {
      admin-group "red" {
        value 1
      }
    }
  }
  router "Base" {
    interface "int-PE-2-PE-1" {
      if-attribute {

```

```

        admin-group ["red"]
    }
}
interface "int-PE-2-PE-6" {
    if-attribute {
        admin-group ["red"]
    }
}

```

Define a route next hop policy template "LFA_NH_exclRed", which excludes IP admin group "red".

```

# on PE-2:
configure {
    routing-options {
        route-next-hop-policy {
            template "LFA_NH_exclRed" {
                exclude-group "red" { }
            }
        }
    }
}

```

Apply the policy to the OSPF interfaces toward PE-1 and PE-6:

```

# on PE-2:
configure {
    router "Base" {
        ospf 0 {
            area 0.0.0.0 {
                interface "int-PE-2-PE-1" {
                    loopfree-alternate {
                        policy-map {
                            route-nh-template "LFA_NH_exclRed"
                        }
                    }
                }
            }
            interface "int-PE-2-PE-6" {
                loopfree-alternate {
                    policy-map {
                        route-nh-template "LFA_NH_exclRed"
                    }
                }
            }
        }
    }
}

```

From the moment that the route next hop policy template "LFA_NH_exclRed" is applied to the OSPF interfaces toward PE-1 and PE-6, the LFA next hops for both LDP FEC prefixes change. They now both point to the IP interface from PE-2 to PE-5 as LFA backup next hop:

```

[/]
A:admin@PE-2# show router ldp bindings active prefixes prefix 192.0.2.1/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static          (M) - Multi-homed Secondary Support

```



```

(B) - BGP Next Hop      (BU) - Alternate Next-hop for Fast Re-Route
(I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
(C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                Op
IngLbl                EgrLbl
EgrNextHop            EgrIf/LspId
-----
192.0.2.1/32          Push
--                    524287
192.168.12.1         1/1/c2/1:1000

192.0.2.1/32          Push
--                    524286BU
192.168.25.2        1/1/c4/1:1000

192.0.2.1/32          Swap
524286                524287
192.168.12.1         1/1/c2/1:1000

192.0.2.1/32          Swap
524286                524286BU
192.168.25.2        1/1/c4/1:1000
-----
No. of IPv4 Prefix Active Bindings: 4
=====

```

```

[/]
A:admin@PE-2# show router ldp bindings active prefixes prefix 192.0.2.6/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
              (IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static          (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop    (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                Op
IngLbl                EgrLbl
EgrNextHop            EgrIf/LspId
-----
192.0.2.6/32          Push
--                    524287
192.168.26.2         1/1/c3/1:1000

192.0.2.6/32          Push
--                    524282BU
192.168.25.2        1/1/c4/1:1000

```

```

192.0.2.6/32          Swap
524282              524287
192.168.26.2        1/1/c3/1:1000

192.0.2.6/32          Swap
524282              524282BU
192.168.25.2        1/1/c4/1:1000

-----
No. of IPv4 Prefix Active Bindings: 4
=====

```

Example 2: LFA policy with SRLG constraint

The objective is to force the LFA next hop for both LDP FEC prefixes to use the path from PE-2 to PE-5. Define SRLG group "blue" with value 2 and apply it to the IP interfaces "int-PE-2-PE-1" and "int-PE-2-PE-6".

```

# on PE-2:
configure {
  routing-options {
    if-attribute {
      srlg-group "blue" {
        value 2
      }
    }
  }
  router "Base" {
    interface "int-PE-2-PE-1" {
      if-attribute {
        srlg-group "blue" { }
      }
    }
    interface "int-PE-2-PE-6" {
      if-attribute {
        srlg-group "blue" { }
      }
    }
  }
}

```

Define a route next hop policy template "LFA_NH_SRLG", where SRLG is enabled, as follows:

```

# on PE-2:
configure {
  routing-options {
    route-next-hop-policy {
      template "LFA_NH_SRLG" {
        srlg true
      }
    }
  }
}

```

Apply the policy to the OSPF interface toward PE-1 and PE-6:

```

# on PE-2:
configure {
  router "Base" {
    ospf 0 {
      area 0.0.0.0 {
        interface "int-PE-2-PE-1" {
          loopfree-alternate {
            policy-map {

```

```

        }
    }
}
interface "int-PE-2-PE-6" {
    loopfree-alternate {
        policy-map {
            route-nh-template "LFA_NH_SRLG"
        }
    }
}
}
}

```

Only one LFA policy mapping is allowed on an OSPF interface at a time. The new LFA policy mapping replaces the previous one.

The LFA next hops for both LDP FEC prefixes will both point now to the interface from PE-2 to PE-5 as LFA backup next hop, as follows:

```

[/]
A:admin@PE-2# show router ldp bindings active prefixes prefix 192.0.2.1/32
=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static           (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop     (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.1/32                         Push
--                                  524287
192.168.12.1                         1/1/c2/1:1000

192.0.2.1/32                         Push
--                                  524286BU
192.168.25.2                       1/1/c4/1:1000

192.0.2.1/32                         Swap
524286                               524287
192.168.12.1                         1/1/c2/1:1000

192.0.2.1/32                         Swap
524286                               524286BU
192.168.25.2                       1/1/c4/1:1000
-----
No. of IPv4 Prefix Active Bindings: 4

```

```

=====
[/]
A:admin@PE-2# show router ldp bindings active prefixes prefix 192.0.2.6/32
=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.6/32                         Push
--                                  524287
192.168.26.2                         1/1/c3/1:1000

192.0.2.6/32                         Push
--                                  524282BU
192.168.25.2                       1/1/c4/1:1000

192.0.2.6/32                         Swap
524282                               524287
192.168.26.2                         1/1/c3/1:1000

192.0.2.6/32                         Swap
524282                               524282BU
192.168.25.2                       1/1/c4/1:1000
-----
No. of IPv4 Prefix Active Bindings: 4
=====

```

The LFA policy mapping is removed from the OSPF interfaces as follows:

```

# on PE-2:
configure {
  router "Base" {
    ospf 0 {
      area 0.0.0.0 {
        interface "int-PE-2-PE-1" {
          delete loopfree-alternate
        }
        interface "int-PE-2-PE-6" {
          delete loopfree-alternate
        }
      }
    }
  }
}

```

Example 3: LFA policy with next hop type constraint

The objective is to force the LFA next hop for IP prefix 192.0.2.6/32 to use an RSVP tunnel.

Enable IP FRR as follows:

```
# on PE-2:
configure {
  routing-options {
    ip-fast-reroute true
  }
}
```

Set up an RSVP LSP tunnel toward 192.0.2.6 with a strict MPLS path going over PE-2 to PE-4 to PE-5 to PE-6.



Note:

Because an RSVP LSP is set up between PE-2 and PE-6, MPLS and RSVP protocols need to be enabled on all the corresponding IP interfaces along the MPLS path.

```
# on PE-2:
configure {
  router "Base" {
    mpls {
      interface "int-PE-2-PE-4" {
      }
      path "path-PE-2-PE-4-PE-5-PE-6" {
        admin-state enable
        hop 10 {
          ip-address 192.168.24.2
          type strict
        }
        hop 20 {
          ip-address 192.168.45.2
          type strict
        }
        hop 30 {
          ip-address 192.168.56.2
          type strict
        }
      }
      lsp "LSP-PE-2-PE-6-strict" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.6
        primary "path-PE-2-PE-4-PE-5-PE-6" {
        }
      }
    }
  }
}
```

Enable IGP shortcut with resolution filter RSVP within the IGP on PE-2 and indicate that the newly created RSVP LSP is a possible shortcut candidate for LFA backup next hop only.

```
# on PE-2:
configure {
  router "Base" {
    ospf 0 {
      igp-shortcut {
        admin-state enable
        tunnel-next-hop {
          family ipv4 {
            resolution filter
          }
        }
      }
    }
  }
}
```

```

        resolution-filter {
            rsvp true
        }
    }
}
mpls {
    lsp "LSP-PE-2-PE-6-strict" {
        igp-shortcut {
            lfa-type lfa-only
        }
    }
}

```

The following tunnel table on PE-2 for prefix 192.0.2.6 shows that an LDP LSP and an RSVP LSP are available toward PE-6:

```

[/]
A:admin@PE-2# show router tunnel-table 192.0.2.6

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.6/32         rsvp       MPLS 1           7     192.168.24.2 16777215
192.0.2.6/32 [L]    ldp        MPLS 65541          9     192.168.26.2 1
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====

```

The RSVP tunnel with tunnel ID 1 corresponds to the RSVP LSP "LSP-PE-2-PE-6-strict", as follows:

```

[/]
A:admin@PE-2# show router mpls lsp

=====
MPLS LSPs (Originating)
=====
LSP Name              Tun   Fastfail  Adm  Opr
  To                  Id     Config
-----
LSP-PE-2-PE-6-strict  1     No        Up   Up
  192.0.2.6
-----
LSPs : 1
=====

```

By default, the preferred next hop type is IP, not tunnel. Therefore, the RSVP tunnel will not be used for the LFA backup, as follows:

```

[/]
A:admin@PE-2# show router route-table alternative 192.0.2.6/32

=====
Route Table (Router: Base)
=====

```

```

=====
Dest Prefix[Flags]                                Type  Proto  Age      Pref
  Next Hop[Interface Name]                        Metric
  Alt-NextHop                                     Alt-
                                                Metric
-----
192.0.2.6/32                                       Remote OSPF   00h00m22s  10
  192.168.26.2                                       1
  192.168.12.1 (LFA)                                   2
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      Backup = BGP backup route
      LFA = Loop-Free Alternate nexthop
      S = Sticky ECMP requested
=====

```

Define a route next hop policy template "LFA_NH_Tunnel", where the next hop type is set to tunnel.

```

# on PE-2:
configure {
  routing-options {
    route-next-hop-policy {
      template "LFA_NH_Tunnel" {
        nh-type tunnel
      }
    }
  }
}

```

Apply the route next hop policy template to the OSPF interface toward PE-6, as follows:

```

# on PE-2:
configure {
  router "Base" {
    ospf 0 {
      area 0.0.0.0 {
        interface "int-PE-2-PE-6" {
          loopfree-alternate {
            policy-map {
              route-nh-template "LFA_NH_Tunnel"
            }
          }
        }
      }
    }
  }
}

```

The LFA next hop uses the RSVP tunnel. The reference to the RSVP tunnel ID 1 in the following show output corresponds with the tunnel ID shown in the preceding **show router tunnel-table 192.0.2.6** output:

```

[/]
A:admin@PE-2# show router route-table alternative 192.0.2.6/32

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                                Type  Proto  Age      Pref
  Next Hop[Interface Name]                        Metric
  Alt-NextHop                                     Alt-
                                                Metric
-----
192.0.2.6/32                                       Remote OSPF   00h00m38s  10
  192.168.26.2                                       1
  192.0.2.6 (LFA) (tunneled:RSVP:1)             65535
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      Backup = BGP backup route

```

```
LFA = Loop-Free Alternate nexthop
S = Sticky ECMP requested
=====
```

The following command shows the FIB next hop summary:

```
[/]
A:admin@PE-2# show router fib 1 nh-table-usage

=====
FIB Next-Hop Summary
=====
IPv4/IPv6                Active                Available
-----
IP Next-Hop              9                    65535
Tunnel Next-Hop         1                    993279
ECMP Next-Hop           0                    512000
ECMP Tunnel Next-Hop    0                    261120
=====
```

Example 4: Exclude prefix from LFA computation

The objective is to force no LFA next hop for LDP FEC prefix 192.0.2.1/32 where PE-2 is the PLR.

The IP FRR and LDP FRR implementation in SR OS allows to exclude an IGP interface, IGP area (OSPF), or IGP level (IS-IS) from the LFA SPF computation. The user can also exclude specific prefixes from the LFA SPF by using prefix lists and policy statements, which is configured as follows:

```
# on PE-2:
configure {
  policy-options {
    prefix-list "lo0-PE-1" {
      prefix 192.0.2.1/32 type exact {
      }
    }
  }
  policy-statement "LFA_Exclude_PE-1" {
    entry 10 {
      from {
        prefix-list ["lo0-PE-1"]
      }
      action {
        action-type accept
      }
    }
  }
}
}
```

The configured policy statement is applied to the IGP protocol, as follows:

```
# on PE-2:
configure {
  router "Base" {
    ospf 0 {
      loopfree-alternate {
        exclude {
          prefix-policy ["LFA_Exclude_PE-1"]
        }
      }
    }
  }
}
```


From the moment that it is applied, the existing LFA next hop entries for LDP FEC prefix 192.0.2.5/32 disappear instantly (compare with the preceding [example 1](#)):

```
[/]
A:admin@PE-2# show router ldp bindings active prefixes prefix 192.0.2.1/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
      (IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static          (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop    (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix          Op
IngLbl          EgrLbl
EgrNextHop      EgrIf/LspId
-----
192.0.2.1/32    Push
--             524287
192.168.12.1    1/1/c2/1:1000

192.0.2.1/32    Swap
524286          524287
192.168.12.1    1/1/c2/1:1000

-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

Conclusion

In production MPLS networks where IP FRR and/or LDP FRR are deployed, it is possible that the existing calculated LFA next hops are not always taking the most optimal or desirable paths.

With LFA policies, operators have better control on the way in which LFA backup next hops are computed.

Different selection criteria can be part of the route next hop policy: IP admin groups, IP SRLG groups, protection type preference, and next hop type preference.

PBR/PBF Redundancy

This chapter provides information about policy-based routing and policy-based forwarding redundancy.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written based on SR OS Release 14.0.R7, but the MD-CLI in the current edition corresponds to SR OS Release 23.7.R1. Secondary actions in IPv4, IPv6, and MAC access control list (ACL) filter policies are supported in SR OS Release 14.0.R1, and later.

Overview

PBR and PBF

Policy-based routing (PBR) and policy-based forwarding (PBF) are used to make forwarding decisions based on filter policies defined by the network administrator. PBR is L3 traffic steering, whereas PBF is L2 traffic steering. For ordinary routing, the destination IP address is looked up in the routing table; for ordinary forwarding in a VPLS, the destination MAC address is looked up in the forwarding database (FDB). However, with PBR, routing decisions are based on IP filters that use more criteria, such as source and destination IP address, port number, DSCP value, and so on. Packets can take paths that differ from the next hop path specified by the routing table. PBF forwarding decisions can be made based on IP filters, but also on MAC filters that use criteria such as source and destination MAC address, inner and outer VLAN tag, dot1p priority, and so on.

The benefits of PBR/PBF are the following:

- The forwarding decision can be based on multiple attributes of a packet, not only its destination address
- Different QoS treatment can be provided, based on additional criteria
- Cost saving: time-sensitive traffic can be sent over higher-speed links at a higher cost, while bulk file transfers are sent over lower-speed links at a lower cost
- Load sharing: traffic can be load balanced across multiple and unequal paths

In most situations, PBR/PBF works on inbound unicast packets; therefore, a filter is applied at the ingress of access or network interfaces. In this chapter, examples will be shown for IPv4 filters and MAC filters applied on SAP ingress. IPv6 filters are also supported, but the examples in this chapter are based on IPv4. Filters are also supported on the egress, but that is beyond the scope of this chapter.

An IPv4 filter contains one or more entries, which can be configured with the following command:

```
[ex:/configure filter ip-filter "IP-1"]  
A:admin@PE-1# entry 10 ?
```

```

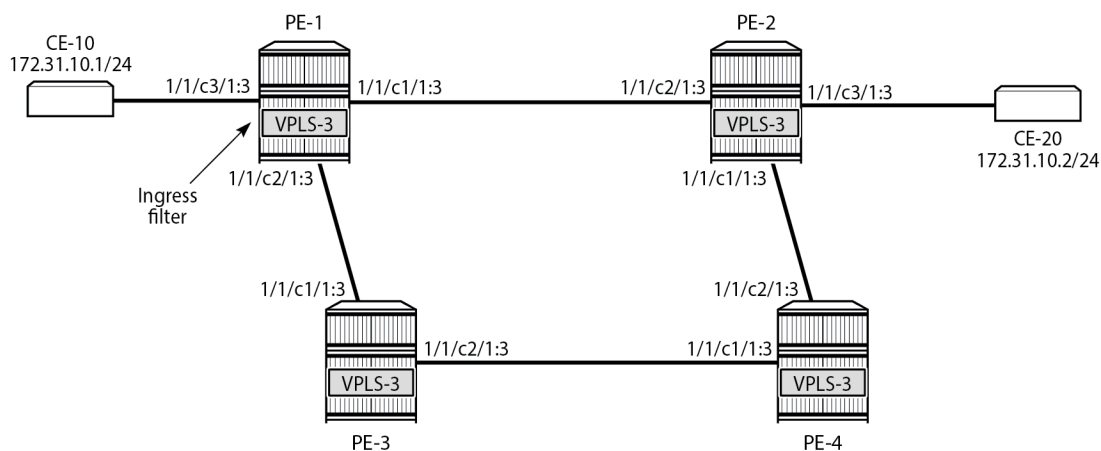
entry

Immutable fields      - egress-pbr

action                + Enable the action context
apply-groups          - Apply a configuration group at this level
apply-groups-exclude - Exclude a configuration group at this level
description           - Text description
egress-pbr            - PBR that has an effect when this filter is applied on egress
filter-sample         - Sample matching traffic if IP interface is set to cflowd ACL mode
interface-sample      - Sample matching traffic if IP interface is set to cflowd interface
mode
log                   - Log that is used for packets matching this entry
match                 + Enter the match context
pbr-down-action-override - Action when PBR or PBF target for this entry is not available
sample-profile        - Cflowd sample profile ID to match packets
sticky-dest           - Time before action with available PBR or PBF destination and highest
                        priority
    
```

Figure 29: PBF in the "VPLS-3" service on PE-1 shows the example topology with the "VPLS-3" service configured on the PEs. PBF is applied in the "VPLS-3" service on PE-1.

Figure 29: PBF in the "VPLS-3" service on PE-1



26309

The following configuration creates an IPv4 filter that forwards all packets matching the source and destination IPv4 addresses, 172.31.10.1/24 and 172.31.10.2/24 respectively, to SAP 1/1/c1/1:3. When SAP 1/1/c1/1:3 is operationally down, the default behavior is to drop the packet. Not every IPv4/v6 filter needs to have match criteria defined, but in this case, only packets with the configured IPv4 SA and IPv4 DA are affected, whereas the other packets are forwarded per the FDB in the "VPLS-3" service on PE-1.

```

configure {
  filter {
    ip-filter "IP-1" {
      filter-id 1
      entry 10 {
        match {
          src-ip {
            address 172.31.10.1
            mask 255.255.255.0
          }
        }
      }
    }
  }
}
    
```

```

    }
    dst-ip {
        address 172.31.10.2
        mask 255.255.255.0
    }
}
action {
    forward {
        sap {
            vpls "VPLS-3"
            sap-id 1/1/c1/1:3
        }
    }
}
}

```

In a similar way, an entry in a MAC filter can be configured with the following command:

```

[ex:/configure filter mac-filter "MAC-2" entry 10]
A:admin@PE-1# ?

action                + Enable the action context
apply-groups          - Apply a configuration group at this level
apply-groups-exclude - Exclude a configuration group at this level
description           - Text description
log                   - Log that is used for packets matching this entry
match                 + Enter the match context
pbr-down-action-override - Action when PBR or PBF target for this entry is not available
sticky-dest           - Time before action with available PBR or PBF destination and highest priority

```

The following MAC filter forwards all frames with source MAC SA 00:00:5e:00:53:01 to SAP 1/1/c1/1:3:

```

configure {
    filter {
        mac-filter "MAC-2" {
            filter-id 2
            entry 10 {
                match {
                    src-mac {
                        address 00:00:5e:00:53:01
                    }
                }
                action {
                    forward {
                        sap {
                            vpls "VPLS-3"
                            sap-id 1/1/c1/1:3
                        }
                    }
                }
            }
        }
    }
}

```

Instead of defining a specific MAC address, a range of MAC addresses can be defined using a mask. The default mask is all 1s, ff:ff:ff:ff:ff:ff (not shown), which corresponds to an exact match of the configured MAC address.

When the primary SAP 1/1/c1/1:3 is down, the default action is drop. However, PBR/PBF redundancy can be configured, as described in the following section.

PBR/PBF redundancy

PBR/PBF redundancy is supported for MAC filters, IPv4 filters, and IPv6 filters. Within each entry in the IP/MAC filter, a secondary action can be configured; for example, for entry 10 in IPv4 filter "IP-1", as follows:

```
configure {
  filter {
    ip-filter "IP-1" {
      filter-id 1
      entry 10 {
        match {
          src-ip {
            address 172.31.10.1
            mask 255.255.255.0
          }
          dst-ip {
            address 172.31.10.2
            mask 255.255.255.0
          }
        }
        action {
          forward {
            sap {
              vpls "VPLS-3"
              sap-id 1/1/c1/1:3
            }
          }
          secondary {
            forward {
              sap {
                vpls "VPLS-3"
                sap-id 1/1/c2/1:3
              }
            }
          }
        }
      }
    }
  }
}
```

The IPv4 filter is applied on the ingress of SAP 1/1/c3/1:3 in the "VPLS-3" service on PE-1. This IPv4 filter only affects packets with IPv4 SA 172.31.10.1/24 and IPv4 DA 172.31.10.2/24. When the primary action SAP 1/1/c1/1:3 is operationally up, the primary action is executed; when SAP 1/1/c1/1:3 is operationally down, the secondary action is executed, until SAP 1/1/c1/1:3 is operationally up again. When both SAPs are down, the default behavior is to drop the packet.

When the primary action SAP 1/1/c1/1:3 is operationally up (PBR Target Status: Up), the primary action is executed (Downloaded Action: Primary), as follows:

```
[/]
A:admin@PE-1# show filter ip "IP-1"

=====
IP Filter
=====
Filter Id       : 1                               Applied       : Yes
Scope          : Template                       Def. Action   : Drop
Type           : Normal
Shared Policer  : Off
System filter   : Unchained
Radius Ins Pt  : n/a
CrCtl. Ins Pt  : n/a
```

```

RadSh. Ins Pt      : n/a
PccRl. Ins Pt     : n/a
Entries           : 1
Description       : (Not Specified)
Filter Name       : IP-1
-----
Filter Match Criteria : IP
-----
Entry             : 10
Description       : (Not Specified)
Log Id            : n/a
Src. IP           : 172.31.10.1/24
Src. Port         : n/a
Dest. IP          : 172.31.10.2/24
Dest. Port        : n/a
Protocol          : Undefined
Dscp              : Undefined
ICMP Type         : Undefined          ICMP Code      : Undefined
Fragment          : Off                Src Route Opt  : Off
Sampling          : Off                Int. Sampling  : On
IP-Option         : 0/0                Multiple Option: Off
Tcp-flag          : (Not Specified)
Option-pres       : Off
Egress PBR        : Disabled
Primary Action   : Forward (SAP)
  Next Hop        : 1/1/c1/1:3
  Service Id      : 3
  PBR Target Status : Up
Secondary Action  : Forward (SAP)
  Next Hop        : 1/1/c2/1:3
  Service Id      : 3
  PBR Target Status : Up
PBR Down Action   : Drop (entry-default)
Downloaded Action : Primary
Dest. Stickiness  : None                Hold Remain    : 0
Ing. Matches      : 205 pkts (21730 bytes)
Egr. Matches      : 0 pkts
=====

```

When the primary action SAP 1/1/c1/1:3 is operationally down, the secondary action is executed. When SAP 1/1/c1/1:3 is down, packets are forwarded to secondary action SAP 1/1/c2/1:3 instead. However, when the primary action SAP 1/1/c1/1:3 is operationally up again, the primary action is executed. This revertive behavior can be disabled by configuring stickiness in the filter entry, as follows:

```

[ex:/configure filter ip-filter "IP-1" entry 10]
A:admin@PE-1# sticky-dest ?

```

```

sticky-dest (<number> | <keyword>)
<number> - <0..65535> - seconds
<keyword> - no-hold-time-up - seconds

```

Time before action with available PBR or PBF destination and highest priority

When both the primary action SAP 1/1/c1/1:3 and the secondary action SAP 1/1/c2/1:3 are down, the default action is drop, unless the **pbr-down-action-override <filter-action>** parameter is configured. When the configured filter action is **forward**, the packets can be forwarded to another object in the service that is up, for example, to another SAP or to an SDP binding, per the packet's destination address. This means that in a VPLS (PBF), the MAC DA is looked up in the FDB; in a VPRN (PBR), the IP DA is looked

up in the routing table. The configuration of the **pbr-down-action-override** parameter is as follows. No specific SAPs or SDP bindings need to be defined.

```
[ex:/configure filter ip-filter "IP-1" entry 10]
A:admin@PE-1# pbr-down-action-override ?

pbr-down-action-override <keyword>
<keyword> - (drop|forward|filter-default-action)

Action when PBR or PBF target for this entry is not available
```

In the example, the filter "IP-1" contains two actions that both forward packets to a SAP, but the PBR/PBF target can also be an SDP binding or—for PBR—a next-hop IP address in a VPRN. [Table 1: Primary and secondary forwarding actions](#) shows the allowed primary and secondary forwarding action combinations within a filter entry.

Table 1: Primary and secondary forwarding actions

primary forwarding action	secondary forwarding action
sap <sap-id>	sap <sap-id>
sap <sap-id>	sdp <sdp-id:vc-id>
sdp <sdp-id:vc-id>	sdp <sdp-id:vc-id>
sdp <sdp-id:vc-id>	sap <sap-id>
next-hop <ipv4/ipv6-address> router <router-instance>	next-hop <ipv4-ipv6-address> router <router-instance>
next-hop indirect <ipv4/ipv6-address> router <router-instance>	next-hop indirect <ipv4/ipv6-address> router <router-instance>

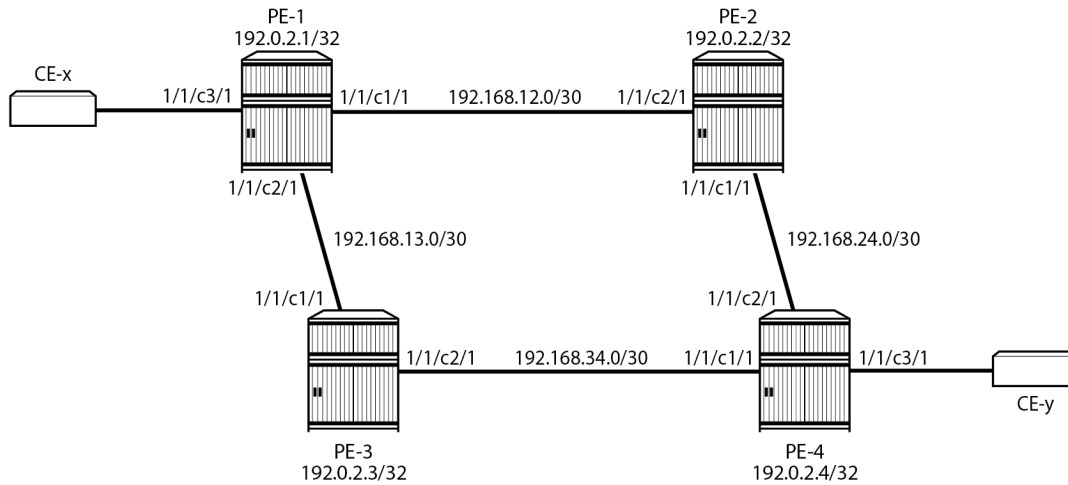
Configuration

In this section, the following examples are described:

- [PBF in a VPLS using an IPv4 filter](#)
- [PBF in a VPLS using a MAC filter](#)
- [PBR in a VPRN using an IPv4 filter](#)

[Figure 30: Example topology](#) shows the example topology with four PEs and two CEs.

Figure 30: Example topology



26308

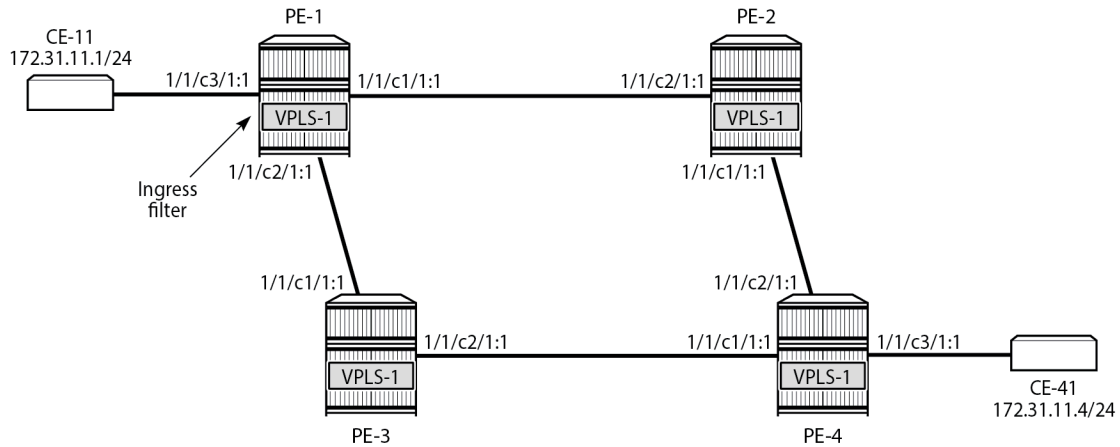
The initial configuration is as follows:

- Cards, MDAs, ports (all ports are in hybrid mode with dot1q encapsulation)
- Router interfaces
- IS-IS as IGP between the PEs (alternatively, OSPF could be configured as IGP)
- LDP between the PEs
- The CEs are emulated using a VPRN on PE-1 or PE-4 with a hairpin to loop the traffic back to the PE.

PBF in a VPLS using an IP filter

Figure 31: PBF in the "VPLS-1" service on PE-1 shows the example topology with the "VPLS-1" service configured on the four PEs. CE-11 is connected with the "VPLS-1" service on PE-1 and CE-14 with the "VPLS-1" service on PE-4. PBF is applied in the "VPLS-1" service on PE-1.

Figure 31: PBF in the "VPLS-1" service on PE-1



26309

The configuration is shown for PE-1. The following cases are described in this section:

1. Initial situation: primary action is executed.
2. Primary action SAP 1/1/c1/1:1 is disabled. The secondary action in the entry in the IPv4 filter is executed.
3. Both primary and secondary action SAPs 1/1/c1/1:1 and 1/1/c2/1:1 are disabled. The default action is drop.
4. Both primary and secondary action SAPs 1/1/c1/1:1 and 1/1/c2/1:1 are disabled. The **pbr-down-action-override** parameter is configured with action *forward*.
5. The secondary action SAP 1/1/c2/1:1 is re-enabled. The secondary action is executed.
6. The primary action SAP 1/1/c1/1:1 is re-enabled. The primary action is executed.
7. Stickiness is configured with a hold timer of, for example, 120 seconds. At timer expiry, stickiness takes effect. If SAP 1/1/c1/1:1 is up at timer expiry, the primary action is programmed; otherwise, if SAP 1/1/c2/1:1 is up, the secondary action is programmed.
8. Stickiness is configured without a hold timer and takes effect immediately.

Configure the "VPLS-1" service with IPv4 filter on SAP ingress

IPv4 filter 10 has one entry with primary action to forward to SAP 1/1/c1/1:1 and secondary action to forward to SAP 1/1/c2/1:1. No match criteria are defined. When all action forward SAPs are operationally down, the default action is drop. No stickiness is configured.

```
# on PE-1:
configure {
  filter {
    ip-filter "IP-10" {
      filter-id 10
      entry 10 {
        action {
          forward {
            sap {
              vpls "VPLS-1"
            }
          }
        }
      }
    }
  }
}
```


IS-IS and LDP. The port statistics are cleared for ports 1/1/c1/1 through 1/1/c3/1 on PE-1. CE-11 sends a series of ICMP echo requests and, afterward, the port statistics on PE-1 are verified.

```
[/]
A:admin@PE-1# ping 172.31.11.4 router-instance "CE-11" source-address 172.31.11.1
                                     interval 0.01 count 200 output-format summary
PING 172.31.11.4 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
----- 172.31.11.4 PING Statistics -----
200 packets transmitted, 200 packets received, 0.00% packet loss
round-trip min = 2.20ms, avg = 2.80ms, max = 11.1ms, stddev = 0.966ms
```

```
[/]
A:admin@PE-1# show port 1/1/c1/1 statistics

=====
Port Statistics on Slot 1
=====
Port                               Ingress Packets      Ingress Octets
Id                                 Egress Packets      Egress Octets
-----
1/1/c1/1                            203                  21545
                                      205                  21743
=====
```

```
[/]
A:admin@PE-1# show port 1/1/c2/1 statistics

=====
Port Statistics on Slot 1
=====
Port                               Ingress Packets      Ingress Octets
Id                                 Egress Packets      Egress Octets
-----
1/1/c2/1                             3                   315
                                      1                   129
=====
```

```
[/]
A:admin@PE-1# show port 1/1/c3/1 statistics

=====
Port Statistics on Slot 1
=====
Port                               Ingress Packets      Ingress Octets
Id                                 Egress Packets      Egress Octets
-----
1/1/c3/1                             200                  21200
                                      200                  21200
=====
```

All traffic is forwarded from ingress SAP 1/1/c3/1:1 to SAP 1/1/c1/1:1 and the reply messages from SAP 1/1/c1/1:1 to SAP 1/1/c3/1:1. No packets are forwarded via SAP 1/1/c2/1:1.

When the primary action SAP 1/1/c1/1:1 is operationally up, the primary action is executed, as follows:

```
[/]
A:admin@PE-1# show filter ip "IP-10"

=====
```

```

IP Filter
=====
Filter Id      : 10                      Applied      : Yes
Scope         : Template                 Def. Action  : Drop
---snip---

-----
Filter Match Criteria : IP
-----
Entry         : 10
---snip---

Primary Action      : Forward (SAP)
  Next Hop         : 1/1/c1/1:1
  Service Id       : 1
  PBR Target Status : Up
Secondary Action   : Forward (SAP)
  Next Hop         : 1/1/c2/1:1
  Service Id       : 1
  PBR Target Status : Up
PBR Down Action    : Drop (entry-default)
Downloaded Action  : Primary
Dest. Stickiness  : None                  Hold Remain  : 0
Ing. Matches      : 203 pkts (21518 bytes)
Egr. Matches      : 0 pkts
=====

```

Primary action PBR target down

The primary action SAP 1/1/c1/1:1 is disabled. Therefore, the primary action cannot be executed, and the secondary action is executed instead. When CE-11 sends ICMP echo requests, all packets are forwarded to SAP 1/1/c2/1:1.

```

# Disable SAP 1/1/c1/1:1 in the "VPLS-1" service on PE-1:
configure {
  service {
    vpls "VPLS-1" {
      sap 1/1/c1/1:1
      admin-state disable
    }
  }
}
[/]
A:admin@PE-1# show filter ip "IP-10"

=====
IP Filter
=====
Filter Id      : 10                      Applied      : Yes
Scope         : Template                 Def. Action  : Drop
---snip---

Entry         : 10
---snip---

Primary Action      : Forward (SAP)
  Next Hop         : 1/1/c1/1:1
  Service Id       : 1
  PBR Target Status : Down
Secondary Action   : Forward (SAP)
  Next Hop         : 1/1/c2/1:1
  Service Id       : 1

```

```

PBR Target Status : Up
PBR Down Action    : Drop (entry-default)
Downloaded Action : Secondary
Dest. Stickiness   : None                      Hold Remain   : 0
Ing. Matches       : 403 pkts (42718 bytes)
Egr. Matches       : 0 pkts
=====

```

Secondary action PBR target down

The secondary action SAP 1/1/c2/1:1 is disabled, as follows:

```

# Disable SAP 1/1/c2/1:1 in the "VPLS-1" service on PE-1:
configure {
  service {
    vpls "VPLS-1" {
      sap 1/1/c2/1:1
        admin-state disable
    }
  }
}

```

Both SAP 1/1/c1/1:1 and SAP 1/1/c2/1:1 are disabled. Neither the primary nor the secondary action in entry 10 of IPv4 filter 10 can be executed. Therefore, the default action is executed, which is drop; see the following output (PBR Down Action: Drop (entry-default)).

```

[/]
A:admin@PE-1# show filter ip "IP-10"

=====
IP Filter
=====
Filter Id       : 10                      Applied       : Yes
Scope          : Template                 Def. Action   : Drop
---snip---

Entry          : 10
---snip---

Primary Action  : Forward (SAP)
  Next Hop     : 1/1/c1/1:1
  Service Id   : 1
  PBR Target Status : Down
Secondary Action : Forward (SAP)
  Next Hop     : 1/1/c2/1:1
  Service Id   : 1
  PBR Target Status : Down
PBR Down Action : Drop (entry-default)
Downloaded Action : Primary
Dest. Stickiness : None                      Hold Remain   : 0
Ing. Matches     : 403 pkts (42718 bytes)
Egr. Matches     : 0 pkts
=====

```

When CE-11 sends ICMP echo requests, they are all dropped.

```

[/]
A:admin@PE-1# ping 172.31.11.4 router-instance "CE-11" source-address 172.31.11.1
                                     interval 0.01 count 50 output-format summary
PING 172.31.11.4 56 data bytes

```

```
.....
---- 172.31.11.4 PING Statistics ----
50 packets transmitted, 0 packets received, 100% packet loss
```

PBR down action override

Both SAPs remain disabled. The default PBR down action is drop, but that can be overruled by configuring the **pbr-down-action-override** parameter, as follows:

```
# on PE-1:
configure {
  filter {
    ip-filter "IP-10" {
      entry 10 {
        pbr-down-action-override forward
      }
    }
  }
}
```

With this configuration added in entry 10 of the "IP-10" filter, the PBR down action will be forward. No specific next hop needs to be defined. The forwarding is based on the destination address. When CE-11 sends ICMP echo requests to CE-41, the traffic is forwarded, as follows:

```
[/]
A:admin@PE-1# ping 172.31.11.4 router-instance "CE-11" source-address 172.31.11.1
                                     interval 0.01 count 200 output-format summary
PING 172.31.11.4 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
---snip---
---- 172.31.11.4 PING Statistics ----
200 packets transmitted, 200 packets received, 1 duplicate
round-trip min = 2.29ms, avg = 2.94ms, max = 12.0ms, stddev = 0.752ms
```

The statistics in the detailed output for spoke-SDP 12:1 in the "VPLS-1" service shows that these packets have been sent over this spoke-SDP. It is possible that spoke-SDP 13:1 in the "VPLS-1" service is used instead.

```
[/]
A:admin@PE-1# show service id 1 sdp 12:1 detail | match Statistics post-lines 5
Statistics
:
I. Fwd. Pkts.      : 203                I. Dro. Pkts.      : 0
I. Fwd. Octs.     : 19818             I. Dro. Octs.     : 0
E. Fwd. Pkts.     : 207                E. Fwd. Octets    : 20020
-----
```

The PBR down action for entry 10 in IPv4 filter 10 is forward, as defined by the **pbr-down-action-override** parameter, and the PBR downloaded action is forward, as follows:

```
[/]
A:admin@PE-1# show filter ip "IP-10"

=====
IP Filter
=====
Filter Id          : 10                Applied           : Yes
Scope              : Template          Def. Action       : Drop
---snip---

Entry              : 10
---snip---
```

```

Primary Action      : Forward (SAP)
Next Hop           : 1/1/c1/1:1
Service Id         : 1
PBR Target Status : Down
Secondary Action   : Forward (SAP)
Next Hop           : 1/1/c2/1:1
Service Id         : 1
PBR Target Status : Down
PBR Down Action   : Forward (pbr-down-action-override)
Downloaded Action : Forward
Dest. Stickiness   : None                      Hold Remain    : 0
Ing. Matches       : 653 pkts (69218 bytes)
Egr. Matches       : 0 pkts
=====

```

Secondary action up - revertive behavior

The primary action SAP 1/1/c1/1:1 remains disabled, whereas secondary action SAP 1/1/c2/1:1 is re-enabled, as follows:

```

# on PE-1:
configure {
  service {
    vpls "VPLS-1" {
      sap 1/1/c2/1:1 {
        admin-state enable
      }
    }
  }
}

```

The secondary action in entry 10 of IPv4 filter 10 is executed (Downloaded Action: Secondary), as follows:

```

[/]
A:admin@PE-1# show filter ip "IP-10"

=====
IP Filter
=====
Filter Id      : 10                      Applied       : Yes
Scope         : Template                 Def. Action   : Drop
---snip---

Entry         : 10
---snip---

Primary Action : Forward (SAP)
Next Hop       : 1/1/c1/1:1
Service Id     : 1
PBR Target Status : Down
Secondary Action : Forward (SAP)
Next Hop       : 1/1/c2/1:1
Service Id     : 1
PBR Target Status : Up
PBR Down Action   : Forward (pbr-down-action-override)
Downloaded Action : Secondary
Dest. Stickiness : None                      Hold Remain    : 0
Ing. Matches     : 853 pkts (90418 bytes)
Egr. Matches     : 0 pkts
=====

```

Primary action up - revertive behavior

As well as the secondary action SAP, also the primary action SAP 1/1/c1/1:1 is re-enabled, as follows:

```
# on PE-1:
configure {
  service {
    vpls "VPLS-1" {
      sap 1/1/c1/1:1 {
        admin-state enable
      }
    }
  }
}
```

The default PBR/PBF behavior is revertive; therefore, the primary action is executed: the packets are forwarded to SAP 1/1/c1/1:1, as follows:

```
[/]
A:admin@PE-1# show filter ip "IP-10"

=====
IP Filter
=====
Filter Id       : 10                               Applied       : Yes
Scope          : Template                         Def. Action   : Drop
---snip---

Entry          : 10
---snip---

Primary Action  : Forward (SAP)
Next Hop       : 1/1/c1/1:1
Service Id     : 1
PBR Target Status : Up
Secondary Action : Forward (SAP)
Next Hop       : 1/1/c2/1:1
Service Id     : 1
PBR Target Status : Up
PBR Down Action : Forward (pbr-down-action-override)
Downloaded Action : Primary
Dest. Stickiness : None                               Hold Remain   : 0
Ing. Matches    : 1053 pkts (111618 bytes)
Egr. Matches    : 0 pkts

=====
```

Stickiness in IP filter with hold timer

When the primary action SAP becomes up, traffic will be forwarded to this SAP instantaneously, unless stickiness applies. Stickiness can be defined on the IPv4/v6 filter entry level to override this revertive behavior. The following command enables stickiness at timer expiry with a hold remain timer of—in this case—120 seconds for entry 10 in IPv4 filter 10:

```
# on PE-1:
configure {
  filter {
    ip-filter "IP-10" {
      entry 10 {
        sticky-dest 120
      }
    }
  }
}
```


The hold remain timer starts counting down when stickiness is configured and at least one PBR target is up. If the primary action SAP 1/1/c1/1:1 remains operationally up for the configured 120 seconds, the primary action will be active, and at timer expiry, stickiness applies. However, if SAP 1/1/c1/1:1 goes down and then up again before timer expiry, the secondary action remains active until the hold remain timer expires, as shown in the following example.

The hold remain timer has not expired. The primary action SAP 1/1/c1/1:1 is disabled, so the secondary action is active, as follows. The hold remain timer keeps counting down.

```
# on PE-1:
configure exclusive
  service {
    vpls "VPLS-1" {
      sap 1/1/c1/1:1 {
        admin-state disable
      }
    }
  }

```

```
[/]
A:admin@PE-1# show filter ip "IP-10"

=====
IP Filter
=====
Filter Id           : 10                               Applied           : Yes
Scope              : Template                         Def. Action       : Drop
---snip---

Entry              : 10
---snip---

Primary Action      : Forward (SAP)
Next Hop           : 1/1/c1/1:1
Service Id         : 1
PBR Target Status : Down
Secondary Action    : Forward (SAP)
Next Hop           : 1/1/c2/1:1
Service Id         : 1
PBR Target Status : Up
PBR Down Action     : Forward (pbr-down-action-override)
Downloaded Action : Secondary
Dest. Stickiness  : 120                               Hold Remain     : 91
Ing. Matches        : 1253 pkts (132818 bytes)
Egr. Matches        : 0 pkts

=====

```

The primary action SAP 1/1/c1/1:1 is restored and the secondary action is active until the hold remain timer expires, as follows:

```
# on PE-1:
configure {
  service {
    vpls "VPLS-1" {
      sap 1/1/c1/1:1 {
        admin-state enable
      }
    }
  }
}

[/]
A:admin@PE-1# show filter ip "IP-10"

=====
IP Filter
=====

```

```

=====
Filter Id       : 10                               Applied      : Yes
Scope          : Template                         Def. Action   : Drop
---snip---

Entry          : 10
---snip---

Primary Action  : Forward (SAP)
  Next Hop     : 1/1/c1/1:1
  Service Id   : 1
  PBR Target Status : Up
Secondary Action : Forward (SAP)
  Next Hop     : 1/1/c2/1:1
  Service Id   : 1
  PBR Target Status : Up
PBR Down Action : Forward (pbr-down-action-override)
Downloaded Action : Secondary
Dest. Stickiness  : 120                               Hold Remain   : 55
Ing. Matches    : 1453 pkts (154018 bytes)
Egr. Matches    : 0 pkts

=====

```

In the preceding output, the secondary action is active and the hold remain time is 55 seconds. When the hold remain timer expires and the primary action SAP 1/1/c1/1:1 is up, the primary action is activated again and stickiness applies from then onward, as follows:

```

[/]
A:admin@PE-1# show filter ip "IP-10"

=====
IP Filter
=====
Filter Id       : 10                               Applied      : Yes
Scope          : Template                         Def. Action   : Drop
---snip---

Primary Action  : Forward (SAP)
  Next Hop     : 1/1/c1/1:1
  Service Id   : 1
  PBR Target Status : Up
Secondary Action : Forward (SAP)
  Next Hop     : 1/1/c2/1:1
  Service Id   : 1
  PBR Target Status : Up
PBR Down Action : Forward (pbr-down-action-override)
Downloaded Action : Primary
Dest. Stickiness  : 120                               Hold Remain   : 0
Ing. Matches    : 1453 pkts (154018 bytes)
Egr. Matches    : 0 pkts

=====

```

The hold remain timer stays at zero. When the primary action cannot be activated, the secondary action is activated and will remain activated even when the primary action SAP 1/1/c1/1:1 is up again. However, when the secondary action SAP 1/1/c2/1:1 is down, the primary action can be activated again.

The hold remain timer starts counting down when it is first configured, or reconfigured with a different value, and at least one of the PBR/PBF targets is up. The hold remain timer also starts counting down after both the primary and the secondary PBR/PBF targets have been down, for example, after a reboot, and at

least one of them transitions to the up status. The secondary action might be available first, even though the primary action is preferred. This situation is automatically resolved when the timer expires: the primary action will be activated if available when the hold remain timer expires.

Force primary action

Stickiness can be enabled without any delay, as follows:

```
# on PE-1:
configure exclusive
  filter {
    ip-filter "IP-10" {
      entry 10 {
        sticky-dest no-hold-time-up
      }
    }
  }

```

Initially, the primary action was executed, but when the primary action SAP 1/1/c1/1:1 is disabled, the secondary action is executed, as follows:

```
# on PE-1:
configure {
  service {
    vpls "VPLS-1" {
      sap 1/1/c1/1:1
      admin-state disable
    }
  }
}

```

```
[/]
A:admin@PE-1# show filter ip "IP-10"

=====
IP Filter
=====
Filter Id          : 10                      Applied           : Yes
Scope              : Template                Def. Action       : Drop
---snip---

Entry              : 10
---snip---

Primary Action     : Forward (SAP)
  Next Hop         : 1/1/c1/1:1
  Service Id       : 1
  PBR Target Status : Down
Secondary Action   : Forward (SAP)
  Next Hop         : 1/1/c2/1:1
  Service Id       : 1
  PBR Target Status : Up
PBR Down Action    : Forward (pbr-down-action-override)
Downloaded Action : Secondary
Dest. Stickiness  : 0                      Hold Remain     : 0
Ing. Matches       : 1653 pkts (175218 bytes)
Egr. Matches       : 0 pkts

=====

```

The secondary action is active and will remain active as long as the secondary action SAP 1/1/c2/1:1 is up. The hold remain timer is not enabled (== value 0). When the primary action SAP 1/1/c1/1:1 is operationally up again, the secondary action remains active, as follows:

```
# on PE-1:
configure {
  service {
    vpls "VPLS-1" {
      sap 1/1/c1/1:1
      admin-state enable
    }
  }
}

[/]
A:admin@PE-1# show filter ip "IP-10"

=====
IP Filter
=====
Filter Id          : 10                      Applied           : Yes
Scope              : Template                Def. Action       : Drop
---snip---

Entry              : 10
---snip---

Primary Action     : Forward (SAP)
Next Hop           : 1/1/c1/1:1
Service Id         : 1
PBR Target Status : Up
Secondary Action   : Forward (SAP)
Next Hop           : 1/1/c2/1:1
Service Id         : 1
PBR Target Status : Up
PBR Down Action    : Forward (pbr-down-action-override)
Downloaded Action : Secondary
Dest. Stickiness : 0                      Hold Remain      : 0
Ing. Matches       : 1853 pkts (196418 bytes)
Egr. Matches       : 0 pkts

=====
```

The following **tools** command forces activation of the primary action in entry 10 of the "IP-10" filter:

```
[/]
A:admin@PE-1# tools perform filter ip-filter 10 entry 10 activate-primary-action
```

The result is that the primary action is executed again, as shown in the following output:

```
[/]
A:admin@PE-1# show filter ip "IP-10"

=====
IP Filter
=====
Filter Id          : 10                      Applied           : Yes
Scope              : Template                Def. Action       : Drop
---snip---

Entry              : 10
---ping---

Primary Action     : Forward (SAP)
```

```

Next Hop      : 1/1/c1/1:1
Service Id    : 1
PBR Target Status : Up
Secondary Action : Forward (SAP)
Next Hop      : 1/1/c2/1:1
Service Id    : 1
PBR Target Status : Up
PBR Down Action : Forward (pbr-down-action-override)
Downloaded Action : Primary
Dest. Stickiness : 0                               Hold Remain   : 0
Ing. Matches    : 2053 pkts (217618 bytes)
Egr. Matches    : 0 pkts
=====

```

This **tools** command can also be used in combination with a running sticky-destination hold remain timer. In that case, the hold remain timer will stop counting down and the primary action immediately reverts.

PBF in a VPLS using a MAC filter

PBF in a VPLS can use a MAC filter instead of an IPv4 filter, but not both. The following MAC filter is defined on PE-1:

```

configure exclusive
  filter {
    mac-filter "MAC-20" {
      filter-id 20
      entry 10 {
        pbr-down-action-override forward
        sticky-dest no-hold-time-up
        match {
          src-mac {
            address 00:00:5e:00:53:11
          }
        }
        action {
          forward {
            sap {
              vpls "VPLS-1"
              sap-id 1/1/c1/1:1
            }
          }
          secondary {
            forward {
              sap {
                vpls "VPLS-1"
                sap-id 1/1/c2/1:1
              }
            }
          }
        }
      }
    }
  }
}

```

MAC filter "MAC-20" cannot be applied next to IPv4 filter "IP-10" on the ingress direction of SAP 1/1/c3/1:1 in the "VPLS-1" service; therefore, an error message is raised, as follows:

```

[ex:/configure service vpls "VPLS-1" sap 1/1/c3/1:1 ingress filter]
A:admin@PE-1# mac "MAC-20"

*[ex:/configure service vpls "VPLS-1" sap 1/1/c3/1:1 ingress filter]

```

```
A:admin@PE-1# info
  mac "MAC-20"
  ip "IP-10"

*[ex:/configure service vpls "VPLS-1" sap 1/1/c3/1:1 ingress filter]
A:admin@PE-1# commit
MINOR: SVCNMR #12: configure service vpls "VPLS-1" sap 1/1/c3/1:1 ingress filter mac
- Inconsistent Value error - another filter is already configured
```

The filter that was applied must be removed first, then the MAC filter can be applied, as follows:

```
# on PE-1:
configure {
  service {
    vpls "VPLS-1" {
      sap 1/1/c3/1:1 {
        ingress {
          delete filter          # remove filter
          filter {
            mac "MAC-20"
          }
        }
      }
    }
  }
}
```

When all SAPs in the "VPLS-1" service are up, the primary action is activated, as follows:

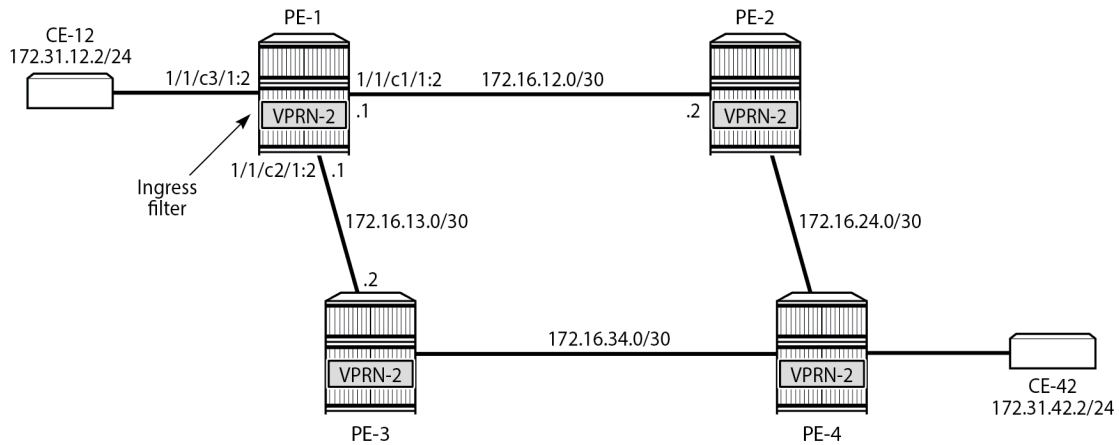
```
[/]
A:admin@PE-1# show filter mac "MAC-20"

=====
Mac Filter
=====
Filter Id       : 20                               Applied        : Yes
Scope          : Template                         Def. Action    : Drop
Entries        : 1                               Type           : normal
Description     : (Not Specified)
Filter Name     : MAC-20
-----
Filter Match Criteria : Mac
-----
Entry          : 10                               FrameType      : Ethernet
Description    : (Not Specified)
Log Id        : n/a
Src Mac       : 00:00:5e:00:53:11 ff:ff:ff:ff:ff:ff
Dest Mac      : Undefined
Dot1p         : Undefined                         Ethertype      : Undefined
DSAP          : Undefined                         SSAP           : Undefined
Snap-pid      : Undefined                         ESnap-oui-zero : Undefined
Primary Action : Forward (SAP)
  Next Hop    : 1/1/c1/1:1
  Service Id  : 1
PBR Target Status : Up
Secondary Action : Forward (SAP)
  Next Hop    : 1/1/c2/1:1
  Service Id  : 1
PBR Target Status : Up
PBR Down Action : Forward (pbr-down-action-override)
Downloaded Action : Primary
Dest. Stickiness : 0                               Hold Remain    : 0
Ing. Matches     : 200 pkts (21200 bytes)
Egr. Matches     : 0 pkts
=====
```

PBR in a VPRN using an IP filter

Figure 32: PBR in a VPRN shows the example topology used with the "VPRN-2" service configured on each PE and the CEs configured as another VPRN service on PE-1 and PE-4.

Figure 32: PBR in a VPRN



26310

The following IPv4 filter is configured on PE-1:

```

configure {
  filter {
    ip-filter "IP-30" {
      filter-id 30
      entry 10 {
        action {
          forward {
            next-hop {
              nh-ip-vrf {
                router-instance "VPRN-2"
                address 172.16.12.2
              }
            }
          }
          secondary {
            forward {
              next-hop {
                nh-ip-vrf {
                  router-instance "VPRN-2"
                  address 172.16.13.2
                }
              }
            }
          }
        }
      }
    }
  }
}

```

The "VPRN-2" service in PE-1 has the "IP-30" filter applied to SAP 1/1/c3/1:2 toward CE-12:

```

configure {

```

```

service {
  vprn "VPRN-2" {
    admin-state enable
    service-id 2
    customer "1"
    bgp-ipvpn {
      mpls {
        admin-state enable
        route-distinguisher "64496:2"
      }
    }
  }
  interface "int-VPRN-2-PE-1-CE-12" {
    ipv4 {
      primary {
        address 172.31.12.1
        prefix-length 24
      }
    }
    sap 1/1/c3/1:2 {
      ingress {
        filter {
          ip "IP-30"
        }
      }
    }
  }
  interface "int-VPRN-2-PE-1-PE-2" {
    ipv4 {
      primary {
        address 172.16.12.1
        prefix-length 30
      }
    }
    sap 1/1/c1/1:2 {
    }
  }
  interface "int-VPRN-2-PE-1-PE-3" {
    ipv4 {
      primary {
        address 172.16.13.1
        prefix-length 30
      }
    }
    sap 1/1/c2/1:2 {
    }
  }
}
}

```

The configuration of the "VPRN-2" service on the remaining PEs is similar, except that static route entries are configured for subnets 172.31.12.0/24 (toward CE-12) and 172.31.42.0/24 (toward CE-42). No filters are applied to the "VPRN-2" service on the other nodes.

The primary action forwards packets from CE-12 to next-hop 172.16.12.2, which is an interface in the "VPRN-2" service on PE-2; the secondary action forwards to next-hop 172.16.13.2, an interface in the "VPRN-2" service on PE-3. When all interfaces are up, the primary action is executed and traffic from CE-12 to CE-42 is forwarded from the "VPRN-2" router on PE-1 to the "VPRN-2" router on PE-2 (next hop 172.16.12.2), as follows:

```

[/]
A:admin@PE-1# show filter ip "IP-30"
=====

```



```

IP Filter
=====
Filter Id       : 30                               Applied       : Yes
Scope          : Template                         Def. Action   : Drop
---snip---

Primary Action  : Forward (Next Hop VRF)
Next Hop       : 172.16.12.2
Router         : 2
PBR Target Status : Up
Extended Action : None                          # optional DSCP remarking for PBR
Secondary Action : Forward (Next Hop VRF)
Next Hop       : 172.16.13.2
Router         : 2
PBR Target Status : Up
Extended Action : None
PBR Down Action : Drop (entry-default)
Downloaded Action : Primary
Dest. Stickiness : None                          Hold Remain   : 0
Ing. Matches     : 201 pkts (21306 bytes)
Egr. Matches     : 0 pkts

=====

```

The output includes an additional line per action: both the primary and the secondary action in PBR can have DSCP remarking as extended action, but that is not configured in this example. It can be configured using the following command; for example, for the primary action, as follows:

```

*[ex:/configure filter ip-filter "IP-30" entry 10 action remark]
A:admin@PE-1# dscp ?

dscp <keyword>
<keyword> - (be|cp1|cp2|cp3|cp4|cp5|cp6|cp7|cs1|cp9|af11|cp11|af12|cp13|af13|cp15|cs2|cp17|
af21|cp19|af22|
              cp21|af23|cp23|cs3|cp25|af31|cp27|af32|cp29|af33|cp31|cs4|cp33|af41|cp35|af42|
cp37|af43|cp39|
              cs5|cp41|cp42|cp43|cp44|cp45|ef|cp47|nc1|cp49|cp50|cp51|cp52|cp53|cp54|cp55|nc2|
cp57|cp58|
              cp59|cp60|cp61|cp62|cp63)

'dscp' is: mandatory

Destination SAP

```

When the primary action cannot be activated, the secondary action is activated, as follows:

```

# on PE-1:
configure {
  service {
    vprn "VPRN-2" {
      interface "int-VPRN-2-PE-1-PE-2" {
        sap 1/1/c1/1:2 {
          admin-state disable
        }
      }
    }
  }
}

*A:PE-1# show filter ip "IP-30"

=====
IP Filter
=====
Filter Id       : 30                               Applied       : Yes

```

```

Scope           : Template           Def. Action    : Drop
---snip---

Entry           : 10
---snip---

Primary Action  : Forward (Next Hop VRF)
  Next Hop      : 172.16.12.2
  Router        : 2
  PBR Target Status : Down
  Extended Action : None
Secondary Action : Forward (Next Hop VRF)
  Next Hop      : 172.16.13.2
  Router        : 2
  PBR Target Status : Up
  Extended Action : None
PBR Down Action : Drop (entry-default)
Downloaded Action : Secondary
Dest. Stickiness : None                    Hold Remain    : 0
Ing. Matches     : 201 pkts (21306 bytes)
Egr. Matches     : 0 pkts
=====

```

When both PBR targets are down, the default action is drop, because the IPv4 filter does not have the **pbr-down-action-override** parameter configured. Stickiness is not enabled in this filter. The configuration of the IPv4/v6 filters is similar for PBR and PBF.

In the preceding PBR example, the primary and secondary next-hop router is the same VRF "VPRN-2", but it can be any mix of VRFs, such as primary next-hop router 100 and secondary next-hop router 200.

PBR can also steer traffic to the base routing instance; for example, with the following IP filter:

```

configure {
  filter {
    ip-filter "IP-40" {
      filter-id 40
      entry 10 {
        action {
          forward {
            next-hop {
              nh-ip-vrf {
                router-instance "Base"
                address 192.0.2.2
              }
            }
          }
        }
      }
      secondary {
        forward {
          next-hop {
            nh-ip-vrf {
              router-instance "Base"
              address 192.0.2.3
            }
          }
        }
      }
    }
  }
}

```

Conclusion

Operators can define two targets for L2 and L3 traffic steering (PBF and PBR): primary and secondary. The primary target is used when both targets are up; the secondary target is used when the primary is down. However, when stickiness is enabled, it is possible that the secondary action is executed, even when the primary action PBR target reverts to up. When both targets are down, the default action is drop, unless the **pbr-down-action-override** parameter is configured. Both 1+1 redundancy and N+1 redundancy are supported.

Weighted ECMP for 6PE over RSVP-TE LSPs

This chapter provides information about Weighted Equal Cost Multipath (ECMP) for IPv6 Provider Edge (6PE) routers over Resource Reservation Protocol with Traffic Engineering (RSVP-TE) Label Switched Paths (LSPs).

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information and configuration in this chapter are based on SR OS Release 23.3.R2. Weighted ECMP for 6PE routers over RSVP-TE LSPs is supported in SR OS Release 15.0.R6, and later.

Chapter *Weighted ECMP for VPRN over RSVP-TE and SR-TE LSPs* in the "Layer 3 Services" volume of the *7450 ESS, 7750 SR, and 7950 XRS MD-CLI Advanced Configuration Guide - Part II* is recommended reading.

Overview

Equal Load Balancing

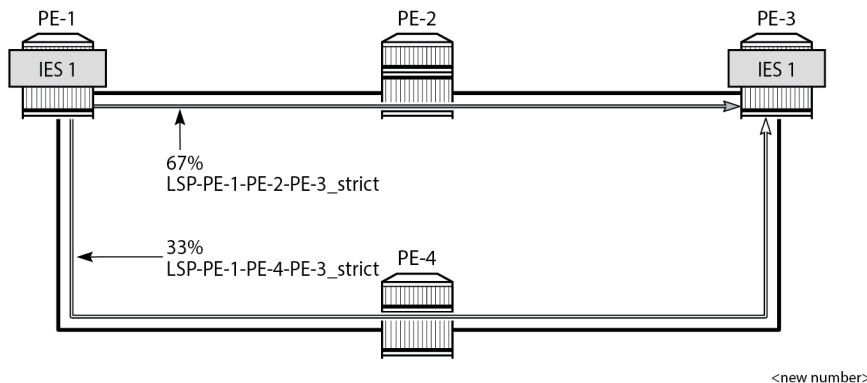
In this chapter, ECMP refers to spraying traffic flows over multiple RSVP-TE LSPs within an ECMP set. ECMP spraying consists of hashing the relevant fields in the packet header and selecting the tunnel next-hop based on the modulo operation of the output of the hash and the number of RSVP-TE LSPs present in the ECMP set. The maximum number of RSVP-TE LSPs in the ECMP set is defined by the **ecmp** command.

Only RSVP-TE LSPs with the same lowest LSP metric can be part of the ECMP set. If the number of such RSVP-TE LSPs exceeds the maximum number of RSVP-TE LSPs allowed in the ECMP set as defined by the **ecmp** command, the RSVP-TE LSPs with the lowest tunnel IDs are selected first. By default, all RSVP-TE LSPs in the ECMP set have the same weight, and traffic flows are spread evenly over all RSVP-TE LSPs in the ECMP set, regardless of the bandwidth of the active path in the RSVP-TE LSPs. By default, ECMP is enabled and set to 1.

Unequal Load Balancing

Weighted ECMP sprays traffic flows over RSVP-TE LSPs proportionally to the **load-balancing-weight** *<weight>* value configured on each RSVP-TE LSP in the ECMP set. [Figure 33: Weighted ECMP in AS 64496](#) shows that PE-1 forwards two thirds of the traffic flows on LSP-PE-1-PE-2-PE-3_strict with weight 2 and one third on LSP-PE-1-PE-4-PE-3_strict with weight 1.

Figure 33: Weighted ECMP in AS 64496



The LSP load balancing weight can be configured in an LSP template or on an RSVP-TE LSP. By default, the load balancing weight equals zero, in which case regular ECMP applies.

Weighted load balancing can be performed only when all the next-hops are associated with the same neighbor and all the RSVP-TE LSPs are configured with a non-zero load balancing weight. If one or more RSVP-TE LSPs in the ECMP set toward a specific next-hop do not have a load balancing weight configured, regular ECMP spraying is used.

The following command is used to configure the weight in an LSP template:

```
configure {
  router "Base" {
    mpls {
      lsp-template "LSPtemplate1" {
        load-balancing-weight ?
      }
    }
  }
}

load-balancing-weight <number>
<number> - <1..4294967295>

Load balancing weight for an MPLS LSP template

Warning: Modifying this element toggles
'configure router "Base" mpls lsp-template "LSPtemplate1" admin-state' automatically
for the new value to take effect.
```

The following command is used to configure the weight on an LSP (for example on "LSP-PE-1-PE-2-PE-3_strict"):

```
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-2-PE-3_strict" {
        load-balancing-weight ?
      }
    }
  }
}

load-balancing-weight <number>
<number> - <1..4294967295>

Load balancing weight for an MPLS LSP
```

The LSP load balancing weight on LSP-PE-1-PE-2-PE-3_strict is configured with a value of 2, as follows:

```
configure {
```

```

router "Base" {
  mpls {
    path "path-PE-1-PE-2-PE-3_strict" {
      admin-state enable
      hop 10 {
        ip-address 192.168.12.2
        type strict
      }
      hop 20 {
        ip-address 192.168.23.2
        type strict
      }
    }
    lsp "LSP-PE-1-PE-2-PE-3_strict" {
      admin-state enable
      type p2p-rsvp
      to 192.0.2.3
      path-computation-method local-cspf
      metric 100
      load-balancing-weight 2
      primary "path-PE-1-PE-2-PE-3_strict" {
      }
    }
  }
}

```

Weighted ECMP for 6PE over RSVP-TE LSPs is enabled in the **bgp next-hop-resolution** context as follows:

```

configure {
  router "Base" {
    bgp {
      next-hop-resolution {
        weighted-ecmp true
      }
    }
  }
}

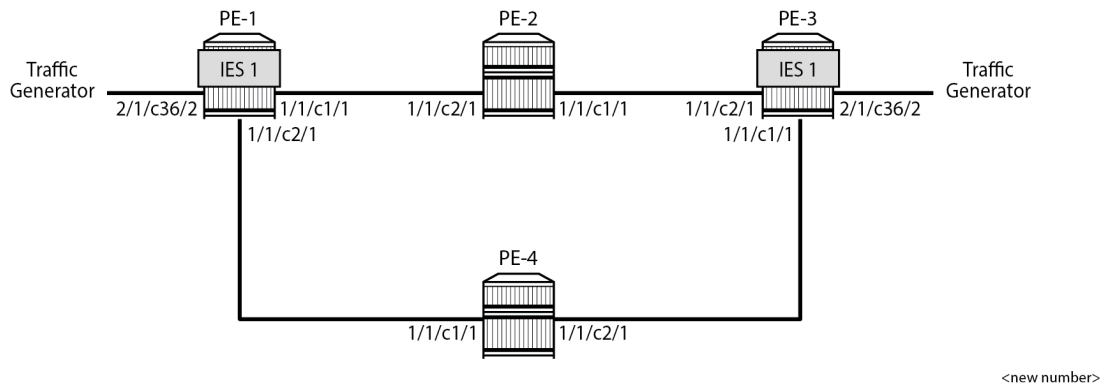
```

The **weighted-ecmp** option controls load balancing to the same next-hop only.

Configuration

[Figure 34: Example Topology for 6PE over RSVP-TE LSPs](#) shows the example topology with four PEs. IES 1 is configured on PE-1 and PE-3. A traffic generator is connected to IES 1 SAP 2/1/c36/2 on PE-1 and IES 1 SAP 2/1/c36/2 on PE-3. The traffic generator generates multiple IPv6 traffic flows with random IP addresses and TCP/UDP port numbers. As a result, these flows are sprayed over different MPLS LSPs between PE-1 and PE-3.

Figure 34: Example Topology for 6PE over RSVP-TE LSPs



Initial Configuration

The initial configuration on the PEs includes the following:

- Cards, MDAs, ports
- Router interfaces
- IS-IS as IGP (alternatively, OSPF can be used) with traffic engineering enabled
- MPLS and RSVP enabled on all router interfaces
- MPLS paths with strict hops from PE-1 to PE-3 and the other way around: one via PE-2 and the other via PE-4. The LSP via PE-2 gets a load balancing weight of 2, whereas the LSP via PE-4 gets a load balancing weight of 1. Both LSPs have the same metric.

The initial configuration on PE-1 is as follows.

```
configure {
  router "Base" {
    interface "int-PE-1-PE-2" {
      port 1/1/c1/1
      ipv4 {
        primary {
          address 192.168.12.1
          prefix-length 30
        }
      }
    }
    interface "int-PE-1-PE-4" {
      port 1/1/c2/1
      ipv4 {
        primary {
          address 192.168.14.1
          prefix-length 30
        }
      }
    }
  }
  interface "system" {
    ipv4 {
      primary {
        address 192.0.2.1
        prefix-length 32
      }
    }
  }
}
```

```
    }
  }
}
isis 0 {
  admin-state enable
  area-address [49.0001]
  traffic-engineering true
  interface "system" {
  }
  interface "int-PE-1-PE-2" {
    interface-type point-to-point
  }
  interface "int-PE-1-PE-4" {
    interface-type point-to-point
  }
}
mpls {
  admin-state enable
  interface "int-PE-1-PE-2" {
  }
  interface "int-PE-1-PE-4" {
  }
  path "path-PE-1-PE-2-PE-3_strict" {
    admin-state enable
    hop 10 {
      ip-address 192.168.12.2
      type strict
    }
    hop 20 {
      ip-address 192.168.23.2
      type strict
    }
  }
  path "path-PE-1-PE-4-PE-3_strict" {
    admin-state enable
    hop 10 {
      ip-address 192.168.14.2
      type strict
    }
    hop 20 {
      ip-address 192.168.34.1
      type strict
    }
  }
}
lsp "LSP-PE-1-PE-2-PE-3_strict" {
  admin-state enable
  type p2p-rsvp
  to 192.0.2.3
  path-computation-method local-cspf
  metric 100
  load-balancing-weight 2
  primary "path-PE-1-PE-2-PE-3_strict" {
  }
}
lsp "LSP-PE-1-PE-4-PE-3_strict" {
  admin-state enable
  type p2p-rsvp
  to 192.0.2.3
  path-computation-method local-cspf
  metric 100
  load-balancing-weight 1
  primary "path-PE-1-PE-4-PE-3_strict" {
  }
}
}
```



```

}
  rsvp {
    admin-state enable
    interface "int-PE-1-PE-2" {
    }
    interface "int-PE-1-PE-4" {
    }
  }
}

```

The configuration on PE-3 is similar.

With the preceding configuration, MPLS and RSVP are enabled on all interfaces, including the system interface, which is added automatically.

Weighted ECMP for 6PE over RSVP-TE LSPs

BGP is configured for the label-IPv6 address family and the next-hop resolution is set to RSVP; see the [6PE Next-Hop Resolution](#) chapter.

In this example, the traffic generator sends IPv6 traffic to the SAP in IES 1. The IPv6 packets are tunneled through the IPv4 network between PE-1 and PE-3. The service configuration on PE-1 is as follows:

```

configure {
  service {
    ies "IES-1" {
      admin-state enable
      service-id 1
      customer "1"
      description "6PE-1"
      interface "int-PE-1-STC" {
        sap 2/1/c36/2 {
        }
      }
      ipv6 {
        address 2001:db8::11:1 {
          prefix-length 120
        }
      }
    }
  }
}

```

The configuration on PE-3 is similar.

On PE-1, the following BGP configuration defines next-hop resolution with weighted ECMP and the resolution filter only allows RSVP-TE LSPs. BGP is configured for the label-IPv6 address family and BGP multipath is configured in the **bgp** context.

```

configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      ibgp-multipath true
      split-horizon true
      next-hop-resolution {
        weighted-ecmp true
        labeled-routes {
          transport-tunnel {
            family label-ipv6 {
              resolution-filter {
                ldp false
                rsvp true
              }
            }
          }
        }
      }
    }
  }
}

```

```

    }
  }
  group "iBGP" {
    path-mtu-discovery true
    peer-as 64496
    export {
      policy ["export-6PE-1"]
    }
  }
  neighbor 192.0.2.3 {
    group "iBGP"
    family {
      label-ipv6 true
    }
  }
}

```

The configuration on PE-3 is similar.

On PE-1 and PE-3, the following export policy is configured:

```

configure {
  policy-options {
    policy-statement "export-6PE-1" {
      entry 10 {
        from {
          protocol {
            name [direct]
          }
        }
        action {
          action-type accept
        }
      }
      default-action {
        action-type reject
      }
    }
  }
}

```

The following command enables ECMP in the base router.

```

configure {
  router "Base" {
    ecmp 2
  }
}

```

On PE-1, the route table in the base router shows that the remote prefix 2001:db8::33:0/120 has flag [2], meaning that the next-hop 192.0.2.3 occurs twice for this prefix, as follows:

```

[/]
A:admin@PE-1# show router route-table 2001:db8::33:0/120
=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]          Metric
-----
2001:db8::33:0/120 [2]            Remote BGP_LABEL 00h01m10s 170
    192.0.2.3 (tunneled:RSVP:2)      100
2001:db8::33:0/120 [2]            Remote BGP_LABEL 00h01m10s 170
    192.0.2.3 (tunneled:RSVP:4)      100
-----

```

```
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

The route table on PE-3 shows a similar route with flag [2] for prefix 2001:db8::11:0/120.

On PE-1, the following detailed route table info (using keyword **extensive**) for prefix 2001:db8::33:0/120 shows that RSVP-TE tunnel 2 and RSVP-TE tunnel 4 are used to reach the next-hop 192.0.2.3. Both RSVP-TE tunnels have metric 100, but the weight of RSVP-TE tunnel 2 is twice as much as the weight of RSVP tunnel 4, so the load on RSVP-TE LSP 2 is twice as high as the load on RSVP LSP 4.

```
[/]
A:admin@PE-1# show router route-table 2001:db8::33:0/120 extensive
```

```
=====
Route Table (Router: Base)
=====
Dest Prefix      : 2001:db8::33:0/120
Protocol         : BGP_LABEL
Age              : 00h01m10s
Preference       : 170
Indirect Next-Hop : 192.0.2.3
  Label          : 2
  QoS             : Priority=n/c, FC=n/c
  Source-Class    : 0
  Dest-Class      : 0
  ECMP-Weight     : N/A
  Resolving Next-Hop : 192.0.2.3 (RSVP tunnel:2)
    Metric        : 100
    ECMP-Weight   : 2
  Resolving Next-Hop : 192.0.2.3 (RSVP tunnel:4)
    Metric        : 100
    ECMP-Weight   : 1
-----
No. of Destinations: 1
=====
```

The following tunnel table on PE-1 shows that RSVP-TE tunnel 2 has PE-2 as next-hop (192.168.12.2) and RSVP-TE tunnel 4 has next-hop PE-4 (192.168.14.2):

```
[/]
A:admin@PE-1# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.3/32     rsvp      MPLS 2          7     192.168.12.2  100
192.0.2.3/32     rsvp      MPLS 4          7     192.168.14.2  100
---snip---
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
```

Traffic Verification

The traffic generator sends IPv6 traffic flows to SAP 2/1/c36/2 of IES 1 on PE-1. The packets are tunneled over the available RSVP-TE LSPs present in the ECMP set. The traffic is load balanced unevenly: two thirds of the traffic flows is tunneled via PE-2 (port 1/1/c1/1) while one third of the traffic flows is tunneled via PE-4 (port 1/1/c2/1). The load on the ports is as follows:

```
[/]
A:admin@PE-1# monitor port 1/1/c1/1 rate interval 3 repeat 3

=====
Monitor statistics for Port 1/1/c1/1
=====
-----
Input                               Output
-----snip-----
-----
At time t = 6 sec (Mode: Rate)
-----
Octets                               101                               444150
Packets                             1                                 431
Errors                                0                                 0
Bits                                  808                               3553200
Utilization (% of port capacity)     ~0.00                             0.03
---snip---
=====

[/]
A:admin@PE-1# monitor port 1/1/c2/1 rate interval 3 repeat 3

=====
Monitor statistics for Port 1/1/c2/1
=====
-----
Input                               Output
-----snip-----
-----
At time t = 6 sec (Mode: Rate)
-----
Octets                               226                               186190
Packets                             2                                 182
Errors                                0                                 0
Bits                                  1808                              1489520
Utilization (% of port capacity)     ~0.00                             0.01
---snip---
=====

[/]
A:admin@PE-1# monitor port 2/1/c36/2 rate interval 3 repeat 3

=====
Monitor statistics for Port 2/1/c36/2
=====
-----
Input                               Output
-----snip-----
-----
At time t = 6 sec (Mode: Rate)
-----
Octets                               602112                             0
Packets                             588                             0
```

```

Errors                0                0
Bits                 4816896         0
Utilization (% of port capacity)  0.04          0.00
---snip---
=====

```

This can also be verified as follows:

```

[/]
A:admin@PE-1# show port 1/1/c1/1 statistics

=====
Port Statistics on Slot 1
=====
Port Id                Ingress Packets      Ingress Octets
                Egress Packets      Egress Octets
-----
1/1/c1/1                47                    4863
                        14730                15157578
=====

[/]
A:admin@PE-1# show port 1/1/c2/1 statistics

=====
Port Statistics on Slot 1
=====
Port Id                Ingress Packets      Ingress Octets
                Egress Packets      Egress Octets
-----
1/1/c2/1                44                    4423
                        6266                6424681
=====

[/]
A:admin@PE-1# show port 2/1/c36/2 statistics

=====
Port Statistics on Slot 2
=====
Port Id                Ingress Packets      Ingress Octets
                Egress Packets      Egress Octets
-----
2/1/c36/2                20904                 21405696
                        0                      0
=====

```

Conclusion

Operators can control how 6PE traffic is load balanced unequally over multiple RSVP-TE LSPs by defining a load balancing weight value on each LSP.

Unicast Routing Protocols

This section provides configuration information for the following topics:

- [Advertising IPv4 NLRI with IPv6 Next-Hop](#)
- [Associating Communities with Static and Aggregate Routes](#)
- [BGP Add-Path](#)
- [BGP Add-Path Policy Control](#)
- [BGP Autonomous System Override](#)
- [BGP Conditional Route Advertisement](#)
- [BGP Convergence — Delayed Route Advertisement](#)
- [BGP Default Route Origination](#)
- [BGP Fast Reroute](#)
- [BGP Fast Reroute Policy Control](#)
- [BGP FlowSpec for IPv4 and IPv6](#)
- [BGP Multipath](#)
- [BGP Optimal Route Reflection for Hierarchical Networks](#)
- [BGP Optimal Route Reflection for Non-Hierarchical Networks](#)
- [BGP Prefix Limit per Address Family](#)
- [BGP Remove-Private ASN](#)
- [BGP Route Leaking](#)
- [BGP Route Refresh](#)
- [BGP Unresolved Route Leaking from Base Router to VPRN](#)
- [BGP Weighted ECMP](#)
- [Dynamic BGP Peers](#)
- [EBGP Default Reject Policy](#)
- [EBGP Route Resolution to a Static Route](#)
- [Flexible Algorithm for IS-IS](#)
- [IS-IS Link Bundling](#)
- [Next-Hop Resolution for Labeled BGP Routes](#)
- [Policy Chaining and Logical Expressions](#)
- [Pop-Label for /32 Label-IPv4 BGP Routes](#)
- [Route Policy Action to Suppress BGP Route Installation](#)
- [Separate BGP RIBs for Labeled Routes](#)

Advertising IPv4 NLRI with IPv6 Next-Hop

This chapter describes Advertising IPv4 NLRI with IPv6 Next-Hop.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

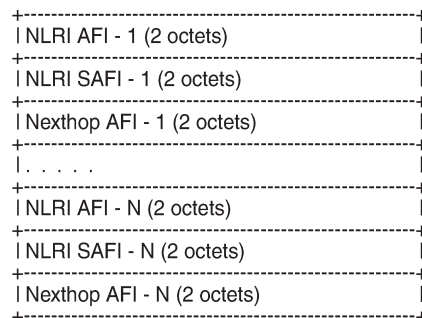
The information and MD-CLI configuration in this chapter are based on SR OS Release 20.7.R2. Advertising IPv4 Network Layer Reachability Information (NLRI) with IPv6 next-hop is supported in SR OS Release 19.5.R1 and later.

Overview

In networks where the routers are interconnected by IPv6-only links, SR OS routers can advertise and receive BGP routes that convey reachability to IPv4-unicast destinations that are reachable through IPv6 next-hops. Advertising and receiving IPv4 routes with IPv6 next-hops is useful in networks or regions with IPv6-only interfaces, such as data center deployments where leaf, spine, and aggregation routers are interconnected by IPv6-only links that carry a mix of unencapsulated IPv4 and IPv6 packets.

This feature requires the Extended Next Hop encoding BGP capability which is described in RFC 5549, *Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop*. BGP capabilities are advertised between peers. For the Extended Next Hop encoding capability, the capability code field must be set to 5, the capability length field set to the length of the capability value field, and a capability value field with following format:

Figure 35: Capability value field format



36526

Each triplet (NLRI AFI, NLRI SAFI, Nexthop AFI) indicates that NLRI AFI/SAFI may be advertised with a next-hop address belonging to the network-layer protocol of "Nexthop AFI".

By default, IPv4-unicast routes are advertised with IPv4 next-hops. However, on IPv6-only TCP transport sessions, IPv4-unicast routes can be advertised with IPv6 next-hops if the **advertise-ipv6-next-hops** command with the **ipv4** option applies to the session. The **advertise-ipv6-next-hops** command can be enabled for several address families, as follows:

```
[ex:configure router "Base" bgp]
A:admin@PE-1# advertise-ipv6-next-hops

advertise-ipv6-next-hops

evpn          - Advertise EVPN routes with IPv6 addresses
ipv4          - Advertise IPv4 routes with IPv6 addresses
label-ipv4    - Advertise label IPv4 routes with IPv6 addresses
label-ipv6    - Advertise label IPv6 routes with IPv6 addresses
vpn-ipv4      - Advertise VPN IPv4 routes with IPv6 addresses
vpn-ipv6      - Advertise VPN IPv6 routes with IPv6 addresses
```

For receiving IPv4-unicast routes with IPv6 next-hop addresses, the **extended-nh-encoding** command with the **ipv4** option must be applied to the session. This advertises the RFC 5549 capability to the peer for the different address families. The **extended-nh-encoding** command can be configured for several address families, as follows:

```
[ex:configure router "Base" bgp]
A:admin@PE-1# extended-nh-encoding

extended-nh-encoding

ipv4          - Advertise extended NH encoding for IPv4 routes
label-ipv4    - Advertise extended NH encoding for label-IPv4 routes
vpn-ipv4      - Advertise extended NH encoding for VPN-IPv4 routes
```

When the BGP session is established, the BGP peers advertise the capability to each other, and the Extended Next Hop encoding capability is both a local and a remote capability, as in the following example between BGP peers 2001:db8::12:1 and 2001:db8::12:2:

```
[ ]
A:admin@PE-1# show router bgp neighbor 2001:db8::12:2 | match Capability post-lines 1
Local Capability      : RtRefresh MPBGP 4byte ASN
                     : EXT_NH_ENCODING
Remote Capability    : RtRefresh MPBGP 4byte ASN
                     : EXT_NH_ENCODING
```

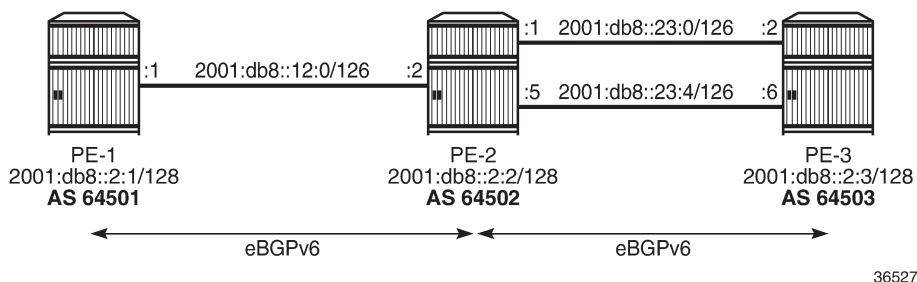
When **next-hop-self** applies to the BGP session and the neighbor address is IPv6, an IPv4-unicast route that is advertised or re-advertised gets the following as next-hop:

- The IPv6 local address used for peering, if the peer opened the BGP session by advertising an extended next-hop encoding capability with NLRI AFI=1, SAFI=1, and nexthop AFI=2, and the session is associated with an **advertise-ipv6-next-hops ipv4** command.
- The IPv4 system interface address in all other cases.

Configuration

[Figure 36: Example topology with IPv6 interfaces](#) shows the example topology with three nodes with IPv6-only interfaces in different Autonomous Systems (ASs).

Figure 36: Example topology with IPv6 interfaces

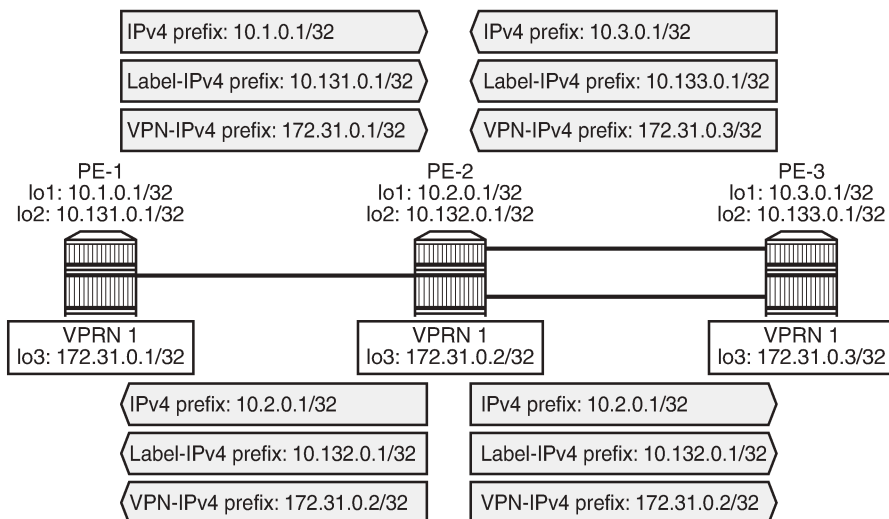


The initial configuration includes:

- Cards, MDAs, ports
- Router interfaces with IPv6 addresses

In the example, IPv4, label-IPv4, and VPN-IPv4 routes will be advertised with an IPv6 next-hop. On PE-1, loopback interfaces lo1 (10.1.0.1/32) and lo2 (10.131.0.1/32) are configured; lo1 will be advertised as an IPv4 route and lo2 as a label-IPv4 route. VPRN 1 is configured on all nodes with loopback interface lo3, and prefix 172.31.0.1/32 will be advertised as a VPN-IPv4 route on PE-1. PE-2 and PE-3 have similar loopback interfaces. [Figure 37: Loopback addresses and advertised IPv4, label-IPv4, and VPN-IPv4 routes](#) shows the loopback addresses and the advertised routes.

Figure 37: Loopback addresses and advertised IPv4, label-IPv4, and VPN-IPv4 routes



36528

On PE-2, eBGP is configured toward three IPv6 neighbors with **next-hop-self** enabled. For each of the BGP neighbors, **extended-nh-encoding** and **advertise-ipv6-next-hops** are configured for different address families. The BGP configuration is as follows:

```
# on PE-2:
configure {
  router "Base" {
    bgp {
```

```
loop-detect discard-route
inter-as-vpn true
split-horizon true
ebgp-default-reject-policy {
    import false
    export false
}
multipath {
    max-paths 2
    ebgp 2
}
group "eBGP-IPv6" {
    family {
        ipv4 true
        vpn-ipv4 true
        ipv6 true
        vpn-ipv6 true
        label-ipv4 true
        label-ipv6 true
    }
    local-as {
        as-number 64502
    }
    import {
        policy ["import-1:1-3:3"]
    }
    export {
        policy ["export-10.2" "export-10.132"]
    }
}
neighbor "2001:db8::12:1" {
    group "eBGP-IPv6"
    next-hop-self true
    peer-as 64501
    extended-nh-encoding {
        vpn-ipv4 true
        label-ipv4 true
        ipv4 true
    }
    advertise-ipv6-next-hops {
        vpn-ipv6 true
        label-ipv6 true
        vpn-ipv4 true
        label-ipv4 true
        ipv4 true
    }
}
neighbor "2001:db8::23:2" {
    group "eBGP-IPv6"
    next-hop-self true
    peer-as 64503
    extended-nh-encoding {
        vpn-ipv4 true
        label-ipv4 true
        ipv4 true
    }
    advertise-ipv6-next-hops {
        vpn-ipv6 true
        label-ipv6 true
        vpn-ipv4 true
        label-ipv4 true
        ipv4 true
    }
}
```

```

neighbor "2001:db8::23:6" {
  group "eBGP-IPv6"
  next-hop-self true
  peer-as 64503
  extended-nh-encoding {
    vpn-ipv4 true
    label-ipv4 true
    ipv4 true
  }
  advertise-ipv6-next-hops {
    vpn-ipv6 true
    label-ipv6 true
    vpn-ipv4 true
    label-ipv4 true
    ipv4 true
  }
}

```

The BGP configuration on PE-1 and PE-3 is similar.

The BGP summary on PE-1 shows that for each of the configured address families, one route is advertised and two routes are received and accepted:

```

[]
A:admin@PE-1# show router bgp summary all

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId          AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
                   PktSent OutQ
-----
2001:db8::12:2
Def. Instance 64502      22   0 00h02m38s 2/2/1 (IPv4)
                   16   0                2/2/1 (IPv6)
                   2/2/1 (VpnIPv4)
                   2/2/1 (VpnIPv6)
                   2/2/1 (Lbl-IPv4)
                   2/2/1 (Lbl-IPv6)
-----

```

On PE-1, the following IPv4 routes with IPv6 next-hop are received and used: route 10.2.0.1/32 originates from PE-2 and route 10.3.0.1/32 from PE-3. Both routes have next-hop 2001:db8::12:2 because next-hop-self is enabled, as follows:

```

[]
A:admin@PE-1# show router bgp routes

=====
BGP Router ID:255.0.0.0      AS:64501      Local AS:64501
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes

```

```

=====
Flag Network                               LocalPref MED
      Nexthop (Router)                    Path-Id   IGP Cost
      As-Path                               Label
-----
u*>i 10.2.0.1/32                            None     None
      2001:db8::12:2                        None     0
      64502                                  -
u*>i 10.3.0.1/32                            None     None
      2001:db8::12:2                        None     0
      64502 64503                          -
-----
Routes : 2
=====

```

On PE-2, the following VPN-IPv4 routes with different IPv6 next-hops are received and used:

```

[]
A:admin@PE-2# show router bgp routes vpn-ipv4
=====
BGP Router ID:255.0.0.0      AS:64502      Local AS:64502
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag Network                               LocalPref MED
      Nexthop (Router)                    Path-Id   IGP Cost
      As-Path                               Label
-----
u*>i 64501:1:172.31.0.1/32                  None     None
      2001:db8::12:1                        None     0
      64501                                  524287
u*>i 64503:1:172.31.0.3/32                  None     None
      2001:db8::23:2                        None     0
      64503                                  524287
u*>i 64503:1:172.31.0.3/32                  None     None
      2001:db8::23:6                        None     0
      64503                                  524287
-----
Routes : 3
=====

```

On PE-3, the following label-IPv4 routes with IPv6 next-hop are received and used. Route 10.131.0.1/32 originates from PE-1 and is re-advertised by PE-2 on two eBGP paths, with next-hop addresses 2001:db8::23:1 and 2001:db8::23:5. Route 10.132.0.1/32 originates from PE-2 and is also advertised over these two eBGP paths.

```

[]
A:admin@PE-3# show router bgp routes label-ipv4
=====
BGP Router ID:255.0.0.0      AS:64503      Local AS:64503
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete

```

```

=====
BGP Routes
=====
Flag Network                               LocalPref MED
  Nexthop (Router)                        Path-Id   IGP Cost
  As-Path                                  Label
-----
u*>i 10.131.0.1/32                          None     None
      2001:db8::23:1                         None     0
      64502 64501                            524284
u*>i 10.131.0.1/32                          None     None
      2001:db8::23:5                         None     0
      64502 64501                            524284
u*>i 10.132.0.1/32                          None     None
      2001:db8::23:1                         None     0
      64502                                  524285
u*>i 10.132.0.1/32                          None     None
      2001:db8::23:5                         None     0
      64502                                  524285
-----
Routes : 4
=====

```

The route table on PE-3 includes BGP IPv4 and label-IPv4 routes with IPv6 next-hops, as follows:

```

[]
A:admin@PE-3# show router route-table
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                        Type  Proto  Age           Pref
  Next Hop[Interface Name]                Metric
-----
10.1.0.1/32                               Remote BGP    00h03m36s 170
      2001:db8::23:1                       0
10.1.0.1/32                               Remote BGP    00h03m36s 170
      2001:db8::23:5                       0
10.2.0.1/32                               Remote BGP    00h03m36s 170
      2001:db8::23:1                       0
10.2.0.1/32                               Remote BGP    00h03m36s 170
      2001:db8::23:5                       0
10.3.0.1/32                               Local  Local   00h04m20s 0
      lo1                                   0
10.131.0.1/32                             Remote BGP_LABEL 00h03m36s 170
      2001:db8::23:1                       0
10.131.0.1/32                             Remote BGP_LABEL 00h03m36s 170
      2001:db8::23:5                       0
10.132.0.1/32                             Remote BGP_LABEL 00h03m36s 170
      2001:db8::23:1                       0
10.132.0.1/32                             Remote BGP_LABEL 00h03m36s 170
      2001:db8::23:5                       0
10.133.0.1/32                             Local  Local   00h04m20s 0
      lo2                                   0
192.0.2.3/32                             Local  Local   00h04m20s 0
      system                               0
-----
No. of Routes: 11

```

The tunnel table on PE-3 shows four BGP tunnels with IPv6 next-hops, as follows:

```

[]
A:admin@PE-3# show router tunnel-table

```

```

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner    Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
10.131.0.1/32    bgp      MPLS  262146   12    2001:db8::23:1 1000
10.131.0.1/32    bgp      MPLS  262146   12    2001:db8::23:5 1000
10.132.0.1/32    bgp      MPLS  262145   12    2001:db8::23:1 1000
10.132.0.1/32    bgp      MPLS  262145   12    2001:db8::23:5 1000
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====

```

The route table for VPRN 1 on PE-3 includes BGP VPN-IPv4 routes with IPv6 next-hops, as follows:

```

[]
A:admin@PE-3# show router 1 route-table
=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]      Type  Proto  Age      Pref
  Next Hop[Interface Name]  Metric
-----
172.31.0.1/32          Remote BGP VPN 00h03m36s 170
    2001:db8::23:1      0
172.31.0.1/32          Remote BGP VPN 00h03m36s 170
    2001:db8::23:5      0
172.31.0.2/32          Remote BGP VPN 00h03m36s 170
    2001:db8::23:1      0
172.31.0.2/32          Remote BGP VPN 00h03m36s 170
    2001:db8::23:5      0
172.31.0.3/32          Local  Local  00h04m20s 0
    lo3                  0
-----
No. of Routes: 5

```

The reachability between source address 172.31.0.3 and destination 172.31.0.1 can be verified, but the following traceroute does not display any address for the intermediate node:

```

[]
A:admin@PE-3# traceroute 172.31.0.1 router-instance "VPRN 1"
                                     source-address 172.31.0.3
traceroute to 172.31.0.1 from 172.31.0.3, 30 hops max, 40 byte packets
 1 0.0.0.0 * * *
 2 172.31.0.1 (172.31.0.1) 3.68 ms 3.51 ms 3.78 ms

```

However, the following traceroute from lo1 on PE-3 to lo1 on PE-1 fails:

```

[]
A:admin@PE-3# traceroute 10.1.0.1 source-address 10.3.0.1
traceroute to 10.1.0.1 from 10.3.0.1, 30 hops max, 40 byte packets
 1 0.0.0.0 * * *
 2 0.0.0.0 * * *
 3 0.0.0.0 * * *
 4 0.0.0.0 * * *
 5 0.0.0.0 * * * traceroute aborted by user

```

Likewise, the traceroute from lo2 on PE-3 to PE-1 will fail (not shown here).

In an IPv6-only network, the IPv4 interfaces are down, as follows:

```
[ ]
A:admin@PE-2# show router interface

=====
Interface Table (Router: Base)
=====
Interface-Name      Adm    Opr(v4/v6)  Mode    Port/SapId
IP-Address          PfxState
-----
int-PE-2-PE-1      Up     Down/Up     Network 1/1/2:100
2001:db8::12:2/126  PREFERRED
fe80::14:1ff:fe01:2/64 PREFERRED
int-PE-2-PE-3_0    Up     Down/Up     Network 1/1/1:100
2001:db8::23:1/126  PREFERRED
fe80::14:1ff:fe01:1/64 PREFERRED
int-PE-2-PE-3_4    Up     Down/Up     Network 1/1/3:100
2001:db8::23:5/126  PREFERRED
fe80::14:1ff:fe01:3/64 PREFERRED
lo1                 Up     Up/Up       Network loopback
10.2.0.1/32         n/a
2001:db8::10:2:0:1/128 PREFERRED
fe80::13:fff:fe00:0/64 PREFERRED
lo2                 Up     Up/Up       Network loopback
10.132.0.1/32      n/a
2001:db8::10:132:0:1/128 PREFERRED
fe80::13:fff:fe00:0/64 PREFERRED
system              Up     Up/Up       Network system
192.0.2.2/32       n/a
2001:db8::2:2/128  PREFERRED
-----
Interfaces : 6
=====
```

To allow CPM-originated or terminated packets, such as IPv4 ping or traceroute traffic, the **forward-ipv4-packets** command is configured in the **ipv6** context of these interfaces, as follows:

```
# on PE-2:
configure {
  router "Base" {
    interface "int-PE-2-PE-1" {
      port 1/1/2:100
      ipv6 {
        forward-ipv4-packets true
        address 2001:db8::12:2 {
          prefix-length 126
        }
      }
    }
    interface "int-PE-2-PE-3_0" {
      port 1/1/1:100
      ipv6 {
        forward-ipv4-packets true
        address 2001:db8::23:1 {
          prefix-length 126
        }
      }
    }
    interface "int-PE-2-PE-3_4" {
      port 1/1/3:100
    }
  }
}
```

```

        ipv6 {
            forward-ipv4-packets true
            address 2001:db8::23:5 {
                prefix-length 126
            }
        }
    }
}

```

The connectivity between the lo1 and lo2 interfaces can now be verified from PE-3, as follows:

```

[]
A:admin@PE-3# traceroute 10.1.0.1 source-address 10.3.0.1
traceroute to 10.1.0.1 from 10.3.0.1, 30 hops max, 40 byte packets
 1 10.2.0.1 (10.2.0.1)  2.02 ms  2.65 ms  2.55 ms
 2 10.1.0.1 (10.1.0.1)  3.85 ms  3.69 ms  3.73 ms

```

```

[]
A:admin@PE-3# traceroute 10.2.0.1 source-address 10.3.0.1
traceroute to 10.2.0.1 from 10.3.0.1, 30 hops max, 40 byte packets
 1 10.2.0.1 (10.2.0.1)  2.39 ms  2.70 ms  2.94 ms

```

```

[]
A:admin@PE-3# traceroute 10.131.0.1 source-address 10.133.0.1
traceroute to 10.131.0.1 from 10.133.0.1, 30 hops max, 40 byte packets
 1 10.2.0.1 (10.2.0.1)  2.38 ms  2.91 ms  2.54 ms
 2 10.131.0.1 (10.131.0.1)  3.74 ms  3.58 ms  5.38 ms

```

```

[]
A:admin@PE-3# traceroute 10.132.0.1 source-address 10.133.0.1
traceroute to 10.132.0.1 from 10.133.0.1, 30 hops max, 40 byte packets
 1 10.132.0.1 (10.132.0.1)  2.43 ms  2.89 ms  2.92 ms

```

With the **forward-ipv4-packets** command, the IOM is instructed by the CPM to consider the IPv4 operational state of the interface as up when the IPv6 interface is operationally up. IPv4 packets can be sent and received on the interface when the IPv6 interface is up, even when the IPv4 interface is operationally down.

Conclusion

SR OS routers can advertise and receive BGP routes for IPv4 destinations with IPv6 next-hops. This feature requires the Extended Next Hop encoding BGP capability in RFC 5549 and is useful in IPv6-only networks or regions.

Associating Communities with Static and Aggregate Routes

This chapter provides information about associating communities with static and aggregate routes configurations.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 11.0.R3, but the MD-CLI in this edition corresponds to SR OS Release 20.7.R2. There are no prerequisites for this configuration.

Introduction

Border gateway protocol (BGP) communities are optional, transitive attributes attached to BGP route prefixes to carry more information about that route prefix. Multiple route prefixes can have the same community attached such that it can be matched by a route policy. As a result, the presence of a community value can be used to influence and control route policies.

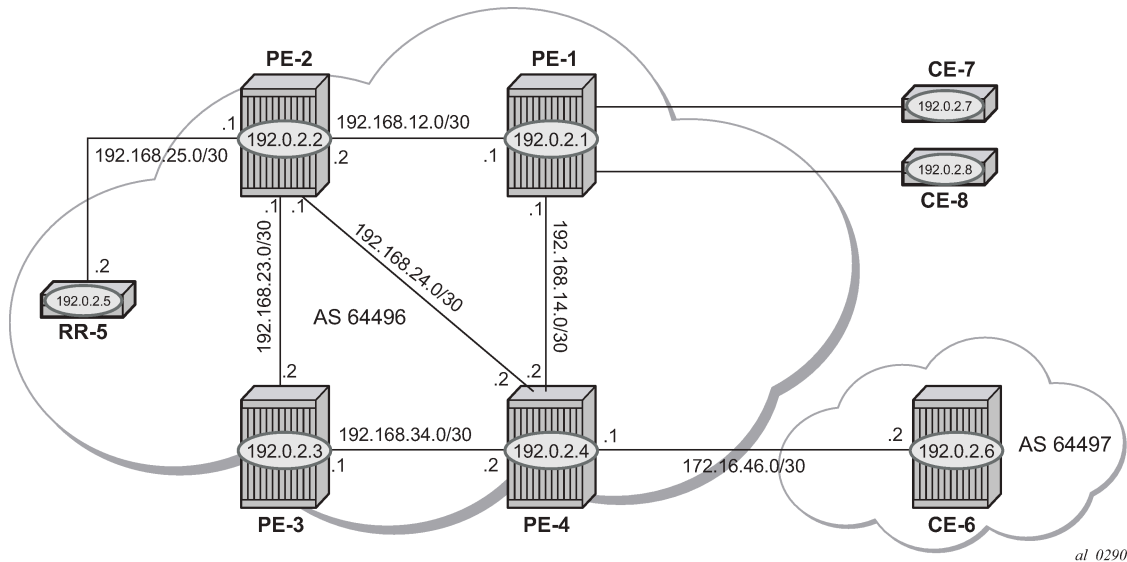
A BGP community is a 32-bit value that is written as two 16-bit numbers separated by a colon. The first number usually represents the autonomous system (AS) number that defines or originates the community while the second is set by the network administrator.

Knowledge of RFC 4271 (BGP-4) and RFC 1997 (BGP Communities Attribute) is assumed throughout this document, as well as knowledge of multi-protocol BGP (MP-BGP) and RFC 4364 (BGP/MPLS IP VPNs).

Overview

[Figure 38: Example topology](#) shows the example topology with 7750 Server Router nodes. PE-1 to PE-4 and the Route Reflector (RR-5) are located in the same Autonomous System (AS): AS 64496. CE-6 is in a separate AS 64497 and peers using eBGP with its directly connected neighbor, PE-4.

Figure 38: Example topology



The objectives are:

- To configure static routes in a VPRN in PE-1 with various community values—including well-known communities—export them to other PEs within the same AS, and then via eBGP to CE-6. During this process, the community values for each route will be examined to ensure that the transitive nature of the attribute is maintained.
- To associate a community with an aggregate route that represents a larger number of composite prefixes. The aggregate will be advertised in place of the composite prefixes.

The following configuration tasks should be completed as a prerequisite:

- Full mesh IS-IS or OSPF between all of the PE routers and the RR.
- iBGP between the RR and all PEs.
- eBGP between PE-4 and CE-6.
- Link-layer LDP between all PEs.

Associating communities with static and aggregate routes

It is possible to add a single community value to a static and aggregate route without using a route policy.

The community value can be in the 4-byte format comprising of a 2-byte AS value, followed by a 2-byte decimal value, separated by a colon. It can also be the name of a well-known standard community, such as: no-export, no-advertise, no-export-subconfed.

Any community added can be matched using a route policy.

The purpose of this example is to provision static and aggregate IPv4 route prefixes and associate a community with each route. These routes are then redistributed into the BGP protocol and advertised to other BGP speakers.

This is shown for IPv4 routes within a VPRN. Well-known, standard communities will also be configured to show that the correct behavior is observed.

Configuration

The first step is to configure an iBGP session between each of the PEs and the Route Reflector (RR). The address family negotiated between peers is VPN-IPv4.

The following BGP configuration is identical for all PEs:

```
# on all PEs:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      group "internal" {
        peer-as 64496
        family {
          vpn-ipv4 true
        }
      }
      neighbor "192.0.2.5" {
        group "internal"
      }
    }
  }
}
```

The IP addresses can be derived from [Figure 38: Example topology](#).

The BGP configuration for RR-5 is as follows:

```
# on RR-5:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      cluster {
        cluster-id 0.0.0.1
      }
      group "RR-clients" {
        peer-as 64496
        family {
          vpn-ipv4 true
        }
      }
      neighbor "192.0.2.1" {
        group "RR-clients"
      }
      neighbor "192.0.2.2" {
        group "RR-clients"
      }
      neighbor "192.0.2.3" {
        group "RR-clients"
      }
      neighbor "192.0.2.4" {
        group "RR-clients"
      }
    }
  }
}
```

The following BGP summary on RR-5 shows that BGP sessions with each PE are established for the VPN-IPv4 address family:

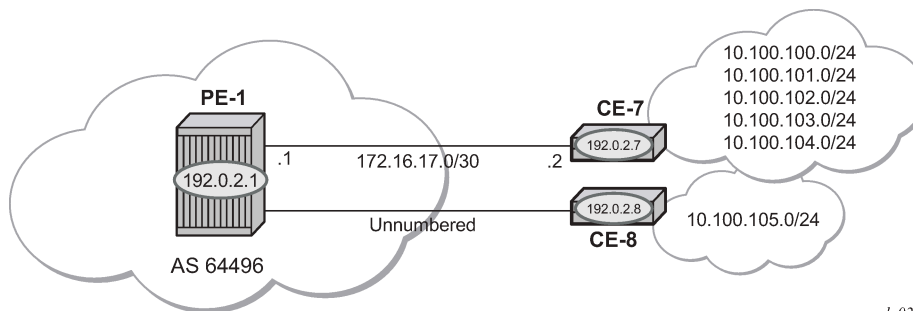
```
[ ]
A:admin@RR-5# show router bgp summary all
```

```
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId          AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
                   PktSent OutQ
-----
192.0.2.1
Def. Instance 64496      3   0 00h00m11s 0/0/0 (VpnIPv4)
                   3   0
192.0.2.2
Def. Instance 64496      3   0 00h00m11s 0/0/0 (VpnIPv4)
                   3   0
192.0.2.3
Def. Instance 64496      3   0 00h00m11s 0/0/0 (VpnIPv4)
                   3   0
192.0.2.4
Def. Instance 64496      3   0 00h00m11s 0/0/0 (VpnIPv4)
                   3   0
-----
```

VPRN: IPv4

Figure 39: CE connections for next-hops shows the Customer Edge (CE) routers connected to PE-1.

Figure 39: CE connections for next-hops



al_0291

The VPRN configuration for PE-1 is as follows:

```
# on PE-1:
configure {
  service {
    vprn "VPRN 1" {
      admin-state enable
      service-id 1
      customer "1"
      route-distinguisher "64496:1"
      vrf-target {
        community "target:64496:1"
      }
    }
    auto-bind-tunnel {
      resolution filter
      resolution-filter {
        ldp true
      }
    }
  }
}
```

```

    }
  }
  interface "int-PE-1-CE-7" {
    ipv4 {
      primary {
        address 172.16.17.1
        prefix-length 30
      }
    }
    sap 1/2/1:1.0 {
    }
  }
  interface "int-PE-1-CE-8" {
    ipv4 {
      unnumbered {
        ip-int-name "loop1"
      }
    }
    sap 1/2/2:1.0 {
    }
  }
  interface "loop1" {
    loopback true
    ipv4 {
      primary {
        address 192.0.2.100
        prefix-length 32
      }
    }
  }
}

```

For unnumbered interfaces, an IP address is borrowed from a loopback interface, see “Unnumbered Interfaces in RSVP-TE and LDP” in the MPLS volume of the *7450 ESS, 7750 SR, and 7950 XRS Advanced Configuration Guide — Part I*.

LDP is used as the label-switching protocol for next-hop resolution.

PE-4 is configured with an interface toward CE-6 that supports eBGP. The following export policy is configured:

```

# on PE-4:
configure {
  policy-options {
    community "1:1" {
      member "1:1" { }
    }
  }
  policy-statement "BGP-VPN-accept" {
    entry 10 {
      from {
        protocol {
          name [bgp-vpn]
        }
      }
      action {
        action-type accept
        community {
          add ["1:1"]
        }
      }
    }
  }
}

```

The configuration of the VPRN service "VPRN 1" on PE-4 is as follows:

```
# on PE-4:
configure {
  service {
    vprn "VPRN 1" {
      admin-state enable
      service-id 1
      customer "1"
      autonomous-system 64496
      route-distinguisher "64496:1"
      vrf-target {
        community "target:64496:1"
      }
      auto-bind-tunnel {
        resolution filter
        resolution-filter {
          ldp true
        }
      }
    }
    bgp {
      group "VPRN1-external" {
        peer-as 64497
        export {
          policy ["BGP-VPN-accept"]
        }
        import {
          policy ["1:1"]
        }
      }
      neighbor "172.16.46.2" {
        group "VPRN1-external"
      }
    }
  }
  interface "int-PE-4-CE-6" {
    ipv4 {
      primary {
        address 172.16.46.1
        prefix-length 30
      }
    }
    sap 1/2/1:1 {
    }
  }
}
```

Static routes with communities

A static route has multiple next-hop options: direct connected IP address, black-hole, indirect IP address, and interface-name.

[Figure 39: CE connections for next-hops](#) shows a pair of CE routers connected to PE-1. The link to CE-7 is a numbered link. The link to CE-8 is an unnumbered link. The loopback interface address is used as a reference address for the unnumbered Ethernet interface.

Beyond CE-7 are several /24 subnets. Static routes to these individual subnets are created on PE-1 using a static route with a next-hop type of "interface address" or an "indirect address". The indirect address is learned using a static route.

Beyond CE-8 is a single /24 subnet. A static route to this subnet is created with an interface-name as the next-hop.

There are several well-known, standard communities:

- **no-export**: the route is not advertised to any external peer. This route should be present in the route tables of all BGP speakers in the originating AS, but not in those in neighboring ASs.
- **no-advertise**: the route is not advertised to any peer. This route should not be present in any router as BGP-learned route.

The requirement for each subnet is:

- 10.100.100.0/24 must not be advertised outside of the AS. This must be associated with the standard, well-known community **no-export**. The community value is encoded as 65535:65281 (0xFFFFF01), but the CLI requires the keyword **no-export**.

```
# on PE-1:
configure {
  service {
    vprn "VPRN 1" {
      static-routes {
        route 10.100.100.0/24 route-type unicast {
          next-hop "172.16.17.2" {
            admin-state enable
            community "no-export"
          }
        }
      }
    }
  }
}
```

- 10.100.101.0/24 must be advertised with a community of 64496:101

```
route 10.100.101.0/24 route-type unicast {
  next-hop "172.16.17.2" {
    admin-state enable
    community "64496:101"
  }
}
```

- 10.100.102.0/24 must not be advertised to any BGP peer. This must be associated with the standard, well-known community **no-advertise**. The community value is encoded as 65535:65282 (0xFFFFF02), but the CLI requires the keyword **no-advertise**.

```
route 10.100.102.0/24 route-type unicast {
  next-hop "172.16.17.2" {
    admin-state enable
    community "no-advertise"
  }
}
```

- 10.100.103.0/24 must be advertised with a community of 64496:103 and a route tag of 10.

```
route 10.100.103.0/24 route-type unicast {
  next-hop "172.16.17.2" {
    admin-state enable
    tag 10
    community "64496:103"
  }
}
```

- 10.100.104.0/24 must be advertised with a community of 64496:104. It is reachable via 192.0.2.7 which, in turn, is reachable via 172.16.17.2. This is using a static route which does not need to be advertised, therefore, it is associated with the **no-advertise** community.

```

route 10.100.104.0/24 route-type unicast {
  indirect 192.0.2.7 {
    admin-state enable
    community "64496:104"
  }
}
route 192.0.2.7/32 route-type unicast {
  next-hop "172.16.17.2" {
    admin-state enable
    community "no-advertise"
  }
}

```

- 10.100.105.0/24 must be advertised with a community of 64496:105. It is reachable via the unnumbered interface to CE-8.

```

route 10.100.105.0/24 route-type unicast {
  interface "int-PE-1-CE-8" {
    admin-state enable
    community "64496:105"
  }
}

```

On PE-1, static routes are configured that match the static routes from [Figure 39: CE connections for next-hops](#), and the preceding conditions.

The default behavior of a VPRN is to export all static and connected routes into a BGP labeled route with the appropriate route-target extended community configured in the VRF-target statement. A single community string can be added using the preceding static-route community commands. If multiple communities are required, then a VRF-export policy should be used, but this is outside the scope of this chapter.

The following BGP table on PE-1 shows which VPN-IPv4 routes have been exported correctly to RR-5:

```

[]
A:admin@PE-1# show router bgp neighbor 192.0.2.5 advertised-routes vpn-ipv4
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                    Path-Id    IGP Cost
      As-Path                               Label
-----
i     64496:1:10.100.100.0/24                100        None
      192.0.2.1                             None        n/a
      No As-Path                             524283
i     64496:1:10.100.101.0/24                100        None
      192.0.2.1                             None        n/a
      No As-Path                             524283

```



```

i      64496:1:10.100.103.0/24      100      None
      192.0.2.1                    None      n/a
      No As-Path                    524283
i      64496:1:10.100.104.0/24      100      None
      192.0.2.1                    None      n/a
      No As-Path                    524283
i      64496:1:10.100.105.0/24      100      None
      192.0.2.1                    None      n/a
      No As-Path                    524283
i      64496:1:172.16.17.0/30       100      None
      192.0.2.1                    None      n/a
      No As-Path                    524283
i      64496:1:192.0.2.100/32       100      None
      192.0.2.1                    None      n/a
      No As-Path                    524283
-----
Routes : 7
=====

```

There are only seven exported routes. The route prefixes associated with the **no-advertise** community are not present, as expected.

Examining the BGP table of PE-4 shows the presence of the expected routes, with the correct community values.

The prefix 10.100.100.0/24 is a member of community **no-export**. This is correctly advertised to PE-4, as follows:

```

[]
A:admin@PE-4# show router bgp routes 10.100.100.0/24 vpn-ipv4 detail
=====
BGP Router ID:192.0.2.4      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Original Attributes

Network       : 10.100.100.0/24
Nextthop     : 192.0.2.1
Route Dist.  : 64496:1          VPN Label    : 524283
Path Id      : None
From        : 192.0.2.5
Res. Nextthop : n/a
Local Pref.  : 100              Interface Name : int-PE-4-PE-1
Aggregator AS : None           Aggregator    : None
Atomic Aggr. : Not Atomic      MED           : None
AIGP Metric  : None            IGP Cost      : 10
Connector    : None
Community    : no-export target:64496:1
Cluster      : 0.0.0.1
Originator Id : 192.0.2.1      Peer Router Id : 192.0.2.5
Fwd Class    : None           Priority       : None
Flags        : Used Valid Best IGP
Route Source : Internal
AS-Path      : No As-Path
Route Tag    : 0
Neighbor-AS  : n/a

```

```

Orig Validation: N/A
Source Class   : 0                      Dest Class   : 0
Add Paths Send : Default
Last Modified  : 01h16m07s
VPRN Imported  : 1
---snip---
    
```

The following command shows all members of the community **no-export**:

```

[]
A:admin@PE-4# show router bgp routes vpn-ipv4 community no-export
=====
BGP Router ID:192.0.2.4      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  64496:1:10.100.100.0/24                100        None
      192.0.2.1                             None        10
      No As-Path                             524283
-----
Routes : 1
=====
    
```

Because the community no-export is encoded as community 65535:65281, the same output can be retrieved as follows:

```

[]
A:admin@PE-4# show router bgp routes vpn-ipv4 community 65535:65281
=====
BGP Router ID:192.0.2.4      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  64496:1:10.100.100.0/24                100        None
      192.0.2.1                             None        10
      No As-Path                             524283
-----
Routes : 1
=====
    
```

The prefix 10.100.101.0/24 is a member of community 64496:101. This is correctly advertised to PE-4.

```
[ ]
A:admin@PE-4# show router bgp routes 10.100.101.0/24 vpn-ipv4 detail
=====
BGP Router ID:192.0.2.4      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Original Attributes

Network       : 10.100.101.0/24
NextHop       : 192.0.2.1
Route Dist.   : 64496:1          VPN Label     : 524283
Path Id       : None
From          : 192.0.2.5
Res. NextHop  : n/a
Local Pref.   : 100
Aggregator AS : None           Interface Name : int-PE-4-PE-1
Atomic Aggr.  : Not Atomic     Aggregator    : None
AIGP Metric   : None           MED           : None
Connector     : None           IGP Cost      : 10
Community     : 64496:101 target:64496:1
Cluster       : 0.0.0.1
Originator Id : 192.0.2.1       Peer Router Id : 192.0.2.5
Fwd Class     : None           Priority       : None
Flags         : Used Valid Best IGP
Route Source  : Internal
AS-Path       : No As-Path
Route Tag     : 0
Neighbor-AS   : n/a
Orig Validation: N/A
Source Class  : 0              Dest Class     : 0
Add Paths Send : Default
Last Modified : 01h34m23s
VPRN Imported : 1
---snip---
```

The prefix 10.100.103.0/24 is a member of community 64496:103. This is correctly advertised to PE-4, as follows:

```
[ ]
A:admin@PE-4# show router bgp routes 10.100.103.0/24 vpn-ipv4 detail
=====
BGP Router ID:192.0.2.4      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Original Attributes

Network       : 10.100.103.0/24
```

```

Nexthop      : 192.0.2.1
Route Dist.  : 64496:1          VPN Label    : 524283
Path Id      : None
From         : 192.0.2.5
Res. Nexthop : n/a
Local Pref.  : 100              Interface Name : int-PE-4-PE-1
Aggregator AS : None           Aggregator    : None
Atomic Aggr. : Not Atomic      MED           : None
AIGP Metric  : None           IGP Cost      : 10
Connector    : None
Community   : 64496:103 target:64496:1
Cluster      : 0.0.0.1
Originator Id : 192.0.2.1      Peer Router Id : 192.0.2.5
Fwd Class    : None           Priority       : None
Flags        : Used Valid Best IGP
Route Source : Internal
AS-Path      : No As-Path
Route Tag    : 0
Neighbor-AS  : n/a
Orig Validation: N/A
Source Class : 0              Dest Class    : 0
Add Paths Send : Default
Last Modified : 01h26m24s
VPRN Imported : 1
---snip---

```

The prefix 10.100.104.0/24 is a member of community 64496:104. This is correctly advertised to PE-4, as follows:

```

[]
A:admin@PE-4# show router bgp routes 10.100.104.0/24 vpn-ipv4 detail
=====
BGP Router ID:192.0.2.4      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Original Attributes

Network      : 10.100.104.0/24
Nexthop      : 192.0.2.1
Route Dist.  : 64496:1          VPN Label    : 524283
Path Id      : None
From         : 192.0.2.5
Res. Nexthop : n/a
Local Pref.  : 100              Interface Name : int-PE-4-PE-1
Aggregator AS : None           Aggregator    : None
Atomic Aggr. : Not Atomic      MED           : None
AIGP Metric  : None           IGP Cost      : 10
Connector    : None
Community   : 64496:104 target:64496:1
Cluster      : 0.0.0.1
Originator Id : 192.0.2.1      Peer Router Id : 192.0.2.5
Fwd Class    : None           Priority       : None
Flags        : Used Valid Best IGP
Route Source : Internal
AS-Path      : No As-Path
Route Tag    : 0

```

```
Neighbor-AS : n/a
Orig Validation: N/A
Source Class : 0                      Dest Class : 0
Add Paths Send : Default
Last Modified : 01h20m45s
VPRN Imported : 1
---snip---
```

The prefix 10.100.105.0/24 is a member of community 64496:105. This is correctly advertised to PE-4.

```
[ ]
A:admin@PE-4# show router bgp routes 10.100.105.0/24 vpn-ipv4 detail
=====
BGP Router ID:192.0.2.4      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Original Attributes

Network       : 10.100.105.0/24
NextHop      : 192.0.2.1
Route Dist.  : 64496:1          VPN Label    : 524283
Path Id      : None
From        : 192.0.2.5
Res. NextHop : n/a
Local Pref. : 100              Interface Name : int-PE-4-PE-1
Aggregator AS : None          Aggregator   : None
Atomic Aggr. : Not Atomic     MED          : None
AIGP Metric  : None           IGP Cost     : 10
Connector    : None
Community  : 64496:105 target:64496:1
Cluster      : 0.0.0.1
Originator Id : 192.0.2.1      Peer Router Id : 192.0.2.5
Fwd Class    : None           Priority      : None
Flags        : Used Valid Best IGP
Route Source : Internal
AS-Path      : No As-Path
Route Tag    : 0
Neighbor-AS  : n/a
Orig Validation: N/A
Source Class : 0                      Dest Class : 0
Add Paths Send : Default
Last Modified : 01h18m11s
VPRN Imported : 1
---snip---
```

The following route table of VPRN 1 on PE-4 shows that these seven BGP-learned routes are present as valid routes.

```
[ ]
A:admin@PE-4# show router 1 route-table protocol bgp-vpn
=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]                Type   Proto   Age      Pref
```

```

Next Hop[Interface Name]                               Metric
-----
10.100.100.0/24                                         Remote BGP VPN 01h54m30s 170
    192.0.2.1 (tunneled)                                0
10.100.101.0/24                                         Remote BGP VPN 01h46m55s 170
    192.0.2.1 (tunneled)                                0
10.100.103.0/24                                         Remote BGP VPN 01h37m47s 170
    192.0.2.1 (tunneled)                                0
10.100.104.0/24                                         Remote BGP VPN 01h30m18s 170
    192.0.2.1 (tunneled)                                0
10.100.105.0/24                                         Remote BGP VPN 01h26m58s 170
    192.0.2.1 (tunneled)                                0
172.16.17.0/30                                          Remote BGP VPN 01h54m30s 170
    192.0.2.1 (tunneled)                                0
192.0.2.100/32                                         Remote BGP VPN 01h54m30s 170
    192.0.2.1 (tunneled)                                0
-----
No. of Routes: 7
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

The following route table on CE-6 shows six valid BGP-learned routes, as expected:

```

[]
A:admin@CE-6# show router route-table protocol bgp

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                                     Type  Proto  Age           Pref
  Next Hop[Interface Name]                             Metric
-----
10.100.101.0/24                                         Remote BGP    00h04m31s 170
    172.16.46.1                                         0
10.100.103.0/24                                         Remote BGP    00h04m31s 170
    172.16.46.1                                         0
10.100.104.0/24                                         Remote BGP    00h04m31s 170
    172.16.46.1                                         0
10.100.105.0/24                                         Remote BGP    00h04m31s 170
    172.16.46.1                                         0
172.16.17.0/30                                          Remote BGP    00h04m31s 170
    172.16.46.1                                         0
192.0.2.100/32                                         Remote BGP    00h04m31s 170
    172.16.46.1                                         0
-----
No. of Routes: 6
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

The prefix 10.100.100.0/24 is not received from PE-4 because it is a member of the **no-export** community.

```

[]
A:admin@CE-6# show router bgp routes 10.100.100.0/24 detail

=====
BGP Router ID:192.0.2.6      AS:64497      Local AS:64497
=====

```

```
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
```

```
=====
BGP IPv4 Routes
=====
```

```
No Matching Entries Found
=====
```

Static route 10.100.101.0/24 is received on CE-6 with the correct community 64496:101, as follows:

```
[ ]
A:admin@CE-6# show router bgp routes community 64496:101
=====
BGP Router ID:192.0.2.6      AS:64497      Local AS:64497
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
u*>i  10.100.101.0/24          None       None
      172.16.46.1          None       0
      64496                 -         -
-----
Routes : 1
=====
```

Static route 10.100.103.0/24 is received on CE-6 with the correct community 64496:103, as follows:

```
[ ]
A:admin@CE-6# show router bgp routes community 64496:103
=====
BGP Router ID:192.0.2.6      AS:64497      Local AS:64497
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
u*>i  10.100.103.0/24          None       None
      172.16.46.1          None       0
      64496                 -         -
-----
Routes : 1
=====
```

Static route 10.100.104.0/24 is received on CE-6 with the correct community 64496:104, as follows:

```
[ ]
A:admin@CE-6# show router bgp routes community 64496:104
=====
BGP Router ID:192.0.2.6      AS:64497      Local AS:64497
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
u*>i 10.100.104.0/24          None       None
      172.16.46.1           None       0
      64496                  -         -
-----
Routes : 1
=====
```

Static route 10.100.105.0/24 is received on CE-6 with the correct community 64496:105.

```
[ ]
A:admin@CE-6# show router bgp routes community 64496:105
=====
BGP Router ID:192.0.2.6      AS:64497      Local AS:64497
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
u*>i 10.100.105.0/24          None       None
      172.16.46.1           None       0
      64496                  -         -
-----
Routes : 1
=====
```

Aggregate routes with communities

An aggregate route can be configured to represent a larger number of prefixes. For example, a set of prefixes 10.101.0.0/24 to 10.101.7.0/24 can be represented as a single aggregate prefix of 10.101.0.0/21.

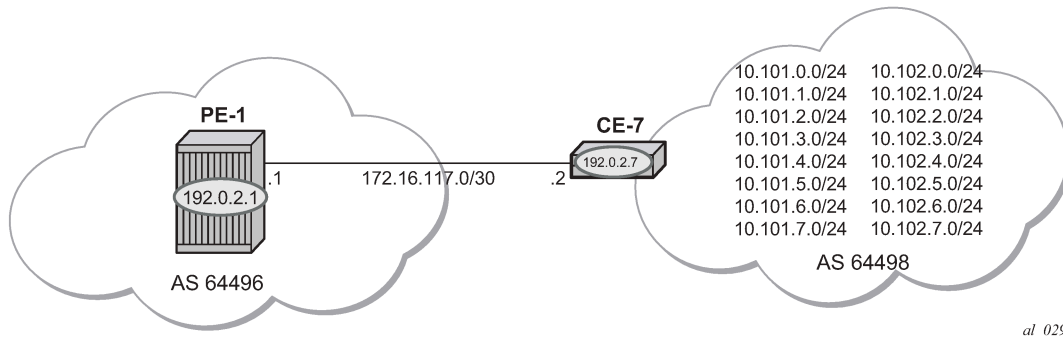
This is because the third octet in the range 0 to 7 can be represented by the 8 bits 00000000 to 00000111. The first 5 bits of this octet are common, along with the previous 2 octets, giving a prefix where the first 21 bits are common. Therefore, the aggregate can be written as 10.101.0.0/21.

To illustrate the configuration of an aggregate, consider following.

Figure 40: CE-7 connectivity shows a CE router (CE-7), in AS 64498, that advertises a series of contiguous prefixes via BGP.

- 10.101.0.0/24 to 10.101.7.0/24
- 10.102.0.0/24 to 10.102.7.0/24

Figure 40: CE-7 connectivity



Instead of advertising all these prefixes out of the VPRN toward an external CE individually, an aggregate route can be configured that summarizes each set of eight prefixes and a community can be directly associated with each aggregate route.

The configuration for VPRN service "VPRN 2" on PE-1, including the external BGP configuration is as follows:

```
# on PE-1:
configure {
  policy-options {
    community "1:2" {
      member "1:2" { }
    }
  }
  policy-statement "1:2" {
    entry 10 {
      from {
        community {
          name "1:2"
        }
      }
      action {
        action-type accept
      }
    }
  }
}
service {
  vprn "VPRN 2" {
    admin-state enable
    service-id 2
    customer "1"
    autonomous-system 64496
    route-distinguisher "64496:2"
    vrf-target {
      community "target:64496:2"
    }
    auto-bind-tunnel {
```

```

        resolution filter
        resolution-filter {
            ldp true
        }
    }
    bgp {
        group "external" {
            peer-as 64498
            import {
                policy ["1:2"]
            }
            export {
                policy ["1:2"]
            }
        }
        neighbor "172.16.117.2" {
            group "external"
        }
    }
    interface "int-PE-1-CE-7_2nd" {
        ipv4 {
            primary {
                address 172.16.117.1
                prefix-length 30
            }
        }
        sap 1/2/1:2.0 {
        }
    }
}

```

The BGP neighbor relationship on PE-1 shows the following:

```

[]
A:admin@PE-1# show router 2 bgp neighbor
=====
BGP Neighbor
=====
-----
Peer          : 172.16.117.2
Description   : (Not Specified)
Group         : external
-----
Peer AS       : 64498           Peer Port      : 179
Peer Address  : 172.16.117.2
Local AS      : 64496           Local Port     : 50195
Local Address : 172.16.117.1
Peer Type     : External       Dynamic Peer   : No
State        : Established     Last State     : Active
Last Event   : recvOpen
Last Error   : Unrecognized Error
Local Family : IPv4
Remote Family: IPv4
Hold Time    : 90              Keep Alive     : 30
Min Hold Time : 0
Active Hold Time : 90         Active Keep Alive : 30
Cluster Id   : None
Preference   : 170
Input Queue  : 0
Input Messages : 7           Output Queue   : 0
Input Octets : 247          Output Messages : 7
Input Updates : 1           Output Octets  : 232
Input RtRefresh : 0         Output Updates : 1
Output RtRefresh : 0         Output RtRefresh : 0

```

```
TTL Security      : Disabled      Min TTL Value      : n/a
Graceful Restart  : Disabled      Stale Routes Time  : n/a
Restart Time     : n/a
Long-Lived GR    : Disabled
Advertise Inactive : Disabled      Peer Tracking      : Disabled
Auth key chain   : n/a
Disable Cap Nego : Disabled      Bfd Enabled        : Disabled
Default Route Tgt : Disabled
Aigp Metric      : Disabled      Split Horizon      : Disabled
Damp Peer Oscillatio*: Disabled    Update Errors      : 0
GR Notification  : Disabled      Fault Tolerance    : Disabled
Rem Idle Hold Time : 00h00m00s
Next-Hop Unchanged : None
sel-lbl-ipv4-install : Disabled
Local Capability  : RtRefresh MPBGP 4byte ASN
Remote Capability : RtRefresh MPBGP 4byte ASN
Routes Resolve To St*: Disabled
Local AddPath Capabi*: Disabled
Remote AddPath Capab*: Send - None
                  : Receive - None
Import Policy     : 1:2
                  : Default Reject
Export Policy     : 1:2
                  : Default Reject
---snip---
-----
Neighbors shown : 1
=====
* indicates that the corresponding row element may have been truncated.
```

The following output shows the 16 received BGP routes on PE-1:

```
[ ]
A:admin@PE-1# show router 2 bgp routes
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path
-----
u*>i  10.101.0.0/24             None       None
      172.16.117.2          None       0
      64498                  -
u*>i  10.101.1.0/24             None       None
      172.16.117.2          None       0
      64498                  -
u*>i  10.101.2.0/24             None       None
      172.16.117.2          None       0
      64498                  -
u*>i  10.101.3.0/24             None       None
      172.16.117.2          None       0
      64498                  -
u*>i  10.101.4.0/24             None       None
      172.16.117.2          None       0
```

```

64498
u*>i 10.101.5.0/24      None      None
      172.16.117.2      None      0
      64498              -
u*>i 10.101.6.0/24      None      None
      172.16.117.2      None      0
      64498              -
u*>i 10.101.7.0/24      None      None
      172.16.117.2      None      0
      64498              -
u*>i 10.102.0.0/24      None      None
      172.16.117.2      None      0
      64498              -
u*>i 10.102.1.0/24      None      None
      172.16.117.2      None      0
      64498              -
u*>i 10.102.2.0/24      None      None
      172.16.117.2      None      0
      64498              -
u*>i 10.102.3.0/24      None      None
      172.16.117.2      None      0
      64498              -
u*>i 10.102.4.0/24      None      None
      172.16.117.2      None      0
      64498              -
u*>i 10.102.5.0/24      None      None
      172.16.117.2      None      0
      64498              -
u*>i 10.102.6.0/24      None      None
      172.16.117.2      None      0
      64498              -
u*>i 10.102.7.0/24      None      None
      172.16.117.2      None      0
      64498              -
-----
Routes : 16
=====

```

PE-4 also has a VPRN 2 instance configured, so that it will receive the imported BGP routes. The service configuration for "VPRN 2" on PE-4 is as follows:

```

# on PE-4:
configure {
  service {
    vprn "VPRN 2" {
      admin-state enable
      service-id 2
      customer "1"
      autonomous-system 64496
      route-distinguisher "64496:2"
      vrf-target {
        community "target:64496:2"
      }
      auto-bind-tunnel {
        resolution filter
        resolution-filter {
          ldp true
        }
      }
    }
    bgp {
      group "VPRN2-external" {
        peer-as 64497
        import {

```

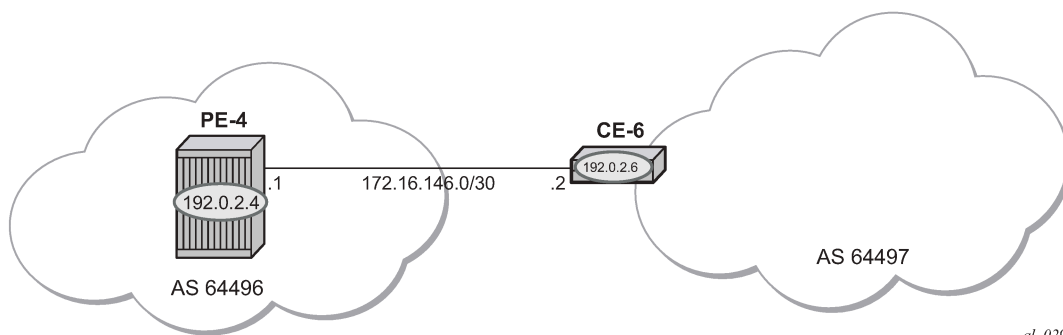
```

        policy ["1:2"]
        }
        export {
            policy ["1:2"]
        }
    }
    neighbor "172.16.146.2" {
        group "VPRN2-external"
    }
}
interface "int-PE-4-CE-6_2nd" {
    ipv4 {
        primary {
            address 172.16.146.1
            prefix-length 30
        }
    }
    sap 1/2/1:2 {
    }
}
}

```

Figure 41: CE-6 connectivity shows the connectivity between PE-4 and CE-6. PE-4 will only forward a summarizing aggregate route toward CE-6.

Figure 41: CE-6 connectivity



al_0292

PE-4 receives labeled BGP route prefixes from PE-1 via the route reflector and installs them in the FIB for router instance 2, as follows:

```

[]
A:admin@PE-4# show router 2 route-table
=====
Route Table (Service: 2)
=====
Dest Prefix[Flags]                                Type  Proto  Age           Pref
  Next Hop[Interface Name]                        Metric
-----
10.101.0.0/24                                     Remote BGP VPN 00h01m07s 170
  192.0.2.1 (tunneled)                            0
10.101.1.0/24                                     Remote BGP VPN 00h01m07s 170
  192.0.2.1 (tunneled)                            0
10.101.2.0/24                                     Remote BGP VPN 00h01m07s 170
  192.0.2.1 (tunneled)                            0
10.101.3.0/24                                     Remote BGP VPN 00h01m07s 170
  192.0.2.1 (tunneled)                            0
10.101.4.0/24                                     Remote BGP VPN 00h01m07s 170
  192.0.2.1 (tunneled)                            0

```

```

10.101.5.0/24 Remote BGP VPN 00h01m07s 170
    192.0.2.1 (tunneled) 0
10.101.6.0/24 Remote BGP VPN 00h01m07s 170
    192.0.2.1 (tunneled) 0
10.101.7.0/24 Remote BGP VPN 00h01m07s 170
    192.0.2.1 (tunneled) 0
10.102.0.0/24 Remote BGP VPN 00h01m07s 170
    192.0.2.1 (tunneled) 0
10.102.1.0/24 Remote BGP VPN 00h01m07s 170
    192.0.2.1 (tunneled) 0
10.102.2.0/24 Remote BGP VPN 00h01m07s 170
    192.0.2.1 (tunneled) 0
10.102.3.0/24 Remote BGP VPN 00h01m07s 170
    192.0.2.1 (tunneled) 0
10.102.4.0/24 Remote BGP VPN 00h01m07s 170
    192.0.2.1 (tunneled) 0
10.102.5.0/24 Remote BGP VPN 00h01m07s 170
    192.0.2.1 (tunneled) 0
10.102.6.0/24 Remote BGP VPN 00h01m07s 170
    192.0.2.1 (tunneled) 0
10.102.7.0/24 Remote BGP VPN 00h01m07s 170
    192.0.2.1 (tunneled) 0
172.16.117.0/30 Remote BGP VPN 00h02m41s 170
    192.0.2.1 (tunneled) 0
172.16.146.0/30 Local Local 00h02m42s 0
    int-PE-4-CE-6_2nd 0
-----
No. of Routes: 18
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

On CE-6, an additional interface is configured toward PE-4, as follows:

```

# on CE-6:
configure {
  service {
    ies "IES 2" {
      admin-state enable
      service-id 2
      customer "1"
      interface "int-CE-6-PE-4_2nd" {
        sap 1/1/1:2 {
        }
        ipv4 {
          primary {
            address 172.16.146.2
            prefix-length 30
          }
        }
      }
    }
  }
}

```

The BGP configuration of CE-6 is as follows:

```

# on CE-6:
configure {
  router "Base" {
    bgp {
      group "external-toVPRN2onPE-4" {
        peer-as 64496
        import {

```

```

        policy ["1:2"]
        }
        export {
            policy ["1:2"]
        }
    }
    neighbor "172.16.146.1" {
        group "external-toVPRN2onPE-4"
    }
}

```

The BGP neighbor state for PE-4 is as follows:

```

[]
A:admin@PE-4# show router 2 bgp neighbor 172.16.146.2
=====
BGP Neighbor
=====
-----
Peer           : 172.16.146.2
Description    : (Not Specified)
Group         : VPRN2-external
-----
Peer AS       : 64497           Peer Port      : 179
Peer Address  : 172.16.146.2
Local AS     : 64496           Local Port     : 50683
Local Address : 172.16.146.1
Peer Type    : External       Dynamic Peer   : No
State        : Established    Last State     : Active
Last Event   : recvOpen
Last Error   : Unrecognized Error
Local Family : IPv4
Remote Family: IPv4
Hold Time    : 90             Keep Alive     : 30
Min Hold Time : 0
Active Hold Time : 90       Active Keep Alive : 30
Cluster Id   : None
Preference   : 170           Num of Update Flaps : 0
Input Queue  : 0             Output Queue    : 0
Input Messages : 25         Output Messages  : 20
Input Octets  : 750          Output Octets   : 387
Input Updates : 5           Output Updates  : 0
Input RtRefresh : 0         Output RtRefresh : 0
TTL Security : Disabled     Min TTL Value   : n/a
Graceful Restart : Disabled  Stale Routes Time : n/a
Restart Time : n/a
Long-Lived GR : Disabled
Advertise Inactive : Disabled Peer Tracking   : Disabled
Auth key chain : n/a
Disable Cap Nego : Disabled Bfd Enabled     : Disabled
Default Route Tgt : Disabled
Aigp Metric      : Disabled Split Horizon    : Disabled
Damp Peer Oscillatio*: Disabled Update Errors    : 0
GR Notification  : Disabled Fault Tolerance  : Disabled
Rem Idle Hold Time : 00h00m00s
Next-Hop Unchanged : None
sel-lbl-ipv4-install : Disabled
Local Capability  : RtRefresh MPBGP 4byte ASN
Remote Capability : RtRefresh MPBGP 4byte ASN
Routes Resolve To St*: Disabled
Local AddPath Capabi*: Disabled
Remote AddPath Capab*: Send - None

```

```

: Receive - None
Import Policy      : 1:2
                  : Default Reject
Export Policy      : 1:2
                  : Default Reject
---snip---

-----
Neighbors shown : 1
=====
* indicates that the corresponding row element may have been truncated.

```

To advertise a summarizing aggregate route with an associated community string, an aggregate route is required. In this case, the 10.101.x.0/24 group of prefixes will be associated with community 64496:101. The 10.102.x.0/24 group of prefixes will be associated with the standard community **no-export**, so that it will not be advertised to any external peer. These aggregate routes are configured in VPRN 2 on PE-4, as follows:

```

# on PE-4:
configure {
  service {
    vprn "VPRN 2" {
      aggregates {
        aggregate 10.101.0.0/21 {
          community ["64496:101"]
        }
        aggregate 10.102.0.0/21 {
          community ["no-export"]
        }
      }
    }
  }
}

```

The following export policy is required on PE-4 to allow the advertising of the aggregate route. No community is applied using this policy.

```

# on PE-4:
configure {
  policy-options {
    policy-statement "PE-4-VPN-Agg" {
      entry 10 {
        from {
          protocol {
            name [aggregate]
          }
        }
        action {
          action-type accept
          community {
            add ["1:2"]
          }
        }
      }
    }
  }
}

```

This is applied as an export policy within the group context of the BGP configuration of the VPRN, as follows:

```

# on PE-4:
configure {
  service {

```



```
vprn "VPRN 2" {
  bgp {
    group "VPRN2-external" {
      export {
        policy ["PE-4-VPN-Agg"]
      }
    }
  }
}
```

The aggregate route 10.101.0.0/21 is received at CE-6 via BGP. The community that was associated with this prefix is seen: 64496:101. The route is seen as an aggregate, with PE-4 as the aggregating router (192.0.2.4). The "Atomic Aggregate" attribute is present, meaning that PE-4 has not advertised any details of the AS Paths of the composite routes.

```
[]
A:admin@CE-6# show router bgp routes 10.101.0.0/21 hunt
=====
BGP Router ID:192.0.2.6      AS:64497      Local AS:64497
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
-----
RIB In Entries
-----
Network       : 10.101.0.0/21
Nextthop      : 172.16.146.1
Path Id       : None
From          : 172.16.146.1
Res. Protocol : LOCAL                      Res. Metric   : 0
Res. Nextthop : 172.16.146.1
Local Pref.   : None                      Interface Name : int-CE-6-PE-4_2nd
Aggregator AS : 64496                      Aggregator    : 192.0.2.4
Atomic Aggr. : Atomic                   MED           : None
AIGP Metric   : None                      IGP Cost      : 0
Connector     : None
Community     : 64496:101
Cluster       : No Cluster Members
Originator Id : None                      Peer Router Id : 192.0.2.4
Fwd Class     : None                      Priority       : None
Flags         : Used Valid Best IGP
Route Source  : External
AS-Path       : 64496
Route Tag     : 0
Neighbor-AS   : 64496
Orig Validation: NotFound
Source Class  : 0                      Dest Class    : 0
Add Paths Send : Default
Last Modified : 00h02m07s
---snip---
```

The aggregate route 10.102.0.0/21 is not received at CE-6, because PE-4 does not advertise it, due to the fact that it is associated with the "no-export" community.

```
[]
A:admin@CE-6# show router bgp routes 10.102.0.0/21 hunt
=====
```

```
BGP Router ID:192.0.2.6      AS:64497      Local AS:64497
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
No Matching Entries Found
=====
```

Conclusion

Community strings can be added to static and aggregate routes. This example shows the configuration of communities with both static and aggregate routes, together with the associated show outputs which can be used to verify and troubleshoot them.

BGP Add-Path

This chapter provides information about BGP Add-Path.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The chapter was initially written for SR OS Release 14.0.R7, but the MD-CLI in the current edition is based on SR OS Release 22.2.R2.

Overview

When a BGP router learns multiple paths for the same prefix, it selects one route as its best path and advertises only this route to its BGP peers. The BGP add-path feature allows advertising the best n paths for the same prefix, where n is configurable. If the set of n paths includes multiple paths with the same BGP next hop, only the best route with a specific next hop is advertised and the other paths are suppressed.

The BGP add-path feature increases path visibility in the Autonomous System (AS), because more routes are stored in the Routing Information Base (RIB). BGP add-path has the following benefits:

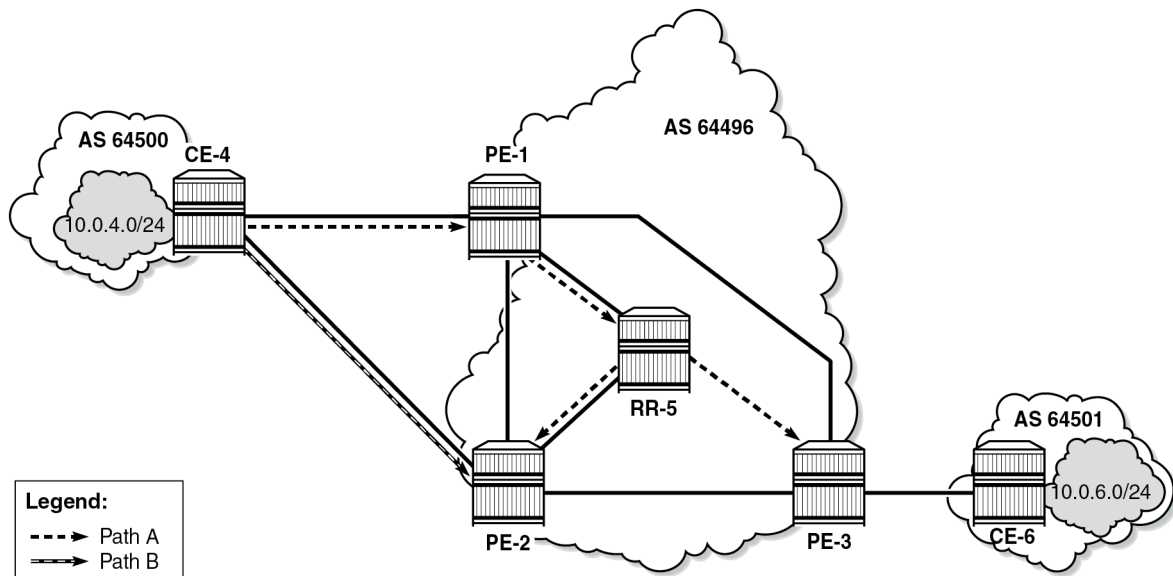
- Faster convergence after failure
- Enhanced load-sharing
- Reduced routing churn

These benefits are described in the following sections.

Faster convergence after failure

[Figure 42: RR advertises best path only – path A preferred over path B](#) shows a network that does not support add-path. CE-4 advertises two paths for prefix 10.0.4.0/24 to its EBGP neighbors: PE-1 and PE-2. PE-1 has an import policy that sets the local preference (LP) of path A to 200; PE-2 has an import policy that keeps the default LP of 100 for path B. Therefore, path A that is advertised to PE-1 is preferred in AS 64496. The route reflector RR-5 advertises the preferred path A to PE-2 and PE-3. PE-2 suppresses the advertisement of its external path (B) to RR-5, because path A is preferred. Traffic from CE-6 to CE-4 is sent via PE-3 and PE-1.

Figure 42: RR advertises best path only – path A preferred over path B



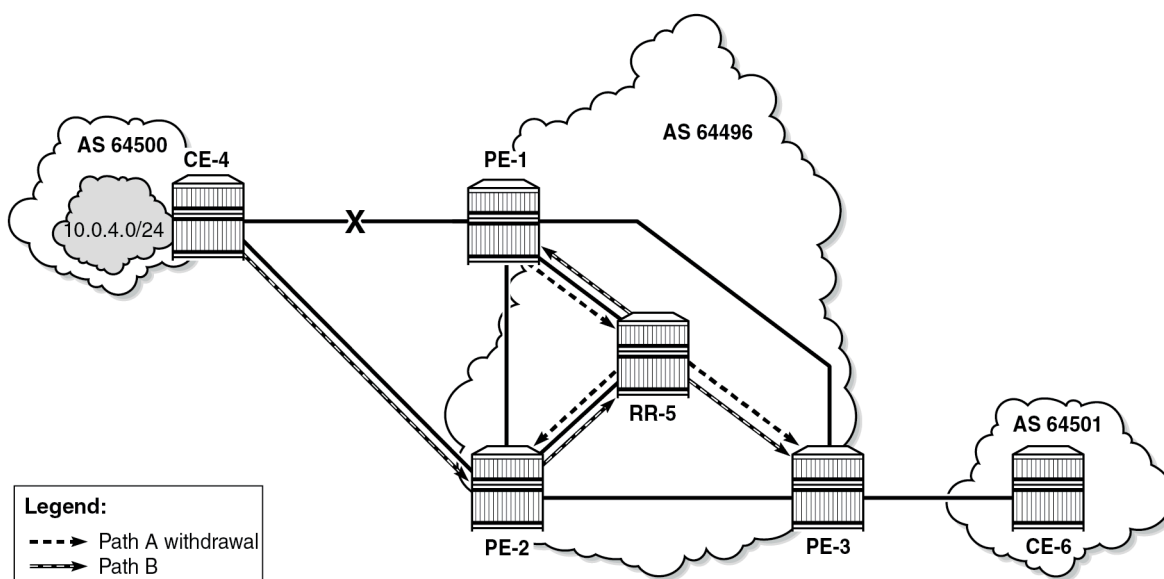
26362

When the link between CE-4 and PE-1 fails, the following steps take place for reconvergence:

1. PE-1 sends a BGP update withdrawing path A to RR-5.
2. RR-5 receives and propagates the withdrawal to its other clients: PE-2 and PE-3.
3. PE-2 receives the withdrawal of path A and reruns the BGP decision process. PE-2 selects path B as its best route and advertises path B to RR-5.
4. RR-5 receives the BGP update for path B and reruns its BGP decision process. RR-5 selects path B as its best path and advertises path B to its other clients: PE-1 and PE-3.
5. PE-1 and PE-3 rerun their BGP decision process and determine that path B is the best path. Traffic can flow from CE-6 to CE-4 via PE-3 and PE-2.

Figure 43: Reconvergence after path failure (without add-path) shows the BGP updates sent to withdraw path A and advertise path B.

Figure 43: Reconvergence after path failure (without add-path)



26363

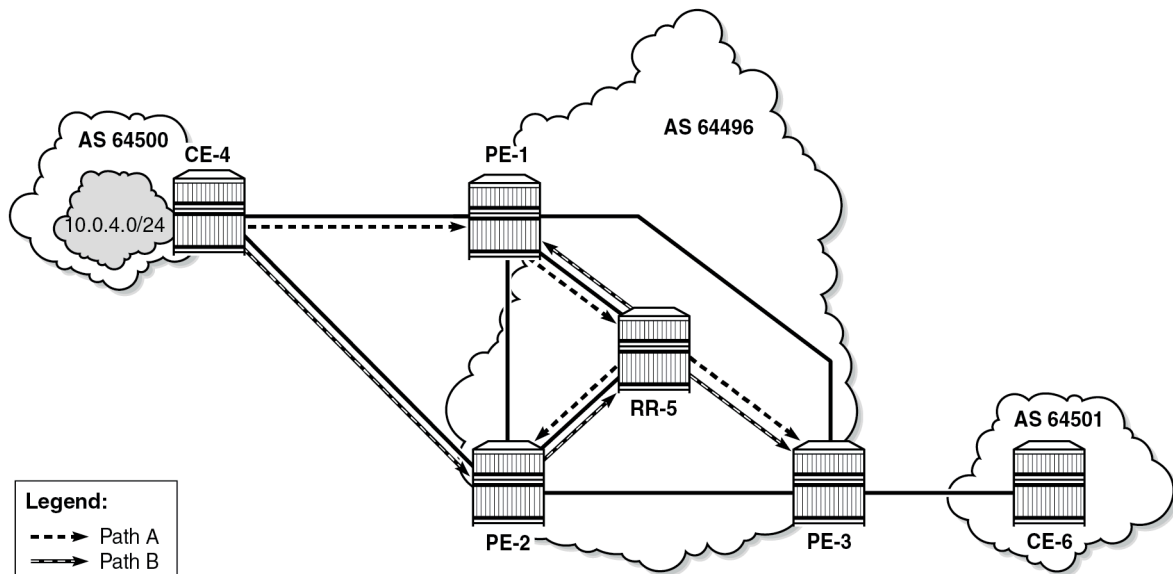
If the propagation time of a BGP update message between RR-5 and any of its clients is X, the convergence time is four times X, plus processing, transmission, and queuing delays.

With the use of add-path on all BGP routers in AS 64496, the convergence time can be reduced considerably, because PE-3 has more than one path for prefix 10.0.4.0/24 in its RIB-IN before the failure takes place. When there are no failures, PE-2 decides that path A is best, and PE-2 also advertises its second-best path (B)—which is its best external path—to RR-5. With add-path enabled, the RR has knowledge of two paths for prefix 10.0.4.0/24 and advertises both to its clients. PE-3 receives two routes for prefix 10.0.4.0/24, reruns the BGP decision process, and updates its forwarding table based on the results. The following options are possible:

- Path A is the best path, whereas path B is maintained in the RIB-IN. The FIB entry for destination 10.0.4.0/24 points at path {A} only.
- When BGP FRR is enabled as described in chapter BGP Fast Reroute, path A is the best path and path B is the second-best path. The FIB entry for destination 10.0.4.0/24 points to path {A,B}. If path A is available, it is used for all traffic to the destination; if path A is unavailable but path B is available, then all traffic to the destination is directed to path B. In this case, path B is effectively a pre-computed, pre-installed backup path for the destination.
- When Equal Cost Multi-Path (ECMP) and BGP multipath are enabled and the paths have an equal cost, both paths A and B represent the best path. The FIB entry for destination 10.0.4.0/24 points to multipath entry {A,B}. When both paths are available, traffic to the destination is load-shared across paths A and B. If only one path is available, traffic is directed to that available path.

Figure 44: Advertised paths when BGP add-path is enabled in PEs and RR shows the BGP update messages prior to any failures. RR-5 receives path A from PE-1 and path B from PE-2, whereas it advertises path B to PE-1, path A to PE-2, and both path A and path B to PE-3. Path B has the default LP 100, whereas path A gets LP 200 as per import policy on PE-1. However, in case of ECMP, both paths keep the default LP 100.

Figure 44: Advertised paths when BGP add-path is enabled in PEs and RR

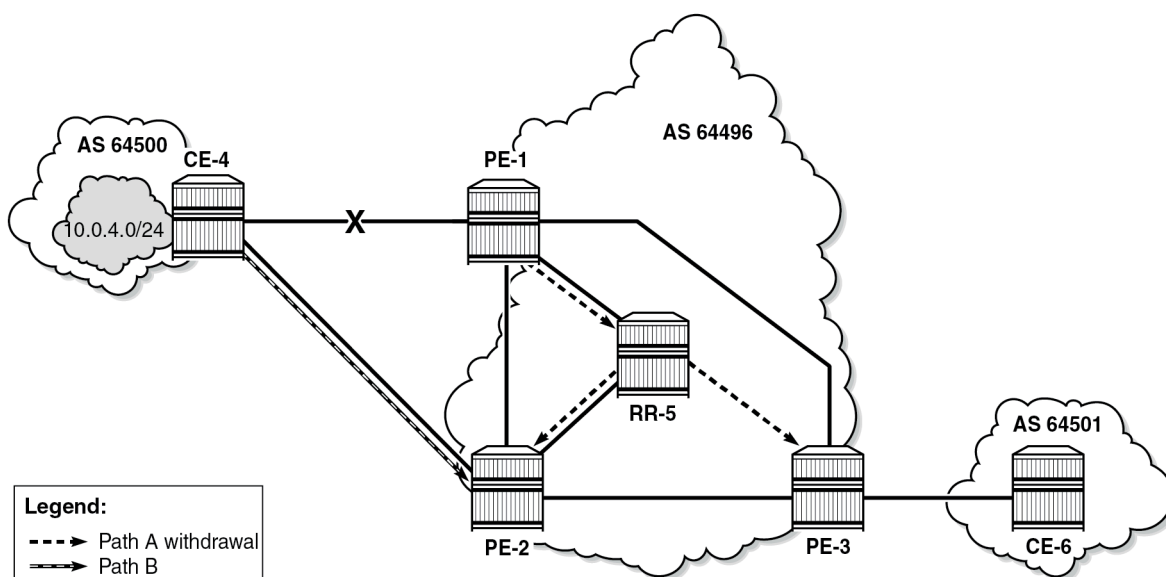


26364

Figure 45: [Reconvergence after path failure when BGP add-path is enabled](#) shows the BGP update messages that are sent after a link failure between CE-4 and PE-1. With add-path, fewer steps are required for convergence:

1. PE-1 sends a BGP update message withdrawing path A.
2. RR-5 receives the withdrawal and propagates it to its clients PE-2 and PE-3.
3. PE-2 and PE-3 receive the withdrawal, rerun the BGP decision process, and update the forwarding entry for destination 10.0.4.0/24: path B is best.

Figure 45: Reconvergence after path failure when BGP add-path is enabled



26365

The convergence time with add-path is much shorter than without add-path. If X is the propagation time of a BGP update message between RR and any of the PEs, then the convergence time is the time required for the BGP update from PE-1 to RR-5 (X) plus the time required for the BGP update propagation from RR-5 to the other PEs (X), in addition to delays for processing, transmission, and queuing. The convergence with add-path is twice as fast as without add-path.

For some types of failures, the convergence can be even faster:

- When PE-1 becomes unreachable, the next-hop tracking by PE-3 will invalidate path A before the BGP withdrawal message is received from RR-5.
- If PE-3 implements BGP FRR and path A has been marked as unusable, PE-3 can switch traffic destined to 10.0.4.0/24 to path B.
- When Bidirectional Forwarding Detection (BFD) is enabled on the EBGP sessions and on the IGP protocol, the failure is detected faster and BGP convergence can be sped up when BGP FRR is enabled.

Enhanced load-sharing

When paths A and B are equal in cost or preference, and ECMP and BGP multipath are enabled on all PEs, load-sharing can be done for traffic with destination 10.0.4.0/24. With BGP add-path, both paths A and B are advertised to the PEs. PE-3 runs the BGP decision process and determines that paths A and B are both best paths to destination 10.0.4.0/24, so paths A and B are combined into one multipath forwarding entry: {A,B}.

The benefits of load-sharing for traffic to destination 10.0.4.0/24 are the following:

- More even bandwidth utilization of the links in AS 64496
- More even bandwidth utilization for traffic across peering points PE-1 and PE-2 with AS 64500

- Faster reaction to some failures; for example, the BGP next hop for one of the paths becomes unreachable in the IGP and next hop tracking is enabled.

Reduced routing churn

Routing churn refers to repeated advertisements and withdrawals of a prefix and path. Some degree of routing churn is normal and expected in most networks. However, it should be contained as much as possible to avoid overloading router CPUs. Routing churn can be caused by:

- Flapping links (links that repeatedly transition between up and down state)
- Route oscillation (networks that use RRs or AS confederations and BGP path selection relies on Multi Exit Discriminator (MED) and IGP cost comparisons)

Add-path helps to reduce routing churn by constraining the effect of some failures to the local AS where they occur. For example, the link between CE-4 and PE-1 could repeatedly cycle up and down due to a misconfiguration. When the link goes down, a BGP withdrawal message is sent by PE-1 to RR-5 and from RR-5 to the other RR clients (PE-2 and PE-3). PE-3 will withdraw and advertise path A to its EBGP peer CE-6 in AS 64501, but path B is constantly advertised to CE-6 (when add-path has been negotiated between PE-3 and CE-6).

Without add-path, PE-2 would be affected by the instability in AS 64496 and there would be periods of time when AS 64501 has no paths to destination 10.0.4.0/24 (between the withdrawal of path A and the advertisement of path B).

Add-path implementation

BGP **add-path** is configured in the base routing instance, for IBGP or EBGP, per address family at different levels: in the global **bgp** context, per **group**, and per **neighbor**. The following address families are supported:

```
[ex:/configure router "Base" bgp]
A:admin@PE-1# add-paths ?

add-paths

evpn          + Enter the evpn context
ipv4          + Enter the ipv4 context
ipv6          + Enter the ipv6 context
label-ipv4    + Enter the label-ipv4 context
label-ipv6    + Enter the label-ipv6 context
mcast-vpn-ipv4 + Enter the mcast-vpn-ipv4 context
mcast-vpn-ipv6 + Enter the mcast-vpn-ipv6 context
mvpn-ipv4     + Enter the mvpn-ipv4 context
mvpn-ipv6     + Enter the mvpn-ipv6 context
vpn-ipv4      + Enter the vpn-ipv4 context
vpn-ipv6      + Enter the vpn-ipv6 context
```

Up to 16 paths are configurable per address family per peer (send-limit):

```
*[ex:/configure router "Base" bgp add-paths ipv4]
A:admin@PE-1# ?

receive      - Receive multiple routes per unlabeled IPv4 prefix
send         - Max paths advertised per unlabeled IPv4 unicast prefix
```


Only the number of advertised routes per prefix is controlled, not the number of received routes. All routes advertised by an add-path peer are accepted; otherwise, routing loops might occur. If a BGP speaker is configured with **send n**, but has more than n paths available in the LOC-RIB, it selects the n best paths with unique BGP next hops following the Add-n path selection algorithm described in *draft-ietf-idr-add-paths-guidelines*. Also, the send limit n can be overridden, for specific prefixes, using route policies.

When BGP add-path is configured for an address family, the BGP capability will be announced to the BGP peer as part of the BGP open message, as follows:

```
# Enable debugging for BGP open messages on PE-1 (in classic CLI):
//debug router bgp open
```

```
23 2022/05/06 15:58:35.200 CEST MINOR: DEBUG #2001 Base BGP
"BGP: OPEN
Peer 1: 172.16.14.2 - Send (Active) BGP OPEN: Version 4
AS Num 64496: Holdtime 90: BGP_ID 192.0.2.1: Opt Length 26 (ExtOpt F)
Opt Para: Type CAPABILITY: Length = 24: Data:
  Cap_Code GRACEFUL-RESTART: Length 2
  Bytes: 0x0 0x78
  Cap_Code MP-BGP: Length 4
  Bytes: 0x0 0x1 0x0 0x1
  Cap_Code ROUTE-REFRESH: Length 0
  Cap_Code 4-OCTET-ASN: Length 4
  Bytes: 0x0 0x0 0xfb 0xf0
  Cap_Code ADD-PATH: Length 4
  Bytes: 0x0 0x1 0x1 0x3
"
```

The BGP add-path capability code value typically consists of one or more blocks of four bytes; two octets for the Address Family Identifier (AFI), one octet for the Subsequent Address Family Identifier (SAFI), and one octet for send/receive. In this example, AFI/SAFI bytes point to an IPv4 address family and send/receive value "3" means that the sender is able to receive and send multiple paths from/to its BGP peer.

In BGP update messages, a 4-octet path identifier (ID) is added to the Network Layer Reachability Information (NLRI) field. The combination of both prefix and path ID identifies a BGP path. SR OS allocates path IDs sequentially on a per address family basis, not per prefix. The path ID is only locally significant, which means that when a BGP speaker re-advertises a route with path IDs, it must generate its own path ID.

```
# Enable debugging for BGP UPDATE messages on RR-5 (in classic CLI):
//debug router bgp update
```

RR-5 received the following BGP update for prefix 10.0.4.0/24 with path ID.

```
44 2022/05/06 15:59:37.463 CEST MINOR: DEBUG #2001 Base Peer 1: 192.0.2.2
"Peer 1: 192.0.2.2: UPDATE
Peer 1: 192.0.2.2 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 27
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 6 AS Path:
    Type: 2 Len: 1 < 64500 >
  Flag: 0x40 Type: 3 Len: 4 Nexthop: 192.0.2.2
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
  NLRI: Length = 8
    10.0.4.0/24 Path-ID 1
"
```

When routers have negotiated to advertise (and receive) routes with path identifiers, all BGP updates (advertisements or withdrawals) without path identifier will be rejected. There will be an NLRI parsing error—because the BGP update has an incorrect length—and a notification will be sent.

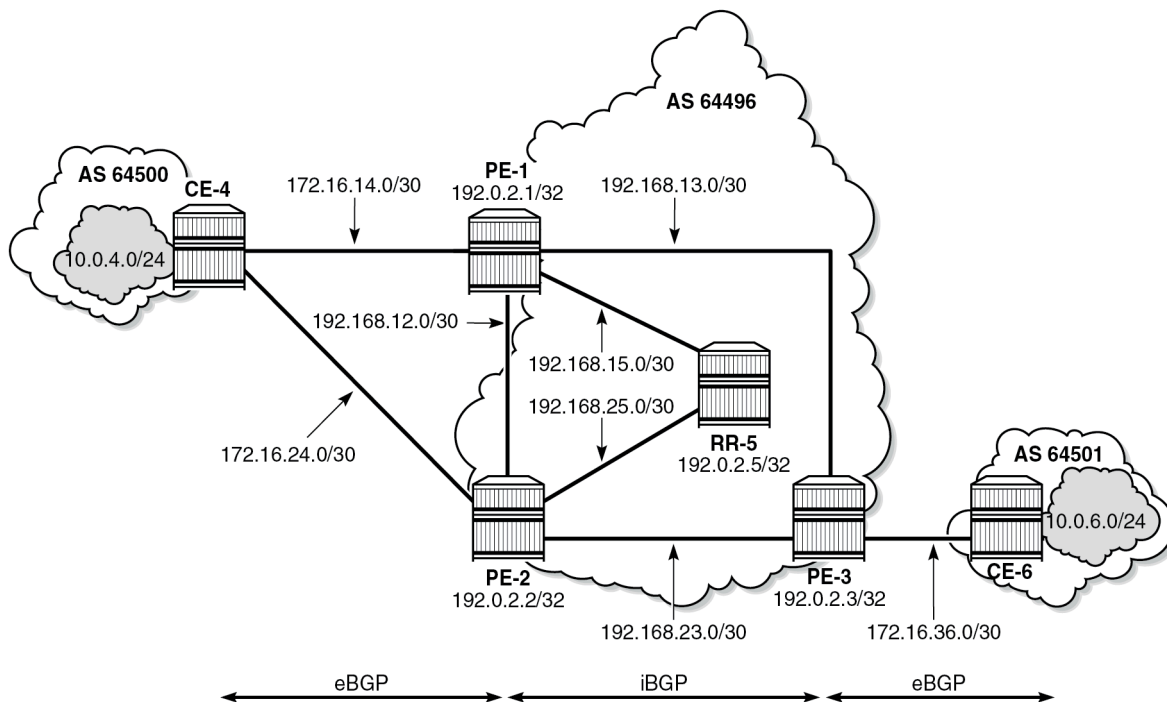
Configuration

The following configuration examples are in this section:

- BGP without add-path
- BGP with add-path for address family IPv4: no BGP FRR, no ECMP
- BGP with add-path for address family IPv4 and BGP FRR enabled
- BGP with add-path for address family IPv4 and ECMP enabled
- BGP with add-path for address family VPN-IPv4 and BGP FRR enabled
- BGP with add-path for address family VPN-IPv4 and ECMP enabled

Figure 46: Example topology shows the example topology with CE-4 in AS 64500 advertising route 10.0.4.0/24 to its EBGP peers PE-1 and PE-2 in AS 64496. PE-1 has an import policy that sets the LP for this route to 200, whereas PE-2 has an import policy that keeps the default local preference of 100. RR-5 is RR for all PEs in AS 64496. CE-6 in AS 64501 peers with PE-3 in AS 64496 and can send traffic to CE-4 in AS 64500.

Figure 46: Example topology



26366

Initial configuration

The initial configuration on all nodes includes:

- Cards, MDAs, ports
- Router interfaces
- IS-IS as IGP on all interfaces within AS 64496 (alternatively, OSPF can be used)
- LDP on all interfaces between the PEs in AS 64496, but not toward RR-5

BGP is configured on all the nodes. CE-4 peers with PE-1 and PE-2 and exports prefix 10.0.4.0/24 to both EBGP peers, as follows:

```
# on CE-4:
configure {
  policy-options {
    prefix-list "10.0.0.0/8" {
      prefix 10.0.0.0/8 type longer {
      }
    }
    prefix-list "10.0.4.0/24" {
      prefix 10.0.4.0/24 type exact {
      }
    }
  }
  policy-statement "import-bgp-10.x" {
    entry 10 {
      from {
        prefix-list ["10.0.0.0/8"]
      }
      action {
        action-type accept
      }
    }
  }
  policy-statement "export-bgp-10.0.4.x" {
    entry 10 {
      from {
        prefix-list ["10.0.4.0/24"]
      }
      action {
        action-type accept
      }
    }
  }
}
router "Base" {
  autonomous-system 64500
  bgp {
    rapid-withdrawal true
    split-horizon true
    group "EBGP" {
      peer-as 64496
      import {
        policy ["import-bgp-10.x"]
      }
      export {
        policy ["export-bgp-10.0.4.x"]
      }
    }
  }
  neighbor "172.16.14.1" {
    group "EBGP"
  }
  neighbor "172.16.24.1" {
```

```

    group "EBGP"
  }

```

The BGP configuration on CE-6 is similar.

PE-1 peers with CE-4 in AS 64500 and RR-5 in AS 64496. An import policy is configured to set the LP to 200 for all routes received from CE-4, as follows:

```

# on PE-1:
configure {
  policy-options {
    prefix-list "10.0.0.0/8" {
      prefix 10.0.0.0/8 type longer {
      }
    }
    policy-statement "export-bgp-10.x" {
      entry 10 {
        from {
          prefix-list ["10.0.0.0/8"]
        }
        action {
          action-type accept
        }
      }
    }
    policy-statement "import-bgp-LP200" {
      entry 10 {
        from {
          prefix-list ["10.0.0.0/8"]
        }
        action {
          action-type accept
          local-preference 200
        }
      }
    }
  }
}
router "Base" {
  autonomous-system 64496
  bgp {
    rapid-withdrawal true
    split-horizon true
    group "EBGP" {
      peer-as 64500
      import {
        policy ["import-bgp-LP200"]
      }
      export {
        policy ["export-bgp-10.x"]
      }
    }
    group "IBGP" {
      next-hop-self true
      peer-as 64496
    }
    neighbor "172.16.14.2" {
      group "EBGP"
    }
    neighbor "192.0.2.5" {
      group "IBGP"
    }
  }
}

```

The BGP configuration on PE-2 and PE-3 is similar, but with an import policy that does not set the LP, so the default LP of 100 applies.

The BGP configuration on RR-5 is as follows:

```
# on RR-5:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      rapid-withdrawal true
      split-horizon true
      group "IBGP" {
        peer-as 64496
        cluster {
          cluster-id 192.0.2.5
        }
      }
      neighbor "192.0.2.1" {
        group "IBGP"
      }
      neighbor "192.0.2.2" {
        group "IBGP"
      }
      neighbor "192.0.2.3" {
        group "IBGP"
      }
    }
  }
}
```

PE-1 advertises a route for prefix 10.0.4.0/24 with LP 200 to RR-5. RR-5 propagates this route to its other clients: PE-2 and PE-3. When PE-2 learns this route, it does not advertise its own route for 10.0.4.0/24 with LP 100 to RR-5 anymore. PE-3 only learns the route for prefix 10.0.4.0/24 with LP 200, as follows:

```
[/]
A:admin@PE-3# show router bgp routes 10.0.4.0/24
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i 10.0.4.0/24                             200        None
      192.0.2.1                             None       10
      64500                                  -
-----
Routes : 1
=====
```

Reconvergence without add-path

A failure of the link between CE-4 and PE-1 is simulated as follows:

```
# on CE-4:
configure {
  router "Base" {
    interface "int-CE-4-PE-1" {
      admin-state disable
    }
  }
  commit
}
```

The following four BGP update messages are received or sent by RR-5.

RR-5 receives the following withdrawal message from PE-1:

```
# on RR-5:
15 2022/05/06 15:53:26.491 CEST MINOR: DEBUG #2001 Base Peer 1: 192.0.2.1
"Peer 1: 192.0.2.1: UPDATE
Peer 1: 192.0.2.1 - Received BGP UPDATE:
  Withdrawn Length = 4
  10.0.4.0/24
  Total Path Attr Length = 0
"
```

RR-5 propagates this withdrawal to its other clients, for example to PE-2, as follows:

```
# on RR-5:
16 2022/05/06 15:53:26.491 CEST MINOR: DEBUG #2001 Base Peer 1: 192.0.2.2
"Peer 1: 192.0.2.2: UPDATE
Peer 1: 192.0.2.2 - Send BGP UPDATE:
  Withdrawn Length = 4
  10.0.4.0/24
  Total Path Attr Length = 0
"
```

When PE-2 receives this withdrawal, it reruns the BGP decision process and decides that its route for prefix 10.0.4.0/24 with LP 100 is the best route. PE-2 advertises this route to RR-5; it is received by RR-5 as follows:

```
# on RR-5:
18 2022/05/06 15:53:49.540 CEST MINOR: DEBUG #2001 Base Peer 1: 192.0.2.2
"Peer 1: 192.0.2.2: UPDATE
Peer 1: 192.0.2.2 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 27
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 6 AS Path:
    Type: 2 Len: 1 < 64500 >
  Flag: 0x40 Type: 3 Len: 4 Nexthop: 192.0.2.2
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
  NLRI: Length = 4
  10.0.4.0/24
"
```

RR-5 propagates this message to its other clients: PE-1 and PE-3. The following BGP update is sent to PE-3:

```
# on RR-5:
20 2022/05/06 15:53:56.479 CEST MINOR: DEBUG #2001 Base Peer 1: 192.0.2.3
"Peer 1: 192.0.2.3: UPDATE
```

```
Peer 1: 192.0.2.3 - Send BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 41
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 6 AS Path:
    Type: 2 Len: 1 < 64500 >
  Flag: 0x40 Type: 3 Len: 4 Nexthop: 192.0.2.2
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
  Flag: 0x80 Type: 9 Len: 4 Originator ID: 192.0.2.2
  Flag: 0x80 Type: 10 Len: 4 Cluster ID:
    192.0.2.5
  NLRI: Length = 4
    10.0.4.0/24
"
```

Again, PE-3 has only one route for prefix 10.0.4.0/24, but this time with next hop 192.0.2.2, as follows:

```
[/]
A:admin@PE-3# show router bgp routes 10.0.4.0/24
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                  l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
      Nexthop (Router)                     Path-Id   IGP Cost
      As-Path                               Label
-----
u*>i 10.0.4.0/24
      192.0.2.2
      64500
-----
Routes : 1
=====
```

The configuration is restored as follows:

```
# on CE-4:
configure {
  router "Base" {
    interface "int-CE-4-PE-1" {
      admin-state enable
    }
  }
  commit
}
```

Add-path enabled: no BGP FRR, no ECMP

Before add-path is enabled, the following information is displayed on PE-1 for BGP neighbor RR-5:

```
[/]
A:admin@PE-1# show router bgp neighbor 192.0.2.5 | match "Local AddPath" post-lines 2
Local AddPath Capabi*: Disabled
Remote AddPath Capab*: Send - None
                      : Receive - None
```

Add-path is enabled on PE-1 and PE-2 with a send path limit of two for groups "EBGP" and "IBGP"; the **receive true** option implies that the add-path receive capability is negotiated, so IPv4 routes without path ID will be rejected:

```
# on PE-1, PE-2:
configure {
  router "Base" {
    bgp {
      group "EBGP" {
        add-paths {
          ipv4 {
            send 2
            receive true
          }
        }
      }
      group "IBGP" {
        add-paths {
          ipv4 {
            send 2
            receive true
          }
        }
      }
    }
  }
}
```

When the preceding **show** command is repeated on PE-1 or PE-2, the local BGP add-path capabilities are specified for address family IPv4: a maximum of two paths can be sent for a specific IPv4 prefix. The remote peer RR-5 does not have add-path enabled yet.

```
[/]
A:admin@PE-1# show router bgp neighbor 192.0.2.5 | match "Local AddPath" post-lines 3
Local AddPath Capabi*: Send - ipv4 (2)
                       : Receive - ipv4
Remote AddPath Capab*: Send - None
                       : Receive - None
```

Initially, add-path remains disabled on PE-3. On the RR, add-path is enabled for neighbors 192.0.2.1 and 192.0.2.2, but not for 192.0.2.3 yet. For neighbor 192.0.2.1, the **receive false** option implies that the add-path receive capability is not negotiated.

```
# on RR-5:
configure {
  router "Base" {
    bgp {
      neighbor 192.0.2.1 {
        add-paths {
          ipv4 {
            send 2
            receive false
          }
        }
      }
      neighbor 192.0.2.2 {
        add-paths {
          ipv4 {
            send 2
            receive true
          }
        }
      }
    }
  }
}
```


The following output shows that add-path is enabled locally on RR-5 and remotely on PE-1 for address family IPv4. RR-5 can send a maximum of two paths for a specific prefix toward PE-1 and PE-2; toward PE-3, add-path remains disabled.

```
[/]
A:admin@RR-5# show router bgp neighbor 192.0.2.1 | match "Local AddPath" post-lines 3
Local AddPath Capabi*: Send - ipv4 (2)
                        : Receive - None
Remote AddPath Capab*: Send - ipv4
                        : Receive - ipv4
```

```
[/]
A:admin@RR-5# show router bgp neighbor 192.0.2.2 | match "Local AddPath" post-lines 3
Local AddPath Capabi*: Send - ipv4 (2)
                        : Receive - ipv4
Remote AddPath Capab*: Send - ipv4
                        : Receive - ipv4
```

```
[/]
A:admin@RR-5# show router bgp neighbor 192.0.2.3 | match "Local AddPath" post-lines 2
Local AddPath Capabi*: Disabled
Remote AddPath Capab*: Send - None
                        : Receive - None
```

The **receive false** option indicates that RR-5 does not negotiate the add-path receive capability with its peer. PE-1 knows that peer 192.0.2.5 may send IPv4 routes with a path ID, but has no information about what this peer will receive:

```
[/]
A:admin@PE-1# show router bgp neighbor 192.0.2.5 | match "Local AddPath" post-lines 3
Local AddPath Capabi*: Send - ipv4 (2)
                        : Receive - ipv4
Remote AddPath Capab*: Send - ipv4
                        : Receive - None
```

With BGP add-path enabled, PE-2 will advertise its second-best route for prefix 10.0.4.0/24 with LP 100 to RR-5. PE-1, PE-2, and RR-5 will have two routes for prefix 10.0.4.0/24 in their RIB-IN, but only the route with LP 200 will be used. The following output shows the BGP routes on RR-5, but it resembles the output on PE-1 and PE-2:

```
[/]
A:admin@RR-5# show router bgp routes 10.0.4.0/24
=====
BGP Router ID:192.0.2.5      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)        Path-Id    IGP Cost
      As-Path                  Label
-----
u*>i  10.0.4.0/24                200        None
```

```

192.0.2.1          None      10
64500             -
*i 10.0.4.0/24    100      None
192.0.2.2        1         10
64500             -
-----
Routes : 2
=====

```

Even though RR-5 has two routes for this prefix, it only advertises its best route to PE-3, because `add-path` is not enabled for this BGP session. Therefore, PE-3 only has the route for 10.0.4.0/24 with LP 200, as follows:

```

[/]
A:admin@PE-3# show router bgp routes 10.0.4.0/24
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i 10.0.4.0/24           200        None
      192.0.2.1           None        10
      64500                -
-----
Routes : 1
=====

```

When `add-path` is enabled on the session between PE-3 and RR-5, the second route will also be advertised, as follows:

```

# on PE-3:
configure {
  router "Base" {
    bgp {
      group "IBGP" {
        add-paths {
          ipv4 {
            send 2
            receive true
          }
        }
      }
    }
  }
}

```

```

# on RR-5:
configure {
  router "Base" {
    bgp {
      neighbor 192.0.2.3 {
        add-paths {
          ipv4 {
            send 2

```

```

    receive true
  }
}

[/]
A:admin@PE-3# show router bgp routes 10.0.4.0/24
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  10.0.4.0/24                200        None
      192.0.2.1                13         10
      64500                      -
*i    10.0.4.0/24                100        None
      192.0.2.2                14         10
      64500                      -
-----
Routes : 2
=====

```

BGP add-path is enabled, but BGP FRR and ECMP are disabled. The routing table on PE-3 only contains one entry for prefix 10.0.4.0/24:

```

[/]
A:admin@PE-3# show router route-table 10.0.4.0/24
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age          Pref
  Next Hop[Interface Name]                Metric
-----
10.0.4.0/24                        Remote BGP    00h06m44s  170
  192.168.13.1                        10
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====

```

Reconverge with add-path: no BGP FRR, no ECMP

A link failure between CE-4 and PE-1 is simulated as follows:

```
# on CE-4:
```

```
configure {
  router "Base" {
    interface "int-CE-4-PE-1" {
      admin-state disable
    }
    commit
  }
}
```

PE-1 sends a withdrawal message for route 10.0.4.0/24 with LP 200 to RR-5 and reruns the BGP decision process. RR-5 propagates this withdrawal message to its other clients that rerun the BGP decision process. As a result, the route for prefix 10.0.4.0/24 with LP 100 will be used on all nodes; for example, on PE-3:

```
[/]
A:admin@PE-3# show router bgp routes 10.0.4.0/24
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
u*>i  10.0.4.0/24              100        None
      192.0.2.2             14         10
      64500                  -          -
-----
Routes : 1
=====
```

The routing table contains a route to 10.0.4.0/24 with PE-2 as next hop, as follows:

```
[/]
A:admin@PE-3# show router route-table 10.0.4.0/24
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age           Pref
Next Hop[Interface Name]   Type  Proto  Metric
-----
10.0.4.0/24                 Remote BGP    00h04m07s  170
192.168.23.1                10
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```

The convergence with add-path enabled is twice as fast as without BGP add-path. With BGP add-path disabled, four sequential messages are sent:

1. PE-1 sends a withdrawal to RR-5.
2. RR-5 propagates withdrawal.

3. PE-2 advertises its route.
4. RR-5 propagates the route.

In the scenario with add-path, the last two messages are already sent before the failure happened. During convergence, only two withdrawal messages are sent: PE-1 sends a withdrawal to RR-5; RR-5 propagates this to its clients.

Add-path and BGP FRR

The convergence time can be further reduced by enabling BGP FRR, where the BGP decision process runs for the best route and the backup path before any failure happens, as described in chapter [BGP Fast Reroute](#). On all PEs, BGP FRR is enabled for the IPv4 address family, as follows:

```
# on all PEs:
configure {
  router "Base" {
    bgp {
      backup-path {
        ipv4 true
      }
    }
  }
}
```

Each PE has two routes for prefix 10.0.4.0/24 and when BGP FRR is enabled, both are used, but one is used as backup, indicated by the "b"-flag in the following output:

```
[/]
A:admin@PE-3# show router bgp routes 10.0.4.0/24
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  10.0.4.0/24                               200        None
      192.0.2.1                               20         10
      64500                                     -
ub*i  10.0.4.0/24                               100        None
      192.0.2.2                               14         10
      64500                                     -
-----
Routes : 2
=====
```

The following routing table on PE-3 shows the active route for 10.0.4.0/24 and adds an indication "B", indicating that a BGP backup route is available:

```
[/]
A:admin@PE-3# show router route-table 10.0.4.0/24
=====
```

```
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
Next Hop[Interface Name]          Metric
-----
10.0.4.0/24 [B]                   Remote BGP    00h01m48s 170
192.168.13.1                       10
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

The following output shows both the active and the backup route for prefix 10.0.4.0/24:

```
[/]
A:admin@PE-3# show router route-table 10.0.4.0/24 alternative

Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
Next Hop[Interface Name]          Metric
Alt-NextHop                       Alt-
Metric
-----
10.0.4.0/24                       Remote BGP    00h01m48s 170
192.168.13.1                       10
10.0.4.0/24 (Backup)              Remote BGP    00h01m48s 170
192.168.23.1                       10
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
      Backup = BGP backup route
      LFA = Loop-Free Alternate nexthop
      S = Sticky ECMP requested
=====
```

In case of link failure between CE-4 and PE-1, the same BGP withdrawals will be sent from PE-1 to RR-5 and from RR-5 to PE-2 and PE-3. When PE-2 and PE-3 receive the withdrawal, the BGP decision process need not run again. The backup path is promoted to active immediately.

BGP FRR is disabled on the PEs:

```
# on all PEs"
configure {
  router "Base" {
    bgp {
      delete backup-path
    }
  }
}
```

Add-path and ECMP

To have paths with equal cost, the import policy "import-bgp-10.x" on PE-1 does not set the LP:

```
# on PE-1:
configure {
  policy-options {
```

```

prefix-list "10.0.0.0/8" {
  prefix 10.0.0.0/8 type longer {
  }
}
policy-statement "import-bgp-10.x" {
  entry 10 {
    from {
      prefix-list ["10.0.0.0/8"]
    }
    action {
      action-type accept
    }
  }
}
}
router "Base" {
  bgp {
    group "EBGP" {
      delete import
      import {
        policy ["import-bgp-10.x"]
      }
    }
  }
}

```

On all PEs, ECMP is enabled with a value of 2 and BGP multipath is configured with the maximum number of paths equal to 2 in the **bgp** context, as follows:

```

# on all PEs:
configure {
  router "Base" {
    ecmp 2
    bgp {
      multipath {
        max-paths 2
      }
    }
  }
}

```

For more information about BGP multipath, see chapter [BGP Multipath](#).

All PEs have two routes for prefix 10.0.4.0/24 and both are active when ECMP is enabled; for example, for PE-3, as follows:

```

[/]
A:admin@PE-3# show router bgp routes 10.0.4.0/24
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  10.0.4.0/24                             100        None
      192.0.2.1                             20         10
      64500                                  -
u*>i  10.0.4.0/24                             100        None
      192.0.2.2                             14         10

```

```

64500          -
-----
Routes : 2
=====

[/]
A:admin@PE-3# show router route-table 10.0.4.0/24

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]                Metric
-----
10.0.4.0/24                        Remote BGP     00h02m33s  170
      192.168.13.1                      10
10.0.4.0/24                        Remote BGP     00h02m33s  170
      192.168.23.1                      10
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====

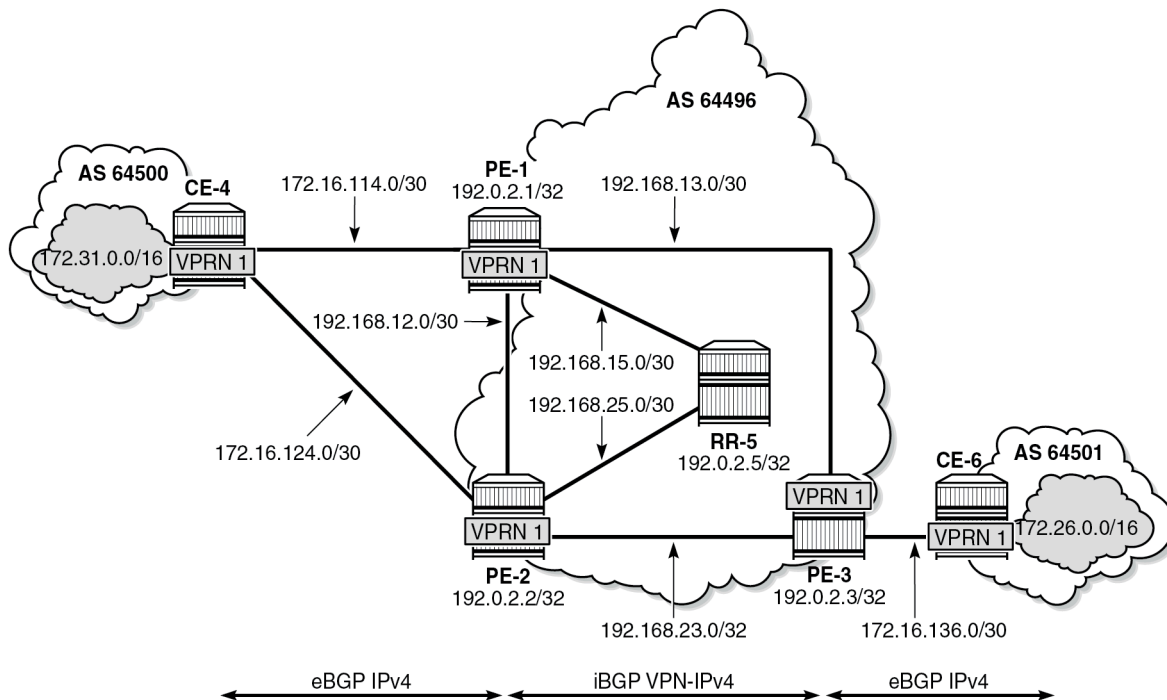
```

Traffic flows with destination 10.0.4.0/24 will be sprayed over the two active paths.

Add-path for family VPN-IPv4 with BGP FRR

[Figure 47: Example topology with VPRNs](#) shows the example topology with VPRN1 configured on the PEs in AS 64496. CE-4 exports prefix 172.31.0.0/16 to VPRN 1 on PE-1 and PE-2.

Figure 47: Example topology with VPRNs



26367

VPRN 1 is configured on all PEs in AS 64496, but not on the RR. BGP FRR is enabled in the VPRN with the **bgp-vpn-backup>ipv4 true** option. The configuration of VPRN 1 is similar on all PEs; for example, for PE-1, the VPRN configuration is as follows:

```
# on PE-1:
configure {
  policy-options {
    prefix-list "172.16.0.0/12" {
      prefix 172.16.0.0/12 type longer {
      }
    }
  }
  policy-statement "import-bgp-vprn-LP200" {
    entry 10 {
      from {
        prefix-list ["172.16.0.0/12"]
      }
      action {
        action-type accept
        local-preference 200
      }
    }
  }
  policy-statement "export-bgp-vprn" {
    entry 10 {
      from {
        protocol {
          name [bgp-vpn]
        }
      }
      to {

```


The **export-inactive-bgp true** option must be configured on PE-2, because the route for prefix 172.31.0.0/16 received by PE-2 from CE-4 is inactive, but should still be advertised as BGP VPN-IPv4 route to RR-5; see chapter "BGP Best External in a VPRN" in the Layer 3 Services volume of *7450 ESS, 7750 SR, and 7950 XRS Advanced Configuration Guide — Book II*. In this example, the **export-inactive-bgp true** option is configured on all PEs.

On the CEs, the configuration is either in the base routing instance—with additional router interfaces and BGP neighbors—or in a VPRN. In this example, the following VPRN is configured on CE-4:

```
# on CE-4:
configure {
  policy-options {
    prefix-list "172.16.0.0/12" {
      prefix 172.16.0.0/12 type longer {
      }
    }
    prefix-list "172.31.0.0/16" {
      prefix 172.31.0.0/16 type longer {
      }
    }
  }
  policy-statement "import-bgp-vprn" {
    entry 10 {
      from {
        prefix-list ["172.16.0.0/12"]
      }
      action {
        action-type accept
      }
    }
  }
  policy-statement "export_172.31.0.0/16" {
    entry 10 {
      from {
        prefix-list ["172.31.0.0/16"]
      }
      action {
        action-type accept
      }
    }
  }
}
service {
  vprn "VPRN 1" {
    admin-state enable
    service-id 1
    customer "1"
    autonomous-system 64500
    bgp {
      split-horizon true
      group "EBGP_1" {
        peer-as 64496
        import {
          policy ["import-bgp-vprn"]
        }
        export {
          policy ["export_172.31.0.0/16"]
        }
      }
      neighbor "172.16.114.1" {
        group "EBGP_1"
      }
      neighbor "172.16.124.1" {
        group "EBGP_1"
      }
    }
  }
}
```

```

    }
  }
  interface "int-CE-4-PE-1_VPRN1" {
    ipv4 {
      primary {
        address 172.16.114.2
        prefix-length 30
      }
    }
    sap 1/1/1:1 {
    }
  }
  interface "int-CE-4-PE-2_VPRN1" {
    ipv4 {
      primary {
        address 172.16.124.2
        prefix-length 30
      }
    }
    sap 1/1/2:1 {
    }
  }
  interface "test_connectedNW" {
    loopback true
    ipv4 {
      primary {
        address 172.31.0.1
        prefix-length 16
      }
    }
  }
}
}
}

```

The configuration on CE-6 is similar.

For all BGP speakers in AS 64496, BGP must be configured for address family VPN-IPv4 as well as for IPv4. BGP add-path cannot be enabled in the **bgp** context within a VPRN. However, BGP add-path can be enabled in the base routing instance for address family VPN-IPv4. This is done on all PEs at group level with the following command:

```

# on PE-1, PE-2, PE-3:
configure {
  router "Base" {
    bgp {
      group "IBGP" {
        family {
          ipv4 true
          vpn-ipv4 true
        }
        add-paths {
          vpn-ipv4 {
            send 2
            receive true
          }
        }
      }
    }
  }
}

```

In this example, BGP add-path is enabled at neighbor level on RR-5, as follows:

```

# on RR-5:
configure {
  router "Base" {
    bgp {

```

```

group "IBGP" {
    family {
        ipv4 true
        vpn-ipv4 true
    }
}
neighbor 192.0.2.1 {
    add-paths {
        vpn-ipv4 {
            send 2
            receive true
        }
    }
}
neighbor 192.0.2.2 {
    add-paths {
        vpn-ipv4 {
            send 2
            receive true
        }
    }
}
neighbor 192.0.2.3 {
    add-paths {
        vpn-ipv4 {
            send 2
            receive true
        }
    }
}
}

```

The BGP configuration for group "IBGP" on PE-1 is as follows:

```

[ex:/configure router "Base" bgp group "IBGP"]
A:admin@PE-1# info
next-hop-self true
peer-as 64496
family {
    ipv4 true
    vpn-ipv4 true
}
add-paths {
    ipv4 {
        send 2
        receive true
    }
    vpn-ipv4 {
        send 2
        receive true
    }
}
}

```

With add-path enabled for address family VPN-IPv4, PE-1 and PE-2 will advertise their route for prefix 172.31.0.0/16 as VPN-IPv4 route to RR-5. RR-5 will advertise both routes to its other RR clients. PE-3 receives two VPN-IPv4 routes for prefix 172.31.0.0/16, as follows:

```

[/]
A:admin@PE-3# show router bgp routes 172.31.0.0/16 vpn-ipv4
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -

```

```
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
```

=====
BGP VPN-IPv4 Routes
=====

Flag	Network	LocalPref	MED
	Nexthop (Router)	Path-Id	IGP Cost
	As-Path		Label
u*>i	64496:1:172.31.0.0/16	200	None
	192.0.2.1	15	10
	64500		524283
ub*i	64496:1:172.31.0.0/16	100	None
	192.0.2.2	17	10
	64500		524284

Routes : 2
=====

Both routes are used: the route via PE-1 is the active route and the route via PE-2 is used as a backup, as indicated by the "b" flag.

The routing table for VPRN 1 on PE-3 shows that there is a backup route for prefix 172.31.0.0/16, as indicated by "B" as follows:

```
[/]
A:admin@PE-3# show router 1 route-table 172.31.0.0/16

Route Table (Service: 1)
=====
Dest Prefix[Flags]                Type  Proto  Age           Pref
Next Hop[Interface Name]          Metric
-----
172.31.0.0/16 [B]                 Remote BGP VPN 00h01m13s 170
192.0.2.1 (tunneled)              10
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```

The active route and the alternative (backup) route are shown in the following output:

```
[/]
A:admin@PE-3# show router 1 route-table 172.31.0.0/16 alternative

Route Table (Service: 1)
=====
Dest Prefix[Flags]                Type  Proto  Age           Pref
Next Hop[Interface Name]          Metric
Alt-NextHop                       Alt-
Metric
-----
172.31.0.0/16                     Remote BGP VPN 00h01m13s 170
192.0.2.1 (tunneled)              10
172.31.0.0/16 (Backup)          Remote BGP VPN 00h01m13s 170
-----
```

192.0.2.2 (tunneled)

10

```
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
      Backup = BGP backup route
      LFA = Loop-Free Alternate nexthop
      S = Sticky ECMP requested
=====
```

BGP FRR is disabled in VPRN 1 on the PEs, as follows:

```
# on PE-1, PE-2, PE-3:
configure {
  service {
    vprn "VPRN 1" {
      delete bgp-vpn-backup
    }
  }
}
```

Add-path for family VPN-IPv4 with ECMP

A different import policy is configured in VPRN 1 on PE-1 to make the cost of the paths via PE-1 and PE-2 equal, as follows:

```
# on PE-1:
configure {
  policy-options {
    prefix-list "172.16.0.0/12" {
      prefix 172.16.0.0/12 type longer {
      }
    }
  }
  policy-statement "import-bgp-vprn" {
    entry 10 {
      from {
        prefix-list ["172.16.0.0/12"]
      }
      action {
        action-type accept
      }
    }
  }
}
service {
  vprn "VPRN 1" {
    bgp {
      group "EBGP_1" {
        delete import
        import {
          policy ["import-bgp-vprn"]
        }
      }
    }
  }
}
```

ECMP is enabled in VPRN 1 on all PEs, as follows:

```
# on PE-1, PE-2, PE-3:
configure {
  service {
    vprn "VPRN 1" {
      ecmp 2
    }
  }
}
```

BGP multipath needs to be enabled in the base routing context, but that already happened.

With ECMP enabled, the two routes that are received on PE-3 from RR-5 are both active, as follows:

```
[/]
A:admin@PE-3# show router bgp routes 172.31.0.0/16 vpn-ipv4
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  64496:1:172.31.0.0/16      100        None
      192.0.2.1              15         10
      64500                    524283
u*>i  64496:1:172.31.0.0/16      100        None
      192.0.2.2              17         10
      64500                    524283
-----
Routes : 2
=====
```

ECMP is enabled with a value of two, so traffic flows in VPRN 1 on PE-3 with destination 172.31.0.0/16 are distributed over two paths: one via PE-1 and another via PE-2, as follows:

```
[/]
A:admin@PE-3# show router 1 route-table 172.31.0.0/16
=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]          Type  Proto  Age           Pref
Next Hop[Interface Name]  Metric
-----
172.31.0.0/16              Remote BGP VPN 00h02m22s 170
      192.0.2.1 (tunneled) 10
172.31.0.0/16              Remote BGP VPN 00h02m22s 170
      192.0.2.2 (tunneled) 10
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

Conclusion

BGP add-path allows BGP speakers to advertise multiple distinct paths for the same prefix. The potential benefits of BGP add-path include reduced routing churn, faster convergence, and better load-sharing.

BGP Add-Path Policy Control

This chapter provides information about BGP add-path policy control.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially based on SR OS Release 15.0.R4, but the CLI in the current edition corresponds to SR OS Release 22.10.R2.

Overview

BGP add-path allows for advertising multiple paths per prefix for faster convergence, load sharing, and reduction of routing churn. See the [BGP Add-Path](#) chapter for more information.

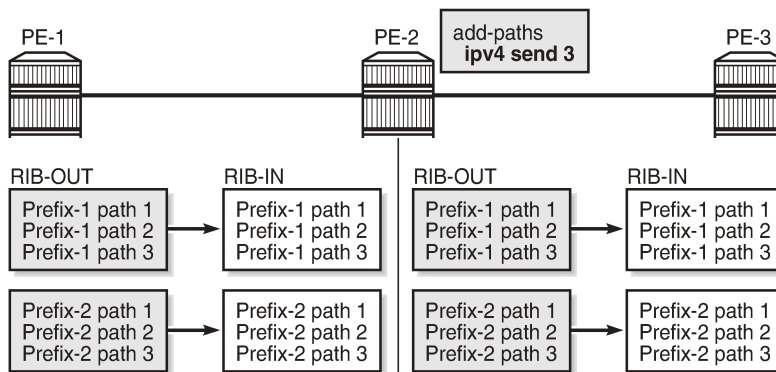
The BGP add-path policy control feature extends the functionality of BGP add-path, which was able to control the number of advertised paths per prefix per address family. This meant that all prefixes that belonged to an address family (such as IPv4, IPv6, and so on) were subject to the same sending limit imposed by the **send** *<number|keyword>* command configured at the BGP instance, group, or neighbor level.

BGP add-path policy control adds the capability to configure the number of advertised paths on a per-prefix basis. The **add-paths-send-limit** route policy action allows overriding the sending limit in the **bgp** context for selected prefixes. This adds finer granularity to BGP add-path, where a global path limit is defined at the relevant BGP level and specific limits can be defined for exceptional prefixes at an import policy level.

A value between 1 and 16 is configurable for **add-paths-send-limit**.

[Figure 48: BGP add-paths before policy control](#) shows a topology for BGP add-paths before policy control.

Figure 48: BGP add-paths before policy control

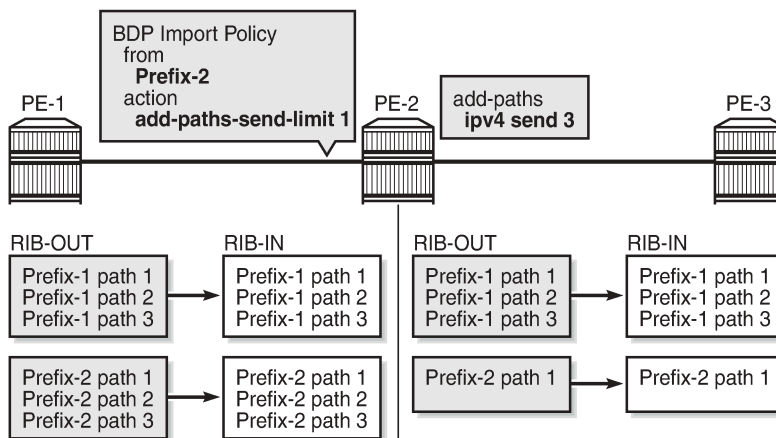


26774

In [Figure 48: BGP add-paths before policy control](#), PE-2 receives two prefixes with three diverse paths from PE-1. PE-2 has a sending limit with a value of 3 configured at a BGP level that is applicable to PE-3. Therefore, PE-2 sends both prefixes with three different path IDs to PE-3.

[Figure 49: BGP add-paths after policy control](#) shows a topology for BGP add-paths after policy control.

Figure 49: BGP add-paths after policy control



26775

In [Figure 49: BGP add-paths after policy control](#), a BGP-import policy is applied on PE-2. The policy selectively applies a sending limit of 1 on the paths received for Prefix-2. Therefore, PE-2 sends only one path for Prefix-2 to PE-3, while the BGP level sending limit of 3 still applies for Prefix-1.

The policy action is only applicable for BGP-import policy and has no effect on BGP-export policy, VRF-import policy, or VRF-export policy. The reason for this is that the policy needs to be applied on the routes accepted into the RIB-IN, otherwise two or more paths may not be present.

The BGP-import policy does not match VPN-IP routes unless the **vpn-apply-import true** command is configured in the BGP global base, group, or neighbor level.



Note:

The route policy only controls the number of advertised paths, not the set of paths.

Configuration

The following configuration examples are in this section:

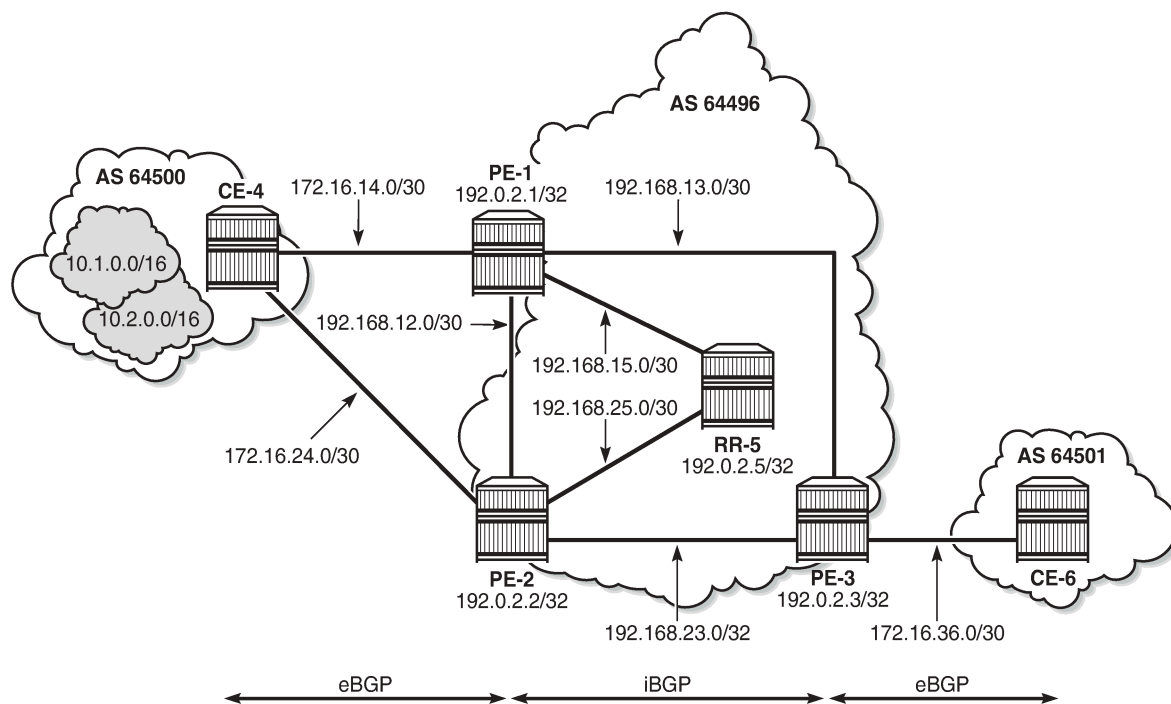
- BGP add-path for address family IPv4 without policy control
- BGP add-path for address family IPv4 with policy control
- BGP add-path for address family VPN-IPv4 with policy control

Example topology

Figure 50: Example topology - IPv4 shows the example topology used for the BGP add-path policy control feature for the IPv4 address family. The topology used is similar to the one in the [BGP Add-Path](#) chapter, with the following characteristics:

- CE-4 in AS 64500 advertises both prefixes 10.1.0.0/16 and 10.2.0.0/16 to its eBGP peers PE-1 and PE-2 in AS 64496.
- RR-5 is route reflector for all PEs in AS 64496.
- add-path is configured on all PE routers and RR-5 with a sending limit of 2.
- CE-6 in AS 64501 peers with PE-3 in AS 64496 and can send traffic to CE-4 in 64500.

Figure 50: Example topology - IPv4



26772

Initial configuration

The initial configuration on all nodes includes:

- Cards, MDAs, ports
- Router interfaces
- IS-IS as IGP on all interfaces within AS 64496 (alternatively, OSPF can be used)
- LDP on all interfaces between the PEs in AS 64496, but not toward RR-5. LDP is used to create the transport tunnels that bind to the VPRN services in the VPN-IPv4 address family section.

BGP is configured on all the nodes. CE-4 peers with PE-1 and PE-2 and exports prefixes 10.1.0.0/16 and 10.2.0.0/16 to both eBGP peers, as follows:

```
# on CE-4:
configure {
  router "Base" {
    autonomous-system 64500
    bgp {
      rapid-withdrawal true
      split-horizon true
      ebgp-default-reject-policy {
        import false
        export false
      }
      group "eBGP" {
        peer-as 64496
        export {
          policy ["export-bgp"]
        }
      }
      neighbor "172.16.14.1" {
        group "eBGP"
      }
      neighbor "172.16.24.1" {
        group "eBGP"
      }
    }
    interface "int-loopback-1" {
      loopback
      ipv4 {
        primary {
          address 10.1.1.1
          prefix-length 16
        }
      }
    }
    interface "int-loopback-2" {
      loopback
      ipv4 {
        primary {
          address 10.2.1.1
          prefix-length 16
        }
      }
    }
  }
  policy-options {
    prefix-list "10.1.0.0/16" {
      prefix 10.1.0.0/16 type longer {
      }
    }
  }
}
```


The BGP configuration on PE-2 and PE-3 is similar to that of PE-1.

RR-5 acts as a route reflector to all the PEs in AS 64500 with a cluster ID of 5.5.5.5. The configuration on RR-5 is as follows:

```
# on RR-5:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      rapid-withdrawal true
      split-horizon true
      ebgp-default-reject-policy {
        import false
        export false
      }
      group "iBGP" {
        peer-as 64496
        cluster {
          cluster-id 5.5.5.5
        }
        add-paths {
          ipv4 {
            send 2
            receive true
          }
        }
      }
      neighbor "192.0.2.1" {
        group "iBGP"
      }
      neighbor "192.0.2.2" {
        group "iBGP"
      }
      neighbor "192.0.2.3" {
        group "iBGP"
      }
    }
  }
}
```

BGP add-path for address family IPv4 without policy control

RR-5 receives both the 10.1.0.0/16 and 10.2.0.0/16 prefixes with two paths from PE-1 and PE-2:

```
[/]
A:admin@RR-5# show router bgp routes
=====
BGP Router ID:192.0.2.5      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
```

```

-----
u*>i 10.1.0.0/16          100      None
      192.0.2.1          2         10
      64500              -
*i   10.1.0.0/16          100      None
      192.0.2.2          2         10
      64500              -
u*>i 10.2.0.0/16          100      None
      192.0.2.1          1         10
      64500              -
*i   10.2.0.0/16          100      None
      192.0.2.2          1         10
      64500              -
-----
Routes : 4
=====

```

RR-5 propagates these updates to its clients, for example to PE-3, as follows:

```

12 2023/01/26 12:57:20.218 CET MINOR: DEBUG #2001 Base Peer 1: 192.0.2.3
"Peer 1: 192.0.2.3: UPDATE
Peer 1: 192.0.2.3 - Send BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 41
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 6 AS Path:
    Type: 2 Len: 1 < 64500 >
  Flag: 0x40 Type: 3 Len: 4 Nexthop: 192.0.2.1
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
  Flag: 0x80 Type: 9 Len: 4 Originator ID: 192.0.2.1
  Flag: 0x80 Type: 10 Len: 4 Cluster ID:
    5.5.5.5
  NLRI: Length = 14
    10.1.0.0/16 Path-ID 11
    10.2.0.0/16 Path-ID 12
"

```

```

3 2023/01/26 12:56:53.218 CET MINOR: DEBUG #2001 Base Peer 1: 192.0.2.3
"Peer 1: 192.0.2.3: UPDATE
Peer 1: 192.0.2.3 - Send BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 41
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 6 AS Path:
    Type: 2 Len: 1 < 64500 >
  Flag: 0x40 Type: 3 Len: 4 Nexthop: 192.0.2.2
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
  Flag: 0x80 Type: 9 Len: 4 Originator ID: 192.0.2.2
  Flag: 0x80 Type: 10 Len: 4 Cluster ID:
    5.5.5.5
  NLRI: Length = 14
    10.1.0.0/16 Path-ID 1
    10.2.0.0/16 Path-ID 2
"

```

PE-3 receives both prefixes in its BGP routing table with two different paths (also, optionally, has ECMP and BGP multipath enabled as described in the [BGP Add-Path](#) chapter):

```

# on PE-3:
configure {
  router "Base" {

```

```

    ecmp 2
    bgp {
        multipath {
            max-paths 2
        }
    }
}

```

```

[/]
A:admin@PE-3# show router bgp routes
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  10.1.0.0/16              100        None
      192.0.2.1             11         10
      64500                  -
u*>i  10.1.0.0/16              100        None
      192.0.2.2             1          10
      64500                  -
u*>i  10.2.0.0/16              100        None
      192.0.2.1             12         10
      64500                  -
u*>i  10.2.0.0/16              100        None
      192.0.2.2             2          10
      64500                  -
-----
Routes : 4
=====

```

BGP add-path for address family IPv4 with policy control

The following policy is enabled on RR-5, which limits the number of advertised paths for prefix 10.2.0.0/16 to one:

```

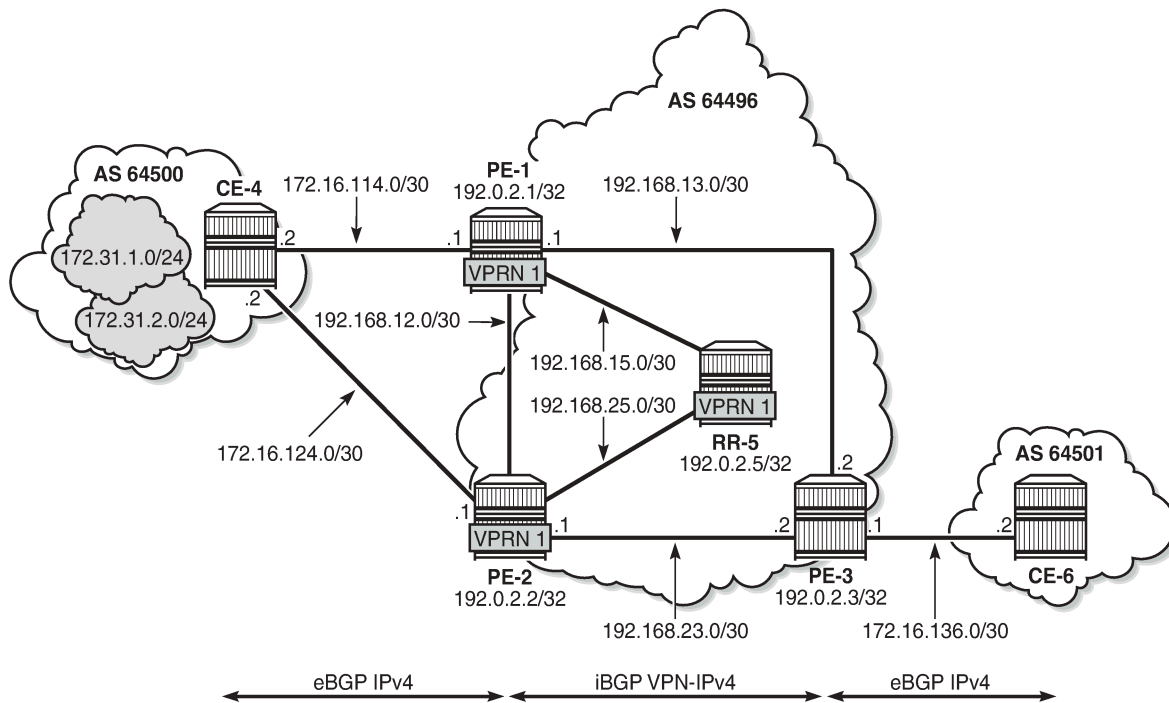
# on RR-5
configure {
    policy-options {
        prefix-list "10.2.0.0/16" {
            prefix 10.2.0.0/16 type longer {
            }
        }
    }
    policy-statement "import-add-path" {
        entry 10 {
            from {
                prefix-list ["10.2.0.0/16"]
            }
            action {

```


BGP add-path for address family VPN-IPv4 with policy control

Figure 51: Example topology - VPN-IPv4 shows the example topology used for the BGP add-path policy control feature for VPN-IPv4 route family. The topology used is similar to the one used in the [BGP Add-Path](#) chapter. CE-4 exports both prefixes 172.31.1.0/24 and 172.31.2.0/24 to VPRN 1 on PE-1 and PE-2.

Figure 51: Example topology - VPN-IPv4



26777

VPRN 1 is configured on all PEs in AS 64496. The configuration of VPRN 1 is similar on all PEs; for example, for PE-1, the VPRN configuration is as follows:

```
# on PE-1:
configure {
  service {
    vprn "VPRN 1" {
      admin-state enable
      service-id 1
      customer "1"
      autonomous-system 64496
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "64496:1"
          vrf-target {
            community "target:64496:1"
          }
          auto-bind-tunnel {
            resolution any
          }
        }
      }
    }
  }
  bgp {
```


The configuration on CE-6 is similar, but no prefix is exported from CE-6.

For all BGP speakers in AS 64496, BGP must be configured for address family VPN-IPv4 as well as for IPv4, as follows:

```
# on PE-1, PE-2, PE-3, RR-5:
configure {
  router "Base" {
    bgp {
      group "iBGP" {
        family {
          ipv4 true
          vpn-ipv4 true
        }
      }
    }
  }
}
```

BGP add-path cannot be enabled in the **bgp** context within a VPRN. However, it can be enabled in the base routing instance for address family VPN-IPv4. This is done on all PEs and RR-5 at group level with the following configuration:

```
# on PE-1, PE-2, PE-3, RR-5:
configure exclusive
router "Base" {
  bgp {
    group "iBGP" {
      add-paths {
        vpn-ipv4 {
          send 2
          receive true
        }
      }
    }
  }
}
```

The BGP configuration on PE-1 is as follows:

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      rapid-withdrawal true
      split-horizon true
      ebgp-default-reject-policy {
        import false
        export false
      }
      group "eBGP" {
        peer-as 64500
      }
      group "iBGP" {
        next-hop-self true
        peer-as 64496
        family {
          ipv4 true
          vpn-ipv4 true
        }
        add-paths {

```



```

service {
  vprn "VPRN 1" {
    ecmp 2
  }
}

```

Alternatively, BGP FRR can be enabled for VPRN 1, as described in the [BGP Add-Path](#) chapter.

To limit the advertisement of prefix 172.31.2.0/24 to a single path, the following route policy is configured on RR-5:

```

# on RR-5:
configure {
  policy-options {
    prefix-list "172.31.2.0/24" {
      prefix 172.31.2.0/24 type longer {
      }
    }
  }
  policy-statement "import-add-path" {
    entry 20 {
      from {
        prefix-list ["172.31.2.0/24"]
      }
      action {
        action-type accept
        add-paths-send-limit 1
      }
    }
  }
}

```

The policy entry for prefix 172.31.2.0/24 can be configured in a new policy-statement or be added to an existing BGP policy (used for the previous IPv4 add-path policy section, for example).

If this is a new policy-statement, apply the policy in the **group "iBGP"** context on RR-5:

```

# on RR-5:
configure {
  router "Base" {
    bgp {
      group "iBGP" {
        import {
          policy ["import-add-path"]
        }
      }
    }
  }
}

```

At this point, PE-3 still has two paths for each of the prefixes:

```

[/]
A:admin@PE-3# show router bgp routes 172.31.0.0/16 vpn-ipv4 longer
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

```

```

=====
BGP VPN-IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                   Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  64496:1:172.31.1.0/24                 100        None
      192.0.2.1                          21         10
      64500                                524283
u*>i  64496:1:172.31.1.0/24                 100        None
      192.0.2.2                          9          10
      64500                                524283
u*>i  64496:1:172.31.2.0/24                 100        None
      192.0.2.1                          20         10
      64500                                524283
u*>i  64496:1:172.31.2.0/24                 100        None
      192.0.2.2                          10         10
      64500                                524283
-----
Routes : 4
=====

```

The following configuration is applied on RR-5 to make the BGP policy effective for VPN-IPV4 routes:

```

# on RR-5:
configure {
  router "Base" {
    bgp {
      vpn-apply-import true
    }
  }
}

```

Upon application of this configuration, RR-5 sends the following withdrawal to PE-3:

```

57 2023/01/26 13:14:03.218 CET MINOR: DEBUG #2001 Base Peer 1: 192.0.2.3
"Peer 1: 192.0.2.3: UPDATE
Peer 1: 192.0.2.3 - Send BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 26
  Flag: 0x90 Type: 15 Len: 22 Multiprotocol Unreachable NLRI:
  Address Family VPN_IPV4
  172.31.2.0/24 RD 64496:1 Label 0 (Raw label 0x1) Path-ID 10
"

```

PE-3 now has a single route for prefix 172.31.2.0/24 in its BGP routing table:

```

[/]
A:admin@PE-3# show router bgp routes 172.31.0.0/16 vpn-ipv4 longer
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====

```


Flag	Network Nexthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
u*>i	64496:1:172.31.1.0/24 192.0.2.1 64500	100 21	None 10 524283
u*>i	64496:1:172.31.1.0/24 192.0.2.2 64500	100 9	None 10 524283
u*>i	64496:1:172.31.2.0/24 192.0.2.1 64500	100 20	None 10 524283

Routes : 3
=====

PE-3 has installed a single route for prefix 172.31.2.0/24 in its VPRN route table:

```
[/]
A:admin@PE-3# show router 1 route-table 172.31.0.0/16 longer

=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]          Type  Proto  Age           Pref
Next Hop[Interface Name]  Metric
-----
172.31.1.0/24              Remote BGP VPN 00h06m31s 170
192.0.2.1 (tunneled)      10
172.31.1.0/24              Remote BGP VPN 00h06m31s 170
192.0.2.2 (tunneled)      10
172.31.2.0/24              Remote BGP VPN 00h01m26s 170
192.0.2.1 (tunneled)      10
-----
No. of Routes: 3
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```

Conclusion

The BGP add-path policy control feature allows BGP speakers to advertise multiple distinct paths for the same prefix. The potential benefits of using BGP add-path policy control are increased granularity and flexibility in advertising multiple paths to BGP neighbors.

BGP Autonomous System Override

This chapter describes BGP Autonomous System Override.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

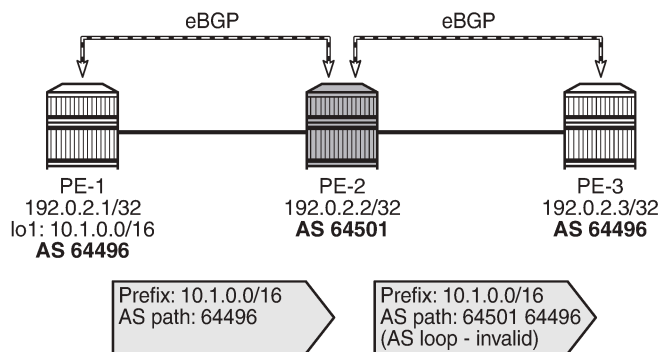
The information and configuration in this chapter are based on SR OS Release 20.5.R1. In SR OS releases earlier than 19.7.R1, BGP Autonomous System (AS) override is only supported in VPRN BGP instances; BGP AS override in the base router is supported in SR OS Release 19.7.R1 and later.

Overview

In some network designs, the same Autonomous System Number (ASN) is reused at different sites or regions that are interconnected by a common service or backbone. This can occur when an enterprise buys an IP VPN service to connect various sites that, in the past, were operated as a single ASN. This can also occur when a service provider builds a common backbone to interconnect regional networks that, for simplicity, reuse the same ASN.

This type of interconnectivity creates a problem because a BGP route originated by one of the sites and propagated through the backbone will appear as an AS path loop when advertised into another site. Routes with an AS loop are invalid; [Figure 52: PE-2 detects AS-path loop and advertises the route to PE-3 as invalid](#) shows an example. PE-2 in AS 64501 receives a BGP route from PE-1 in AS 64496. PE-2 detects that the ASN 64496 in the BGP AS-path attribute equals the ASN of its peer PE-3, so it detects an AS loop and advertises this route to PE-3 as an invalid route.

Figure 52: PE-2 detects AS-path loop and advertises the route to PE-3 as invalid



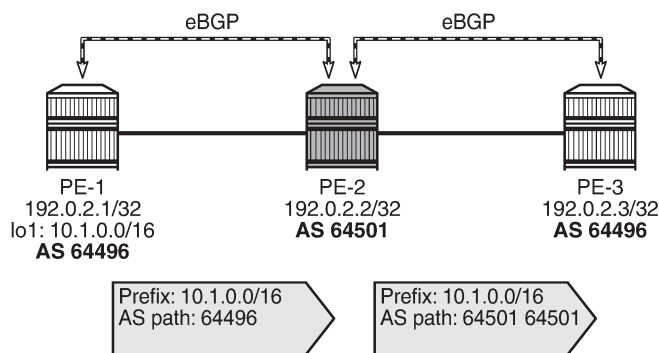
36187

There are different solutions to this problem:

- Use different ASNs per site or region. From an operational point of view, this is a major change in an existing network.
- Disable AS path loop detection within each region. This is not encouraged in case you have external peering to the outside world. Any loops formed between these paths would be undetected.
- Configure the base router or the VPRN instance with BGP AS override.

Most operators prefer to use BGP AS override. A router configured to use BGP AS override on a BGP session monitors outbound routes toward that peer. Whenever a route has the ASN of the peer in its AS-path, all occurrences of this ASN are replaced by the local ASN of the router (or its confederation ID, if the peer is outside the confederation). [Figure 53: BGP AS override replaces the peer ASN in the AS-path with the local ASN](#) shows that PE-2 has replaced ASN 64496 in the AS-path attribute of the BGP route toward PE-3 with its own ASN 64501.

Figure 53: BGP AS override replaces the peer ASN in the AS-path with the local ASN



36188

BGP AS override applies to all supported address families and is supported whether the session is confed-EBGP or EBGP.

The **as-override** command is configurable in the BGP group or neighbor context, both for the base router and the VPRNs.

In SR OS, AS path loop detection is enabled by default. Several actions can be configured when detecting an AS path loop, but those actions are out of the scope of this chapter:

```
configure router bgp / group / neighbor loop-detect
    {drop-peer|ignore-loop|off|discard-route}
```

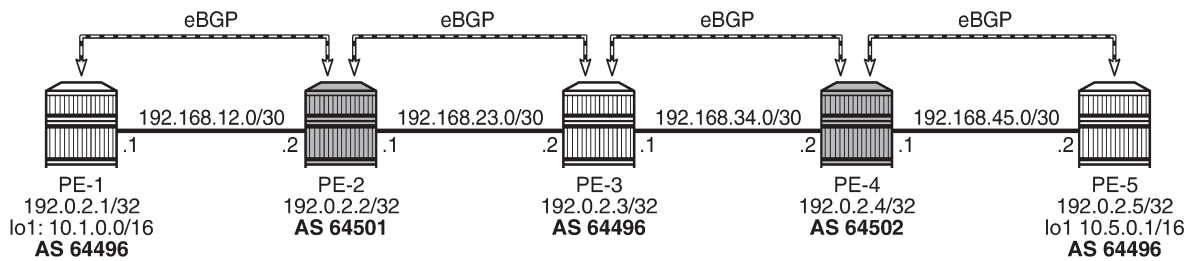
```
configure service vprn bgp / group / neighbor loop-detect
    {drop-peer|ignore-loop|off|discard-route}
```

With the **ignore-loop** parameter configured, the BGP routes are ignored when having an AS-loop flag but BGP peering remains established.

Configuration

[Figure 54: Example topology](#) shows the example topology with five routers: PE-1, PE-3, and PE-5 in AS 64496, PE-2 in AS 64501, and PE-4 in AS 64502.

Figure 54: Example topology



36189

The initial configuration includes:

- Cards, MDAs, ports
- Router interfaces
- EBGP sessions between the nodes

The initial BGP configuration on PE-2 is as follows.

```
# on PE-2:
configure {
  policy-options {
    community "1:1" {
      member "1:1" { }
    }
    policy-statement "1:1" {
      entry 10 {
        from {
          community {
            name "1:1"
          }
        }
        action {
          action-type accept
        }
      }
    }
  }
  router "Base" {
    autonomous-system 64501
    bgp {
      split-horizon true
      group "eBGP" {
        family {
          ipv4 true
        }
        import {
          policy ["1:1"]
        }
        export {
          policy ["1:1"]
        }
      }
      neighbor "192.168.12.1" {
        group "eBGP"
        peer-as 64496
      }
      neighbor "192.168.23.2" {
```

```

        group "eBGP"
        peer-as 64496
    }
}

```

The BGP configuration on the other nodes is similar.

In this chapter, two examples are shown:

- BGP AS override in the base router
- BGP AS override in a VPRN

Default: BGP AS override disabled in base router

By default, BGP AS override is not configured for a BGP group or BGP neighbor; this is verified on PE-2 as follows:

```

[]
A:admin@PE-2# show router bgp neighbor 192.168.12.1 detail | match "AS Override"
Multihop           : 0 (Default)      AS Override       : Disabled

```

```

[]
A:admin@PE-2# show router bgp neighbor 192.168.23.2 detail | match "AS Override"
Multihop           : 0 (Default)      AS Override       : Disabled

```

PE-1 exports BGP route 10.1.0.0/16, defined as a loopback interface in the base routing instance. The configuration is as follows:

```

# on PE-1:
configure {
    policy-options {
        community "1:1" {
            member "1:1" { }
        }
        prefix-list "10.1.0.0/16" {
            prefix 10.1.0.0/16 type longer {
            }
        }
        policy-statement "export-prefix_10.1" {
            entry 10 {
                from {
                    prefix-list ["10.1.0.0/16"]
                }
                to {
                    protocol {
                        name [bgp]
                    }
                }
                action {
                    action-type accept
                    community {
                        add ["1:1"]
                    }
                }
            }
        }
        policy-statement "1:1" {
            entry 10 {
                from {

```



```

=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                  l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                       Path-Id    IGP Cost
      As-Path
-----
i    10.1.0.0/16                             n/a        None
      192.168.23.1                           None        0
      64501 64496                             -
-----
Routes : 1
=====

```

PE-3 receives this route with the following flags:

```

[]
A:admin@PE-3# show router bgp routes hunt | match Flags
Flags          : Invalid IGP AS-Loop

```

Normal BGP rules do not allow invalid routes to be advertised, so PE-3 does not advertise any route to PE-4, as follows:

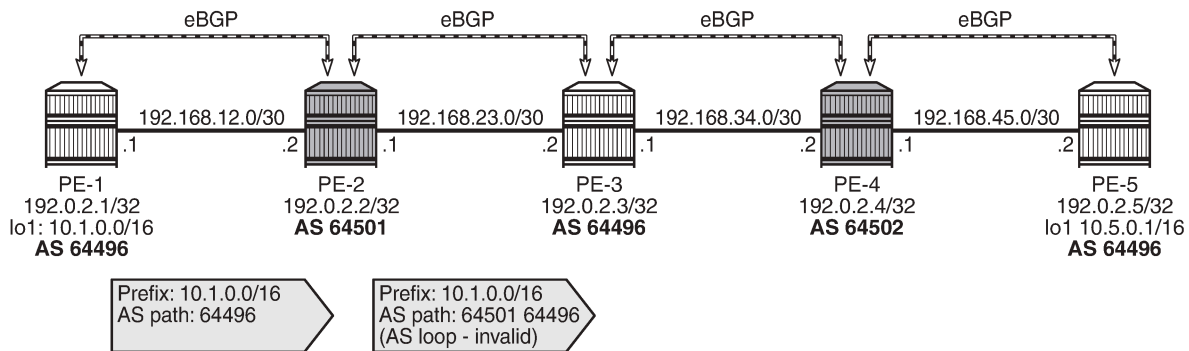
```

[]
A:admin@PE-3# show router bgp neighbor 192.168.34.2 advertised-routes
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                  l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                       Path-Id    IGP Cost
      As-Path
-----
No Matching Entries Found.
=====

```

Figure 55: PE-2 detects AS loop and advertises a route to PE-3 as invalid shows the BGP routes advertised by PE-1 and PE-2 with the corresponding AS-path.

Figure 55: PE-2 detects AS loop and advertises a route to PE-3 as invalid



36190

BGP AS override in base router

On PE-2 and PE-4, the following command configures BGP AS override in the group "eBGP":

```
# on PE-2, PE-4:
configure {
  router "Base" {
    bgp {
      group "eBGP" {
        as-override true
      }
    }
  }
}
```

With this configuration, BGP AS override is configured for both BGP neighbors, as follows:

```
[]
A:admin@PE-2# show router bgp neighbor 192.168.12.1 detail | match "AS Override"
Multihop           : 0 (Default)      AS Override       : Enabled
```

```
[]
A:admin@PE-2# show router bgp neighbor 192.168.23.2 detail | match "AS Override"
Multihop           : 0 (Default)      AS Override       : Enabled
```

PE-2 receives the route from PE-1 with ASN 64496, as follows:

```
[]
A:admin@PE-2# show router bgp routes 10.1.0.0/16
=====
BGP Router ID:192.0.2.2      AS:64501      Local AS:64501
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                                     LocalPref  MED
```



```

      Nexthop (Router)
      As-Path
-----
u*>i 10.1.0.0/16
      192.168.12.1
      64496
-----
      Path-Id
      None
      None
-----
      IGP Cost
      Label
      None
      0
      -
-----
Routes : 1
=====

```

Instead of advertising a route with an AS loop, PE-2 will now replace ASN 64496 in the AS-path attribute with its own ASN 64501, so PE-3 receives the following valid route:

```

[]
A:admin@PE-3# show router bgp routes 10.1.0.0/16
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network
      Nexthop (Router)
      As-Path
-----
u*>i 10.1.0.0/16
      192.168.23.1
      64501 64501
-----
      LocalPref
      Path-Id
      None
      None
-----
      MED
      IGP Cost
      Label
      None
      0
      -
-----
Routes : 1
=====

```

PE-4 receives the following BGP route:

```

[]
A:admin@PE-4# show router bgp routes 10.1.0.0/16
=====
BGP Router ID:192.0.2.4      AS:64502      Local AS:64502
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network
      Nexthop (Router)
      As-Path
-----
u*>i 10.1.0.0/16
      192.168.34.1
      64496 64501 64501
-----
      LocalPref
      Path-Id
      None
      None
-----
      MED
      IGP Cost
      Label
      None
      0
      -
-----
Routes : 1
=====

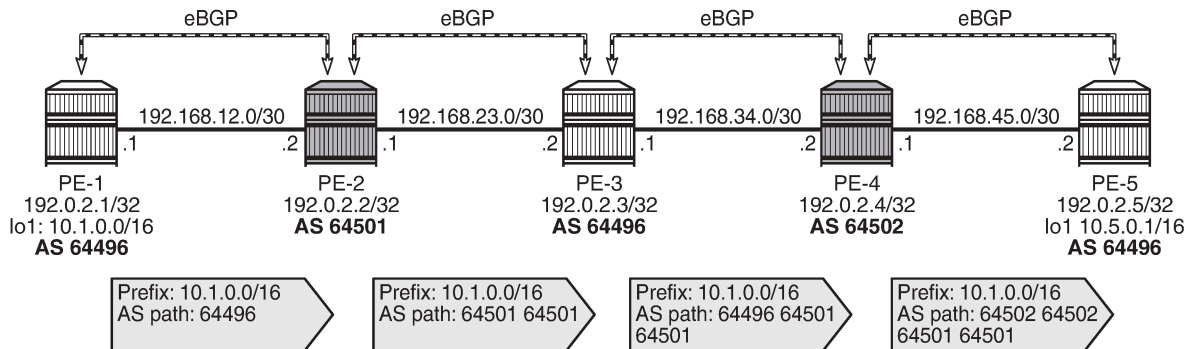
```

PE-4 detects an AS loop when advertising this route to its peer PE-5 in AS 64496, so it replaces ASN 64496 in the AS-path with its own ASN 64502. PE-5 receives the following valid route from PE-4:

```
[ ]
A:admin@PE-5# show router bgp routes 10.1.0.0/16
=====
BGP Router ID:192.0.2.5      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
  Nextthop (Router)                       Path-Id   IGP Cost
  As-Path                                   Label
-----
u*>i 10.1.0.0/16                            None     None
      192.168.45.1                          None     0
      64502 64502 64501 64501
-----
Routes : 1
=====
```

Figure 56: No AS loop when BGP AS override is enabled for group "eBGP" on PE-2 and PE-4 shows the BGP routes advertised by the PEs with the corresponding AS-path.

Figure 56: No AS loop when BGP AS override is enabled for group "eBGP" on PE-2 and PE-4

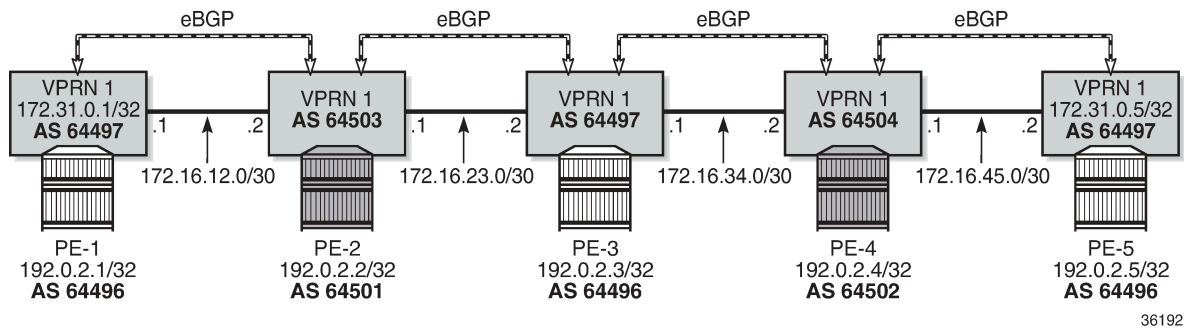


36191

Default: BGP AS override disabled in VPRN

Figure 57: Example topology with VPRN 1 on all PEs shows the example topology with VPRN 1 configured on all PEs.

Figure 57: Example topology with VPRN 1 on all PEs



On PE-2, VPRN 1 is configured as follows. By default, **as-override** is not configured for any BGP group or BGP neighbor.

```
# on PE-2:
configure {
  service {
    vprn "VPRN 1" {
      admin-state enable
      service-id 1
      customer "1"
      autonomous-system 64503
      router-id 172.31.0.2
      route-distinguisher "64503:1"
      vrf-target {
        community "target:1:1"
      }
    }
    bgp {
      split-horizon true
      group "eBGP" {
        peer-as 64497
        local-as {
          as-number 64503
        }
        import {
          policy ["1:1"]
        }
        export {
          policy ["1:1"]
        }
      }
      neighbor "172.16.12.1" {
        group "eBGP"
      }
      neighbor "172.16.23.2" {
        group "eBGP"
      }
    }
  }
  interface "int-VPRN1-PE-2-PE-1" {
    ipv4 {
      primary {
        address 172.16.12.2
        prefix-length 30
      }
    }
    sap 1/1/2:1 {
    }
  }
}
```

```

    }
    interface "int-VPRN1-PE-2-PE-3" {
        ipv4 {
            primary {
                address 172.16.23.1
                prefix-length 30
            }
        }
        sap 1/1/1:1 {
        }
    }
    interface "system" {
        loopback true
        ipv4 {
            primary {
                address 172.31.0.2
                prefix-length 32
            }
        }
    }
}

```

The service configuration on the other nodes is similar. The IP addresses and ASNs are shown in [Figure 57: Example topology with VPRN 1 on all PEs](#).

VPRN 1 on PE-1 exports BGP route 172.31.0.1/32, defined as a loopback interface within the VPRN 1 routing instance. The configuration is as follows:

```

# on PE-1:
configure {
    policy-options {
        prefix-list "172.31.0.0/16" {
            prefix 172.31.0.0/16 type longer {
            }
        }
    }
    policy-statement "export-prefix_172.31" {
        entry 10 {
            from {
                prefix-list ["172.31.0.0/16"]
                protocol {
                    name [direct]
                }
            }
            to {
                protocol {
                    name [bgp]
                }
            }
            action {
                action-type accept
                community {
                    add ["1:1"]
                }
            }
        }
    }
}
service {
    vprn "VPRN 1" {
        admin-state enable
        service-id 1
        customer "1"
        autonomous-system 64497
        router-id 172.31.0.1
    }
}

```

```

route-distinguisher "64497:1"
vrf-target {
  community "target:1:1"
}
bgp {
  split-horizon true
  group "eBGP" {
    peer-as 64503
    local-as {
      as-number 64497
    }
    import {
      policy ["1:1"]
    }
  }
  neighbor "172.16.12.2" {
    group "eBGP"
    export {
      policy ["export-prefix_172.31"]
    }
  }
}
interface "int-VPRN1-PE-1-PE-2" {
  ipv4 {
    primary {
      address 172.16.12.1
      prefix-length 30
    }
  }
  sap 1/1/1:1 {
  }
}
interface "system" {
  loopback true
  ipv4 {
    primary {
      address 172.31.0.1
      prefix-length 32
    }
  }
}

```

VPRN 1 on PE-1 exports route 172.31.0.1/32 with ASN 64497 to VPRN 1 on PE-2. On PE-2, the following route is received in VPRN 1:

```

[ ]
A:admin@PE-2# show router 1 bgp neighbor 172.16.12.1 received-routes
=====
BGP Router ID:172.31.0.2      AS:64503      Local AS:64503
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                       Path-Id    IGP Cost
      As-Path                                Label
-----
u*>i  172.31.0.1/32                            n/a        None

```

```

172.16.12.1
64497
-----
Routes : 1
=====

```

ASN 64497 equals the peer AS of PE-3, so an AS loop is detected, and the following route is advertised to VPRN 1 on PE-3 as invalid:

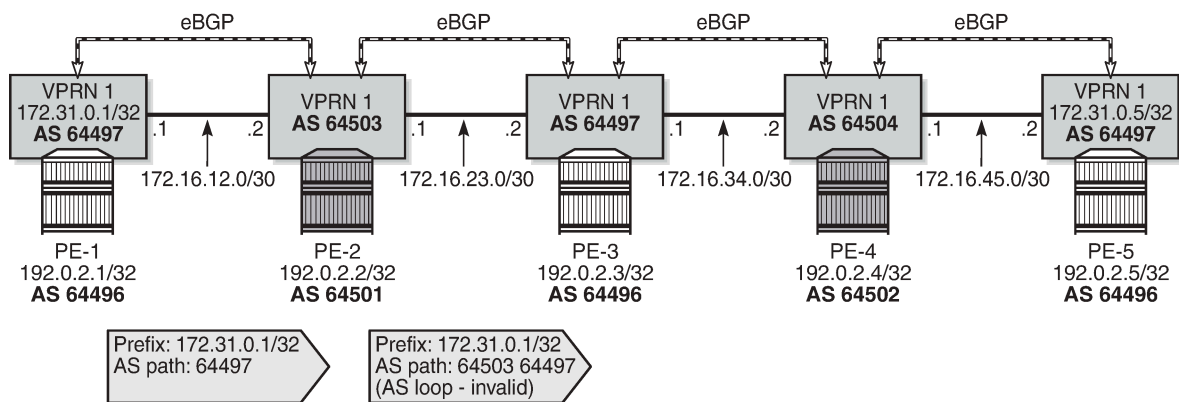
```

[]
A:admin@PE-2# show router 1 bgp neighbor 172.16.23.2 advertised-routes
=====
BGP Router ID:172.31.0.2      AS:64503      Local AS:64503
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                LocalPref  MED
  Nexthop (Router)          Path-Id    IGP Cost
  As-Path                   Label
-----
i  172.31.0.1/32            n/a        None
   172.16.23.1             None        0
   64503 64497              -
-----
Routes : 1
=====

```

Figure 58: AS loop when BGP AS override is not configured in VPRN 1 on PE-2 shows the routes sent by VPRN 1 on PE-1 and PE-2. PE-3 receives an invalid route with an AS loop that is not re-advertised.

Figure 58: AS loop when BGP AS override is not configured in VPRN 1 on PE-2



36193

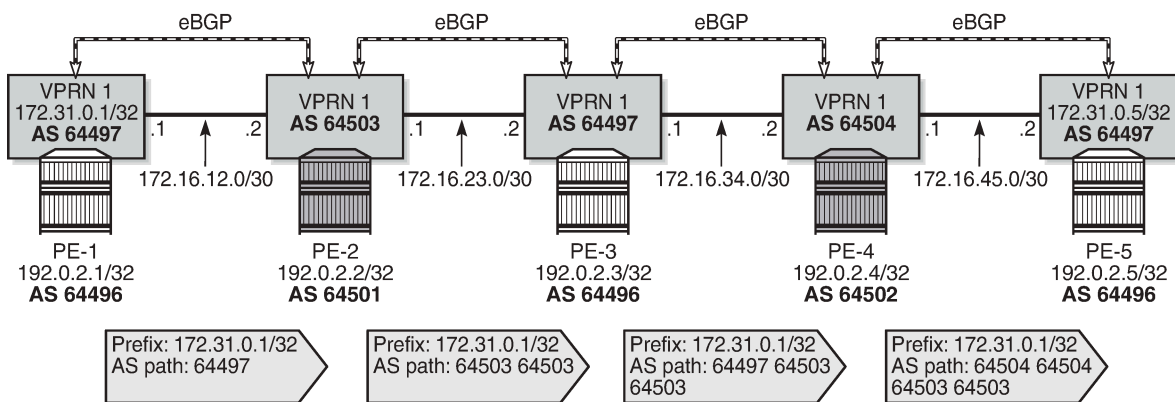
BGP AS override in VPRN

On PE-2 and PE-4, **as-override** is enabled in VPRN 1 for group "eBGP", as follows:

```
# on PE-2, PE-4:
configure {
  service {
    vprn "VPRN 1" {
      bgp {
        group "eBGP" {
          as-override true
        }
      }
    }
  }
}
```

Figure 59: Routes advertised when BGP AS override is enabled in VPRN 1 on the PEs shows the routes advertised in VPRN 1 on the PEs when BGP AS override is enabled on PE-2 and PE-4.

Figure 59: Routes advertised when BGP AS override is enabled in VPRN 1 on the PEs



36194

VPRN 1 on PE-2 receives the route with ASN 64497:

```
[ ]
A:admin@PE-2# show router 1 bgp routes 172.31.0.1/32
=====
BGP Router ID:172.31.0.2      AS:64503      Local AS:64503
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nextthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i 172.31.0.1/32             None       None
      172.16.12.1           None       0
      64497                  -
-----
Routes : 1
```

With AS override enabled, VPRN 1 on PE-3 receives the following valid route where ASN 64497 is replaced by ASN 64503:

```
[ ]
A:admin@PE-3# show router 1 bgp routes 172.31.0.1/32
=====
BGP Router ID:192.0.2.3      AS:64497      Local AS:64497
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  172.31.0.1/32             None       None
      172.16.23.1           None       0
      64503 64503
-----
Routes : 1
=====
```

VPRN 1 on PE-4 receives the following route:

```
[ ]
A:admin@PE-4# show router 1 bgp routes 172.31.0.1/32
=====
BGP Router ID:172.31.0.4    AS:64504      Local AS:64504
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  172.31.0.1/32             None       None
      172.16.34.1           None       0
      64497 64503 64503
-----
Routes : 1
=====
```

VPRN 1 on PE-4 replaces ASN 64497 with its own ASN 64504, so PE-5 receives the following valid route with AS-path <64504 64504 64503 64503>:

```
[ ]
A:admin@PE-5# show router 1 bgp routes 172.31.0.1/32
=====
```



```

BGP Router ID:172.31.0.5      AS:64497      Local AS:64497
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i 172.31.0.1/32              None       None
      172.16.45.1            None       0
      64504 64504 64503 64503
      -
-----
Routes : 1
=====

```

Conclusion

BGP AS override can prevent AS loops in network designs where different sites or regions are interconnected by a common service or backbone. BGP AS override can be enabled for BGP groups or BGP neighbors, both in the base router and in VPRNs.

BGP Conditional Route Advertisement

This chapter provides information about BGP Conditional Route Advertisement.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

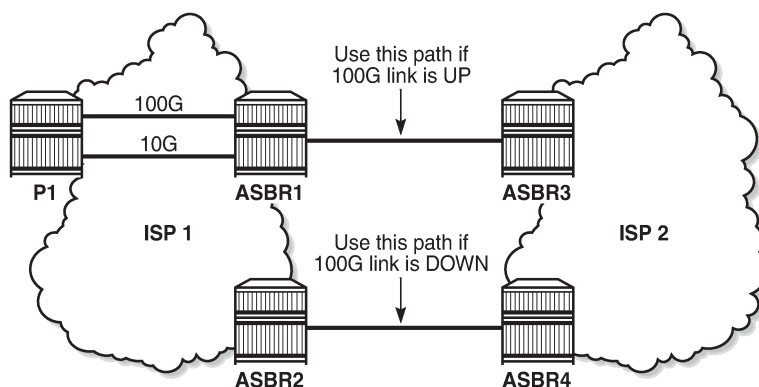
Applicability

The information and configuration in this chapter was originally based on SR OS Release 15.0.R4. The CLI in the current edition is based on SR OS Release 23.3.R2.

Overview

The BGP conditional route advertisement feature allows a router to control the advertisement of routes based on predetermined routes in the route table. [Figure 60: Conditional BGP Route Advertisement - ISP Peering](#) shows an example in which this feature can bring flexibility in an ISP peering scenario.

Figure 60: Conditional BGP Route Advertisement - ISP Peering



26862

ISP 1 and ISP 2 have two peering points; a first between ASBR1 and ASBR3, and a second between ASBR2 and ASBR4. For redundancy, ISP 1 has two links between ASBR1 and the internal P1 router, one with 100 Gb/s and the other with 10 Gb/s capacity. According to the service agreement, ISP 1 instructs ISP 2 to send traffic using the upper path (between ASBR1 and ASBR3) only if the 100 Gb/s link between P1 and ASBR1 is up. If this is not the case, ISP 2 uses the lower path.

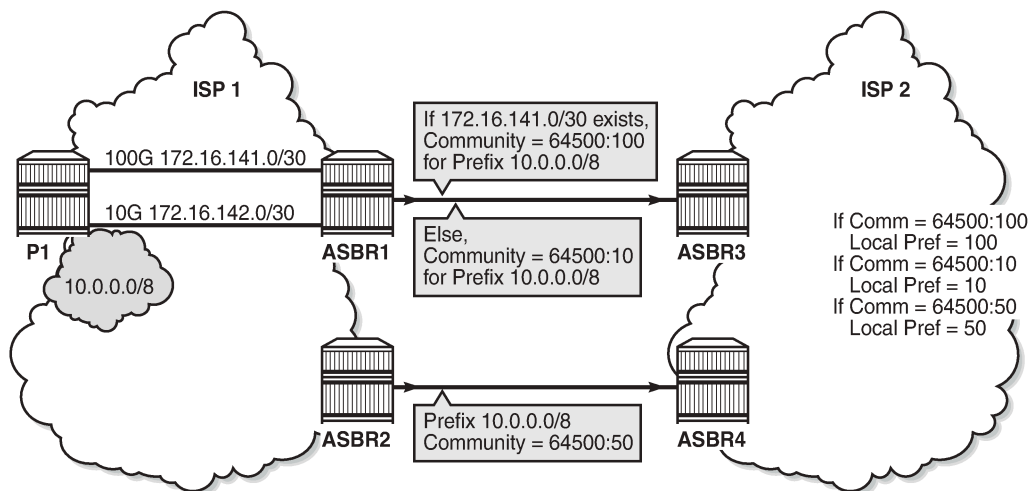
To implement the BGP conditional route advertisement feature, a conditional route policy entry is used. The route policy is as follows:

- Within a **policy-statement** entry, a conditional expression is created.

- The conditional expression tests for active IPv4 or IPv6 routes defined in a prefix list.
- If the expression is true, the **action** commands of the **policy** entry are applied.
- If the expression is false, the entire **policy** entry is skipped and processing continues with the next **policy** entry.
- Conditional expressions are only applicable when the route policy is used as a BGP export policy or a VRF export policy.

Figure 61: Conditional BGP Route Advertisement Implementation Example shows the implementation using the example in Figure 60: Conditional BGP Route Advertisement - ISP Peering.

Figure 61: Conditional BGP Route Advertisement Implementation Example



26863

The prefix of the 100G interface between ASBR1 and P1 is 172.16.141.0/30. ASBR1 receives prefix 10.0.0.0/8 from P1 via BGP. Under standard conditions, the 100G interface is up and 172.16.141.0/30 exists in the route table and ASBR1 advertises 10.0.0.0/8 with a community value of 64500:100. ASBR2 advertises the same prefix with a community value of 64500:50. ASBR3 and ASBR4 in ISP 2 use an import policy that applies local preference values of 100 and 50 on the routes advertised by ASBR1 and ASBR2, respectively. As a result, the routers in ISP 2 prefer ASBR3 as an exit point for traffic flowing toward ISP 1.

If the 100G interface goes down, the prefix 172.16.141.0/30 is withdrawn from the route table and, as a result, ASBR1 starts advertising 10.0.0.0/8 with a community value of 64500:10. ASBR3 and ASBR4 adjust the local preference value for ASBR1 to 10 and, therefore, ASBR4 becomes the preferred exit point for routers in ISP 2.

The only conditional expression that can be contained in a **policy-statement** entry is a route-existence test defined by the **route-exists** keyword in the CLI. The command accepts two parameters: **all** and **none**:

- If neither the **all** nor the **none** parameter is used, the match logic is **any** - that is, the conditional expression is true if any exact match entry in the referenced prefix list has an active route in the route table associated with the policy.
- **all** - the conditional expression is true only if all the exact match entries in the referenced prefix list have an active route in the route table associated with the policy.
- **none** - the conditional expression is true only if none of the exact match entries in the referenced prefix list have an active route in the route table associated with the policy.

Configuration

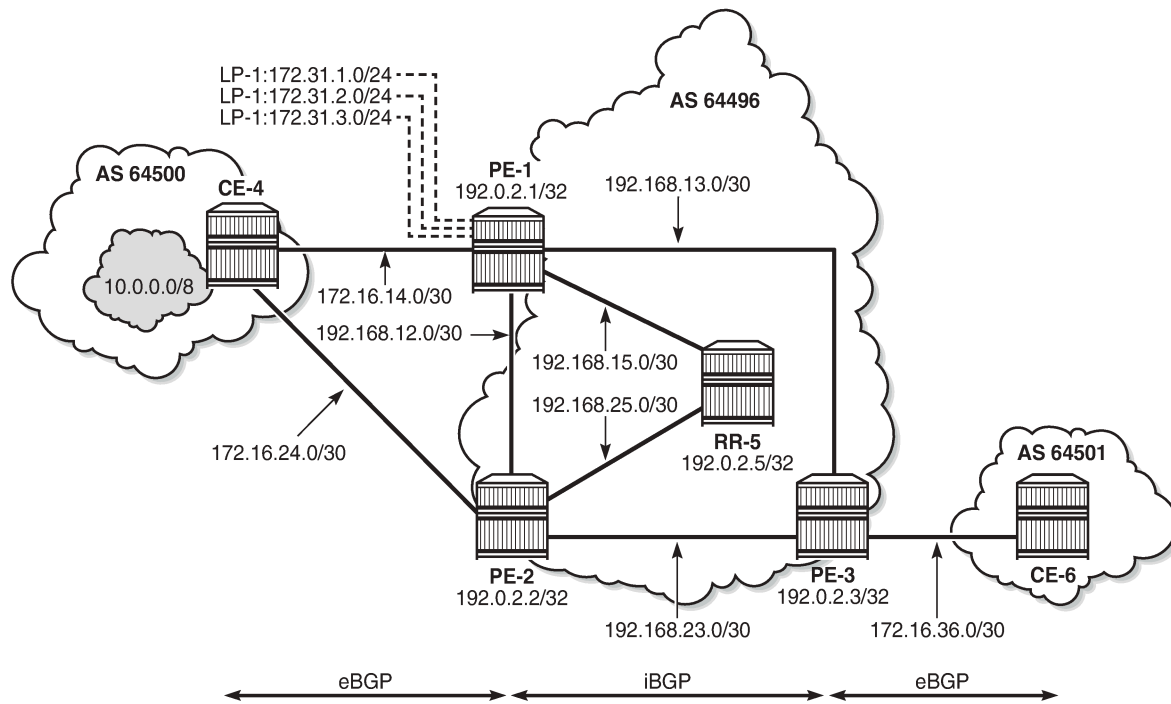
The following configuration examples are covered in this section:

- [BGP Conditional Route Advertisement Using "any" Prefix List Match](#)
- [BGP Conditional Route Advertisement Using "all" Prefix List Match](#)
- [BGP Conditional Route Advertisement Using "none" Prefix List Match](#)

Figure 62: Example Topology shows the example topology for BGP conditional route advertisement with the following characteristics:

- CE-4 in AS 64500 advertises prefix 10.0.0.0/8 to its eBGP peers PE-1 and PE-2 in AS 64496.
- PE-1 has three loopback interfaces configured to demonstrate the use of conditional route advertisement: LP-1, LP-2, and LP-3.
- RR-5 is route reflector for all PEs in AS 64496.
- CE-6 in AS 64501 peers with PE-3 in AS 64496 and can send traffic to CE-4 in 64500.

Figure 62: Example Topology



26864

Initial Configuration

The initial configuration on all nodes includes:

- Cards, MDAs, ports
- LAG configured for the link between CE-4 and PE-1 with two member links

- Router interfaces
- IS-IS as IGP on all interfaces within AS 64496 (alternatively, OSPF can be used)

BGP is configured on all the nodes. CE-4 peers with PE-1 and PE-2 and exports the prefix 10.0.0.0/8 to both eBGP peers, which includes the address of the *int-loopback-1* interface, as follows:

```
# On CE-4:
configure {
  policy-options {
    prefix-list "10.0.0.0/8" {
      prefix 10.0.0.0/8 type longer {
      }
    }
  }
  policy-statement "policy-export-bgp" {
    entry 10 {
      from {
        prefix-list ["10.0.0.0/8"]
      }
      action {
        action-type accept
      }
    }
  }
}
router "Base" {
  autonomous-system 64500
  interface "int-loopback-1" {
    loopback
    ipv4 {
      primary {
        address 10.1.1.1
        prefix-length 8
      }
    }
  }
  bgp {
    rapid-withdrawal true
    split-horizon true
    ebgp-default-reject-policy {
      import false
      export false
    }
    group "eBGP" {
      peer-as 64496
      export {
        policy ["policy-export-bgp"]
      }
    }
    neighbor "172.16.14.1" {
      group "eBGP"
    }
    neighbor "172.16.24.1" {
      group "eBGP"
    }
  }
}
}
```

The BGP configuration on CE-6 is identical, except for the loopback interface and export policy.

PE-1 peers with CE-4 in AS 65400 and RR-5 in AS 64496. The BGP configuration on PE-1 is as follows:

```
# On PE-1:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      rapid-withdrawal true
      split-horizon true
      ebgp-default-reject-policy {
        import false
        export false
      }
      group "eBGP" {
        peer-as 64500
      }
      neighbor "172.16.14.2" {
        group "eBGP"
      }
      group "iBGP" {
        next-hop-self true
        peer-as 64496
      }
      neighbor "192.0.2.5" {
        group "iBGP"
      }
    }
  }
exit all
```

The BGP configuration on PE-2 and PE-3 is similar to that of PE-1.

RR-5 is the route reflector to all the PEs in AS 64500 with a cluster ID of 5.5.5.5. The configuration on RR-5 is as follows:

```
# On RR-5:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      rapid-withdrawal true
      split-horizon true
      ebgp-default-reject-policy {
        import false
        export false
      }
      group "iBGP" {
        cluster {
          cluster-id 5.5.5.5
        }
        peer-as 64496
      }
      neighbor "192.0.2.1" {
        group "iBGP"
      }
      neighbor "192.0.2.2" {
        group "iBGP"
      }
      neighbor "192.0.2.3" {
        group "iBGP"
      }
    }
  }
exit all
```

Three loopback interfaces are configured in PE-1 to be used for route existence tests:

```
# On PE-1:
configure {
  router "Base" {
    interface "int-loopback-1" {
      admin-state enable
      loopback
      ipv4 {
        primary {
          address 172.31.1.1
          prefix-length 24
        }
      }
    }
    interface "int-loopback-2" {
      admin-state enable
      loopback
      ipv4 {
        primary {
          address 172.31.2.1
          prefix-length 24
        }
      }
    }
    interface "int-loopback-3" {
      admin-state enable
      loopback
      ipv4 {
        primary {
          address 172.31.3.1
          prefix-length 24
        }
      }
    }
  }
}
exit all
```

BGP Conditional Route Advertisement Using "any" Prefix List Match

In the initial condition, RR-5 receives the prefix 10.0.0.0/8 from PE-1 and PE-2 with no community values and the default local preference value of 100:

```
[/]
A:admin@RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match '^NextHop|^Community|^Pref'
NextHop      : 192.0.2.1
Local Pref.  : 100                               Interface Name : int-RR-5-PE-1
Community    : No Community Members
NextHop      : 192.0.2.2
Local Pref.  : 100                               Interface Name : int-RR-5-PE-2
Community    : No Community Members
```

The following policy is configured on PE-1 that adds the community 64500:100 to the 10.0.0.0/8 prefix advertised to RR-5 if any of the conditional prefixes in the prefix list are active in the route table:

```
# On PE-1:
configure {
  policy-options {
    community "64500:10" {
      member "64500:10" { }
    }
  }
}
```

```

}
community "64500:100" {
  member "64500:100" { }
}
prefix-list "10.0.0.0/8" {
  prefix 10.0.0.0/8 type longer {
  }
}
prefix-list "prefix-conditional-routes" {
  prefix 172.31.1.0/24 type longer {
  }
  prefix 172.31.2.0/24 type longer {
  }
  prefix 172.31.3.0/24 type longer {
  }
}
policy-statement "policy-bgp-community" {
  entry 10 {
    conditional-expression {
      route-exists "[prefix-conditional-routes]"
    }
    from {
      prefix-list ["10.0.0.0/8"]
    }
    action {
      action-type accept
      community {
        add ["64500:100"]
      }
    }
  }
  entry 20 {
    from {
      prefix-list ["10.0.0.0/8"]
    }
    action {
      action-type accept
      community {
        add ["64500:10"]
      }
    }
  }
}
}
exit all

```

Special attention is required on the policy syntax. The square brackets [...] in the expression of the **route-exists** command are very important.

The following policy is configured on PE-2 that adds the community 64500:50 to the 10.0.0.0/8 prefix advertised to RR-5 without any conditions:

```

# On PE-2:
configure {
  policy-options {
    community "64500:50" {
      member "64500:50" { }
    }
    prefix-list "10.0.0.0/8" {
      prefix 10.0.0.0/8 type longer {
      }
    }
  }
  policy-statement "policy-bgp-community" {
    entry 10 {

```



```

        from {
            prefix-list ["10.0.0.0/8"]
        }
        action {
            action-type accept
            community {
                add ["64500:50"]
            }
        }
    }
}
exit all

```

The policy is applied to the iBGP group on PE-1 and PE-2:

```

# On PE-1 and on PE-2:
configure {
    router "Base" {
        bgp {
            group "iBGP" {
                export {
                    policy ["policy-bgp-community"]
                }
            }
        }
    }
}
exit all

```

The prefix 10.0.0.0/8 is received on RR-5 with the respective community values and still with the default local preference values:

```

[/]
A:admin@RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match '^Nexthop|^Community|^Pref'
Nexthop      : 192.0.2.1
Local Pref.  : 100                               Interface Name : int-RR-5-PE-1
Community    : 64500:100
Nexthop      : 192.0.2.2
Local Pref.  : 100                               Interface Name : int-RR-5-PE-2
Community    : 64500:50

```

The following policy is configured on RR-5 to apply different local preference values based on the corresponding community value:

```

# On RR-5:
configure {
    policy-options {
        community "64500:10" {
            member "64500:10" { }
        }
        community "64500:100" {
            member "64500:100" { }
        }
        community "64500:50" {
            member "64500:50" { }
        }
    }
    policy-statement "policy-bgp-preference" {
        entry 10 {
            from {
                community {
                    name "64500:100"
                }
            }
            action {
                action-type accept
                local-preference 100
            }
        }
    }
}

```

```

    }
  }
  entry 20 {
    from {
      community {
        name "64500:50"
      }
    }
    action {
      action-type accept
      local-preference 50
    }
  }
  entry 30 {
    from {
      community {
        name "64500:10"
      }
    }
    action {
      action-type accept
      local-preference 10
    }
  }
}
exit all

```

The policy is applied on RR-5:

```

# On RR-5:
configure {
  router "Base" {
    bgp {
      group "iBGP" {
        import {
          policy ["policy-bgp-preference"]
        }
      }
    }
  }
}
exit all

```

The following command output shows that the correct local preference values are applied on the routes received from PE-1 and PE-2:

```

[/]
A:admin@RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match '^Nexthop|^Community|^Pref'
Nexthop      : 192.0.2.1
Local Pref.  : 100                               Interface Name : int-RR-5-PE-1
Community    : 64500:100
Nexthop      : 192.0.2.2
Local Pref.  : 50                               Interface Name : int-RR-5-PE-2
Community    : 64500:50
TieBreakReason : LocalPref

```

RR-5 advertises the route with local preference of 100 to PE-3, with next hop PE-1:

```

[/]
A:admin@PE-3# show router bgp routes 10.0.0.0/8 hunt brief | match '^Nexthop|^Community|^Pref'
Nexthop      : 192.0.2.1
Local Pref.  : 100                               Interface Name : int-PE-3-PE-1
Community    : 64500:100

```

The first loopback interface is shutdown on PE-1, which results in the withdrawal of prefix 172.31.1.0/24 from the route table on PE-1:

```
# On PE-1:
configure {
  router "Base" {
    interface "int-loopback-1" {
      admin-state disable
    }
  }
  exit all
}
```

PE-1 still advertises the prefix 10.0.0.0/8 with the community 64500:100:

```
[/]
A:admin@RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match '^NextHop|^Community|^Pref'
NextHop      : 192.0.2.1
Local Pref.  : 100                               Interface Name : int-RR-5-PE-1
Community    : 64500:100
NextHop      : 192.0.2.2
Local Pref.  : 50                               Interface Name : int-RR-5-PE-2
Community    : 64500:50
TieBreakReason : LocalPref
```

The second loopback interface is shutdown on PE-1, which results in the withdrawal of prefix 172.31.2.0/24 from the route on PE-1:

```
# On PE-1:
configure {
  router "Base" {
    interface "int-loopback-2" {
      admin-state disable
    }
  }
  exit all
}
```

PE-1 still advertises the prefix 10.0.0.0/8 with the community 64500:100:

```
[/]
A:admin@RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match '^NextHop|^Community|^Pref'
NextHop      : 192.0.2.1
Local Pref.  : 100                               Interface Name : int-RR-5-PE-1
Community    : 64500:100
NextHop      : 192.0.2.2
Local Pref.  : 50                               Interface Name : int-RR-5-PE-2
Community    : 64500:50
TieBreakReason : LocalPref
```

The third and the last loopback interface is shutdown on PE-1, which results in the withdrawal of prefix 172.31.3.0/24 from the route table on PE-1:

```
# On PE-1:
configure {
  router "Base" {
    interface "int-loopback-3" {
      admin-state disable
    }
  }
  exit all
}
```

PE-1 now starts advertising the prefix 10.0.0.0/8 with the community 64500:10 and RR-5 applies local preference 10 for this route:

```
[/]
A:admin@RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match '^NextHop|^Community|^Pref'
```

```

NextHop      : 192.0.2.2
Local Pref.  : 50                               Interface Name : int-RR-5-PE-2
Community    : 64500:50
NextHop      : 192.0.2.1
Local Pref.  : 10                               Interface Name : int-RR-5-PE-1
Community    : 64500:10
TieBreakReason : LocalPref
    
```

RR-5 advertises prefix 10.0.0.0/8 to PE-3 with the next-hop address of PE-2:

```

[/]
A:admin@PE-3# show router bgp routes 10.0.0.0/8 hunt brief | match '^NextHop|^Community|^Pref'
NextHop      : 192.0.2.2
Local Pref.  : 50                               Interface Name : int-PE-3-PE-2
Community    : 64500:50
    
```

BGP Conditional Route Advertisement Using "all" Prefix List Match

The loopback interfaces on PE-1 are re-enabled:

```

# On PE-1:
configure {
  router "Base" {
    interface "int-loopback-1" {
      admin-state enable
    }
    interface "int-loopback-2" {
      admin-state enable
    }
    interface "int-loopback-3" {
      admin-state enable
    }
  }
  exit all
    
```

The policy on PE-1 is changed so that the prefix 10.0.0.0/8 is advertised with community 64500:100 only if all the prefixes in the prefix list are active:

```

# On PE-1:
configure {
  policy-options {
    policy-statement "policy-bgp-community" {
      entry 10 {
        conditional-expression {
          route-exists "[prefix-conditional-routes] all"
        }
      }
    }
  }
  exit all
    
```

The first loopback interface is shutdown on PE-1, which results in the withdrawal of prefix 172.31.1.0/24 from the route table on PE-1:

```

# On PE-1:
configure {
  router "Base" {
    interface "int-loopback-1" {
      admin-state disable
    }
  }
  exit all
    
```

PE-1 now advertises the prefix 10.0.0.0/8 with the community 64500:10:

```
[/]
A:admin@RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match '^Nexthop|^Community|Pref'
Nexthop      : 192.0.2.2
Local Pref.  : 50                               Interface Name : int-RR-5-PE-2
Community    : 64500:50
Nexthop      : 192.0.2.1
Local Pref.  : 10                               Interface Name : int-RR-5-PE-1
Community    : 64500:10
TieBreakReason : LocalPref
```

RR-5 advertises prefix 10.0.0.0/8 to PE-3 with the next-hop address of PE-2:

```
[/]
A:admin@PE-3# show router bgp routes 10.0.0.0/8 hunt brief | match '^Nexthop|^Community|Pref'
Nexthop      : 192.0.2.2
Local Pref.  : 50                               Interface Name : int-PE-3-PE-2
Community    : 64500:50
```

BGP Conditional Route Advertisement Using "none" Prefix List Match

The loopback interfaces on PE-1 are re-enabled:

```
# On PE-1:
configure {
  router "Base" {
    interface "int-loopback-1" {
      admin-state enable
    }
    interface "int-loopback-2" {
      admin-state enable
    }
    interface "int-loopback-3" {
      admin-state enable
    }
  }
  exit all
```

The policy on PE-1 is changed so that the prefix 10.0.0.0/8 is advertised with community 64500:100 only if none of the prefixes in the prefix list are active:

```
# On PE-1:
configure {
  policy-options {
    policy-statement "policy-bgp-community" {
      entry 10 {
        conditional-expression {
          route-exists "[prefix-conditional-routes] none"
        }
      }
    }
  }
  exit all
```

PE-1 advertises the prefix 10.0.0.0/8 with the community 64500:10, because all loopback interface prefixes are active:

```
[/]
A:admin@RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match '^Nexthop|^Community|Pref'
Nexthop      : 192.0.2.2
Local Pref.  : 50                               Interface Name : int-RR-5-PE-2
Community    : 64500:50
```

```

NextHop      : 192.0.2.1
Local Pref.  : 10                Interface Name : int-RR-5-PE-1
Community    : 64500:10
TieBreakReason : LocalPref
    
```

The loopback interfaces are shut down one by one or together using a range with the following command on PE-1:

```

# On PE-1:
configure {
  router "Base" {
    interface "int-loopback-1" {
      admin-state disable
    }
    interface "int-loopback-2" {
      admin-state disable
    }
    interface "int-loopback-3" {
      admin-state disable
    }
  }
exit all
    
```

PE-1 now advertises the prefix 10.0.0.0/8 with the community 64500:100:

```

[/]
A:admin@RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match '^NextHop|^Community|^Pref'
NextHop      : 192.0.2.1
Local Pref.  : 100                Interface Name : int-RR-5-PE-1
Community    : 64500:100
NextHop      : 192.0.2.2
Local Pref.  : 50                 Interface Name : int-RR-5-PE-2
Community    : 64500:50
TieBreakReason : LocalPref
    
```

RR-5 advertises prefix 10.0.0.0/8 to PE-3 with the next-hop address of PE-1:

```

[/]
A:admin@PE-3# show router bgp routes 10.0.0.0/8 hunt brief | match '^NextHop|^Community|^Pref'
NextHop      : 192.0.2.1
Local Pref.  : 100                Interface Name : int-PE-3-PE-1
Community    : 64500:100
    
```

Conclusion

BGP conditional route advertisement allows the control of BGP updates based on routes in the route table. A conditional policy entry can be created that tests whether any, all, or none of the prefixes in a prefix list are active and executes the related policy actions.

BGP Convergence — Delayed Route Advertisement

This chapter describes BGP Convergence — Delayed Route Advertisement.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information and MD-CLI configuration in this chapter are based on SR OS Release 20.7.R2. BGP Delayed Route Advertisement is supported in SR OS Release 19.7.R1 and later.

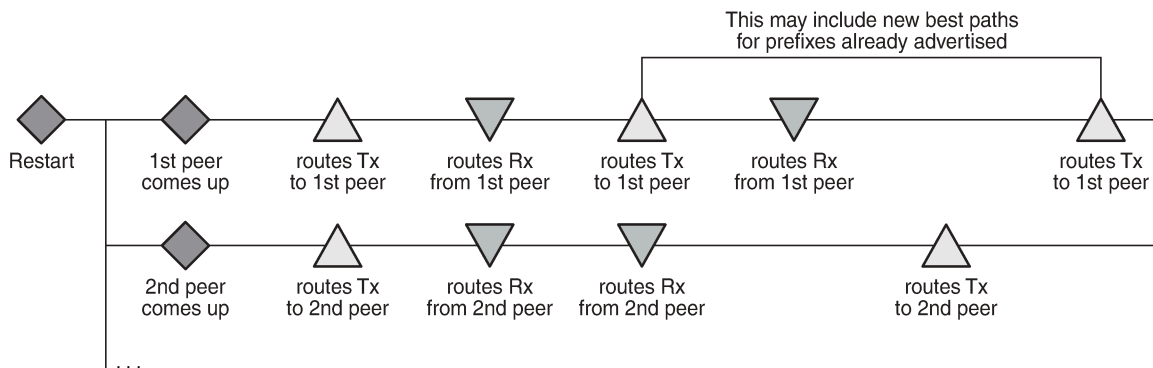
Overview

When the BGP process on a router is starting up or restarting, BGP convergence is finished after the restarting router completes the following actions:

- Reestablish the sessions with configured and discovered BGP neighbors.
- Relearn BGP routes advertised by the direct BGP neighbors (their best paths plus potentially some additional paths).
- Advertise to its direct neighbors the locally originated BGP routes plus the received routes from its set of best paths.

By default, the preceding steps are executed in parallel. After the first BGP session is reestablished, the restarting router starts advertising its own best paths to the BGP neighbor, even though it is still learning BGP routes and rebuilding its RIB-IN database. When more BGP sessions come up and more routes are learned, it is possible that routes previously considered best are no longer best, leading to multiple route advertisements for the same prefix, as shown in [Figure 63: Default SR OS behavior when the BGP process restarts](#).

Figure 63: Default SR OS behavior when the BGP process restarts

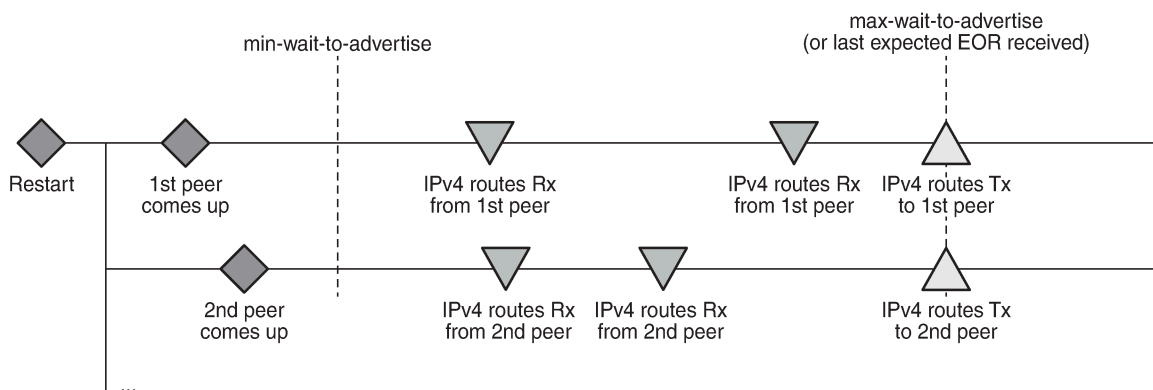


36510

Multiple route advertisements increase the processing workload on the restarting router and on its BGP neighbors. This lengthens the overall convergence time and it can cause short-term inefficiencies in traffic forwarding.

The BGP delayed route advertisement feature provides the following two convergence timers to offer the operator more control on the BGP convergence process when BGP is starting up or restarting: **min-wait-to-advertise** and **max-wait-to-advertise**. This feature applies to IPv4 unicast and IPv6 unicast routes of the base router BGP instance and VPRN BGP instances. BGP convergence tuning allows different timers in the base router and the VPRNs. Also, the **max-wait-to-advertise** timer can be different for IPv4 and IPv6 address families. Figure 64: BGP convergence tuning with delayed route advertisement shows the BGP convergence tuning.

Figure 64: BGP convergence tuning with delayed route advertisement



36511

When a BGP peer has advertised all its routes for a specific address family, it sends an End of RIB (EOR) marker for each address family; for example, peer 192.0.2.4 sent the following EOR for IPv4:

```
171 2020/10/06 13:53:07.312 UTC MINOR: DEBUG #2001 Base Peer 1: 192.0.2.4
"Peer 1: 192.0.2.4: UPDATE
Peer 1: 192.0.2.4 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 0
  End-of-Rib marker (IPv4)
```


"

The restarting node will never advertise routes before the **min-wait-to-advertise** timer has expired. In [Figure 64: BGP convergence tuning with delayed route advertisement](#), no routes were received at that time, but it is possible. Each peer advertises its routes followed by an EOR message per address family. When the restarting node receives all the expected EOR messages (and after the **min-wait-to-advertise** timer expires), it starts advertising its best routes. However, if the **max-wait-to-advertise** timer for the address family expires before all expected EORs have been received, it also starts advertising its best routes.

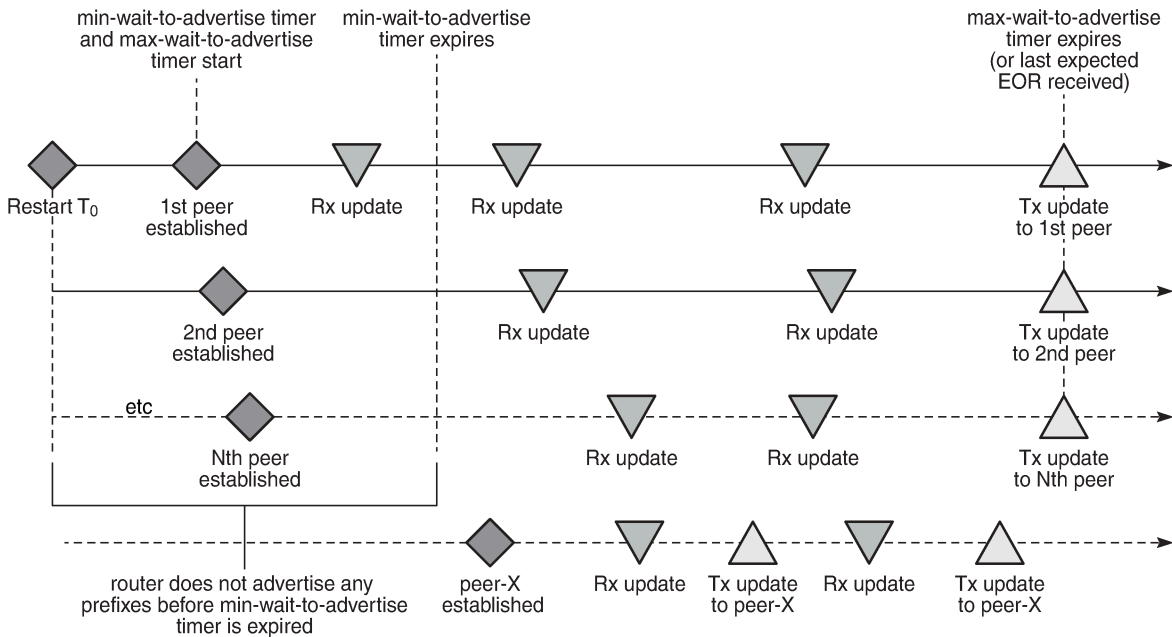


Note:

The timer values must be chosen well, because it is possible that the convergence degrades instead of improves with unsuitable timer values. The timer values depend on the BGP topology (number of peers, number of prefixes per peer, and BGP activeness of the peers). Timer values can be optimized by trial and error, and may have to be reviewed in case of network changes.

[Figure 65: BGP convergence timers](#) shows that the **min-wait-to-advertise** timer starts when the BGP process starts up or restarts, whereas the **max-wait-to-advertise** timer starts when the first peer (dynamic or configured) is established. It also shows that BGP convergence tuning does not apply to a new peer (peer-X) that is established after the **min-wait-to-advertise** timer has expired.

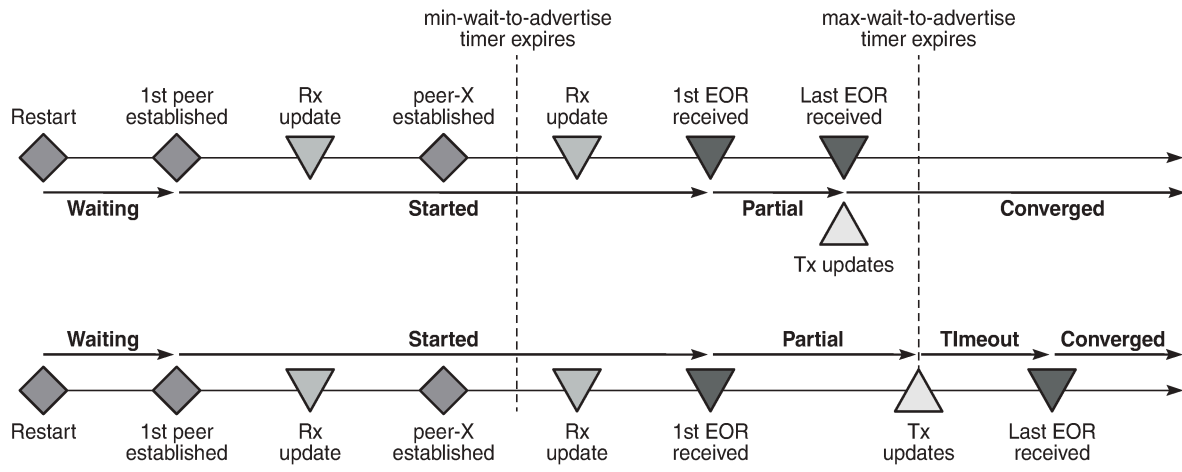
Figure 65: BGP convergence timers



36512

The BGP convergence process can be monitored with the **show router bgp convergence** command. [Figure 66: BGP convergence states](#) shows the different BGP convergence states.

Figure 66: BGP convergence states



36513

The BGP convergence states are:

- **Waiting:** when BGP convergence timers are configured and no peer has reconnected yet.
- **Started:** when the first peer (dynamic or configured) is established.
- **Partial:** when the first EOR is received from a neighbor for a specific address family.
- **Converged:** when the last EOR for an address family is received. If that occurs before the max-time-to-advertise timer expires, the restarting node starts advertising its RIB-OUT.
- **Timeout:** when the max-wait-to-advertise timer expires before the last EOR for an address family is received. The restarting node advertises its RIB-OUT when the timer expires.

When the feature is implemented, BGP maintains information about the convergence process associated with the last startup.

Configuration

The following example shows the principles of SR OS BGP convergence, whereas real-life examples have much larger numbers of BGP sessions and routes. [Figure 67: Example topology](#) shows the example topology with one node in Autonomous System (AS) 64501 and three nodes in AS 64500. On all four nodes, VPRN 1 is configured in AS 64496.

Figure 67: Example topology

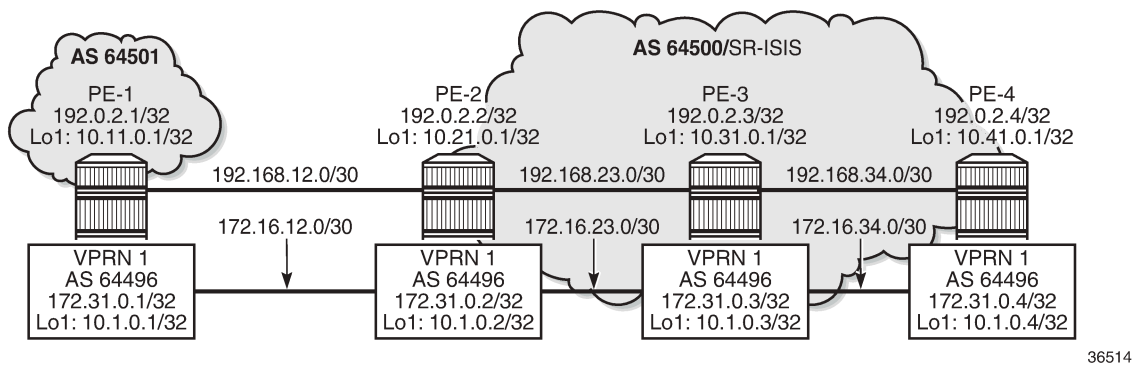


Figure 67: Example topology only shows the IPv4 addresses, but all interfaces also have IPv6 addresses.

The initial configuration on the nodes includes:

- Cards, MDAs, ports
- Router interfaces, with IPv4 and IPv6 addresses
- SR-ISIS in the base router on the three nodes in AS 64500
- IS-IS in VPRN 1 on all four nodes in AS 64496

In the base router, an eBGP session is established between PE-1 in AS 64501 and PE-2 in AS 64500. For the iBGP sessions in AS 64500, PE-2 acts as a Route Reflector (RR). The BGP configuration in the base router on PE-2 is as follows:

```
# on PE-2:
configure {
  router "Base" {
    autonomous-system 64500
    bgp {
      split-horizon true
      group "eBGP" {
        peer-as 64501
        local-address "int-PE-2-PE-1"
        local-as {
          as-number 64500
        }
        import {
          policy ["1:1"]
        }
      }
      group "iBGP-IPv4" {
        peer-as 64500
        family {
          ipv4 true
        }
        cluster {
          cluster-id 192.0.2.2
        }
      }
      group "iBGP-IPv6" {
        peer-as 64500
        family {
          ipv6 true
        }
      }
    }
  }
}
```

```

        cluster {
            cluster-id 192.0.2.2
        }
    }
    neighbor "192.0.2.3" {
        group "iBGP-IPv4"
        next-hop-self true
        export {
            policy ["export-10.21"]
        }
    }
    neighbor "192.0.2.4" {
        group "iBGP-IPv4"
        next-hop-self true
        export {
            policy ["export-10.21"]
        }
    }
    neighbor "192.168.12.1" {
        group "eBGP"
        next-hop-self true
        family {
            ipv4 true
        }
        export {
            policy ["export-10.21" "1:1"]
        }
    }
    neighbor "2001:db8::2:3" {
        group "iBGP-IPv6"
        next-hop-self true
        export {
            policy ["export-10:21"]
        }
    }
    neighbor "2001:db8::2:4" {
        group "iBGP-IPv6"
        next-hop-self true
        export {
            policy ["export-10:21"]
        }
    }
    neighbor "2001:db8::12:1" {
        group "eBGP"
        next-hop-self true
        family {
            ipv6 true
        }
        export {
            policy ["export-10:21" "1:1"]
        }
    }
}
}

```

The policies are the following:

```

# on PE-2:
configure {
    policy-options {
        community "1:1" {
            member "1:1" { }
        }
    }
    prefix-list "10.21.0.0/16" {

```



```
    }
  }
  neighbor "172.31.0.1" {
    group "iBGP-VPRN1"
    local-address 172.31.0.2
    family {
      ipv4 true
    }
    export {
      policy ["export-10.1"]
    }
  }
  neighbor "172.31.0.3" {
    group "iBGP-VPRN1"
    local-address 172.31.0.2
    family {
      ipv4 true
    }
    export {
      policy ["export-10.1"]
    }
  }
  neighbor "172.31.0.4" {
    group "iBGP-VPRN1"
    local-address 172.31.0.2
    family {
      ipv4 true
    }
    export {
      policy ["export-10.1"]
    }
  }
  neighbor "2001:db8::31:0:1" {
    group "iBGP-VPRN1"
    family {
      ipv6 true
    }
    export {
      policy ["export-10:1"]
    }
  }
  neighbor "2001:db8::31:0:3" {
    group "iBGP-VPRN1"
    family {
      ipv6 true
    }
    export {
      policy ["export-10:1"]
    }
  }
  neighbor "2001:db8::31:0:4" {
    group "iBGP-VPRN1"
    family {
      ipv6 true
    }
    export {
      policy ["export-10:1"]
    }
  }
}
```

The configuration is similar on the other nodes.

The following BGP summary on PE-2 shows the different sessions where PE-2 receives one IPv4 or IPv6 route per neighbor and advertises three IPv4 or IPv6 routes per neighbor, both in the base router (Def. Instance) and in VPRN 1 (Svc: 1):

```
[ ]
A:admin@PE-2# show router bgp summary all
=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId          AS PktRcvd InQ Up/Down  State|Rcv/Act/Sent (Addr Family)
                   PktSent OutQ
-----
192.0.2.3
Def. Instance 64500      8  0 00h01m56s 1/1/3 (IPv4)
              11  0
192.0.2.4
Def. Instance 64500      8  0 00h01m43s 1/1/3 (IPv4)
              11  0
192.168.12.1
Def. Instance 64501      9  0 00h02m20s 1/1/3 (IPv4)
              11  0
2001:db8::2:3
Def. Instance 64500      8  0 00h01m56s 1/1/3 (IPv6)
              11  0
2001:db8::2:4
Def. Instance 64500      8  0 00h01m43s 1/1/3 (IPv6)
              11  0
2001:db8::12:1
Def. Instance 64501      9  0 00h02m14s 1/1/3 (IPv6)
              11  0
172.31.0.1
Svc: 1        64496      9  0 00h02m01s 1/1/3 (IPv4)
              12  0
172.31.0.3
Svc: 1        64496      9  0 00h02m02s 1/1/3 (IPv4)
              12  0
172.31.0.4
Svc: 1        64496      8  0 00h01m50s 1/1/3 (IPv4)
              10  0
2001:db8::31:0:1
Svc: 1        64496      8  0 00h01m50s 1/1/3 (IPv6)
              10  0
2001:db8::31:0:3
Svc: 1        64496      8  0 00h01m50s 1/1/3 (IPv6)
              10  0
2001:db8::31:0:4
Svc: 1        64496      8  0 00h01m50s 1/1/3 (IPv6)
              10  0
-----
```

By default, BGP does not delay route advertisement. The following **show** command on PE-2 shows that no **min-wait-to-advertise** timer and no **max-wait-to-advertise** timer is configured (the default value is 0). The number of established peers is 3 for IPv4 and IPv6 in the base router.

```
[ ]
A:admin@PE-2# show router bgp convergence
=====
```

```

BGP IPv4 Convergence
=====
Min wait advertise timer           : 0
Established peers at min wait timer expiry : N/A
Current established peers          : 3
First session established time     : N/A
Last session established time      : N/A
Max Wait advertise timer          : 0
Converged peers                   : N/A
Converged state                   : N/A
Converged time                    : N/A
=====

BGP IPv6 Convergence
=====
Min wait advertise timer           : 0
Established peers at min wait timer expiry : N/A
Current established peers          : 3
First session established time     : N/A
Last session established time      : N/A
Max Wait advertise timer          : 0
Converged peers                   : N/A
Converged state                   : N/A
Converged time                    : N/A
=====

```

A similar command can be launched for VPRN 1: **show router 1 bgp convergence**. The output is similar, but not shown here.

On PE-2, BGP delayed route advertisement is configured with **min-wait-to-advertise** equal to 20 seconds in the base router and **min-wait-to-advertise** equal to 60 seconds in VPRN 1. For both cases, the **max-wait-to-advertise** is three times as long as the **min-wait-to-advertise**, but it is possible to have different **max-wait-to-advertise** timers for IPv4 and IPv6.

In this example, BGP delayed route advertisement is only configured on PE-2, while the other nodes keep advertising their routes immediately after the BGP session is reestablished. PE-2 will accept these routes, but it will only advertise them after receiving the last expected EOR for IPv4 or IPv6 (for the base router or VPRN 1) and **min-wait-to-advertise** timer expires. If the **max-wait-to-advertise** timer expires before the last expected EOR is received for IPv4 or IPv6, PE-2 will start advertising the received routes.

```

# on PE-2:
configure {
  router "Base" {
    bgp {
      convergence {
        min-wait-to-advertise 20
      }
      family ipv4 {
        max-wait-to-advertise 30
      }
      family ipv6 {
        max-wait-to-advertise 30
      }
    }
  }
}
service {
  vprn "VPRN 1" {
    bgp {
      convergence {
        min-wait-to-advertise 60
      }
      family ipv4 {

```



```

Min wait advertise timer           : 20
Established peers at min wait timer expiry : 0
Current established peers         : 3
First session established time     : 00h00m01s
Last session established time     : 00h00m01s
Max Wait advertise timer         : 30
Converged peers                   : 0
Converged state                 : started
Converged time                    : N/A
=====

=====
BGP IPv6 Convergence
=====
Min wait advertise timer           : 20
Established peers at min wait timer expiry : 0
Current established peers         : 3
First session established time     : 00h00m01s
Last session established time     : 00h00m01s
Max Wait advertise timer         : 30
Converged peers                   : 0
Converged state                 : started
Converged time                    : N/A
=====

```

The **show router 1 bgp convergence** command shows a similar output for VPRN 1, but is not shown here.

After a few seconds, PE-2 receives IPv4 and IPv6 routes from PE-3 and PE-4, both in the base router and VPRN 1, as follows:

```

[]
A:admin@PE-2# show router bgp summary all

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====

Neighbor
Description
ServiceId      AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
              PktSent OutQ
-----
192.0.2.3
Def. Instance  64500      5   0 00h00m01s 1/1/0 (IPv4)
              4   0
192.0.2.4
Def. Instance  64500      5   0 00h00m01s 1/1/0 (IPv4)
              4   0
192.168.12.1
Def. Instance  64501      3   0 00h00m00s 0/0/0 (IPv4)
              4   0
2001:db8::2:3
Def. Instance  64500      5   0 00h00m01s 1/1/0 (IPv6)
              4   0
2001:db8::2:4
Def. Instance  64500      5   0 00h00m01s 1/1/0 (IPv6)
              4   0
2001:db8::12:1
Def. Instance  64501      3   0 00h00m00s 0/0/0 (IPv6)
              4   0
172.31.0.1

```

```

Svc: 1          64496      3    0 00h00m00s 0/0/0 (IPv4)
                3    0
172.31.0.3
Svc: 1          64496      5    0 00h00m01s 1/1/0 (IPv4)
                4    0
172.31.0.4
Svc: 1          64496      5    0 00h00m01s 1/1/0 (IPv4)
                4    0
2001:db8::31:0:1
Svc: 1          64496      3    0 00h00m00s 0/0/0 (IPv6)
                3    0
2001:db8::31:0:3
Svc: 1          64496      5    0 00h00m01s 1/1/0 (IPv6)
                4    0
2001:db8::31:0:4
Svc: 1          64496      5    0 00h00m01s 1/1/0 (IPv6)
                4    0
-----

```

PE-2 accepts the received routes, but does not advertise the routes because the **min-wait-to-advertise** timer has not expired yet, and PE-2 only received IPv4 and IPv6 routes and EORs from PE-3 and PE-4, not from PE-1, so the converged state is "partial", as follows:

```

[]
A:admin@PE-2# show router bgp convergence
=====
BGP IPv4 Convergence
=====
Min wait advertise timer           : 20
Established peers at min wait timer expiry : 0
Current established peers          : 3
First session established time      : 00h00m01s
Last session established time       : 00h00m02s
Max Wait advertise timer           : 30
Converged peers                    : 2
Converged state                   : partial
Converged time                      : N/A
=====

BGP IPv6 Convergence
=====
Min wait advertise timer           : 20
Established peers at min wait timer expiry : 0
Current established peers          : 3
First session established time      : 00h00m01s
Last session established time       : 00h00m02s
Max Wait advertise timer           : 30
Converged peers                    : 2
Converged state                   : partial
Converged time                      : N/A
=====

```

The **show router 1 bgp convergence** command shows a similar output for VPRN 1, but is not shown here.

After a few seconds, all IPv4 and IPv6 routes have been received in the base router. PE-2 has received an EOR message from each neighbor in the base router. The following BGP summary shows that PE-2 has

received and advertised all IPv4 and IPv6 routes in the base router, whereas it only received IPv4 and IPv6 routes from two neighbors in VPRN 1, not yet from PE-1:

```
[ ]
A:admin@PE-2# show router bgp summary all

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId          AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
                PktSent OutQ
-----
192.0.2.3
Def. Instance 64500      5   0 00h00m12s 1/1/3 (IPv4)
                7   0
192.0.2.4
Def. Instance 64500      5   0 00h00m11s 1/1/3 (IPv4)
                7   0
192.168.12.1
Def. Instance 64501      5   0 00h00m14s 1/1/3 (IPv4)
                5   0
2001:db8::2:3
Def. Instance 64500      5   0 00h00m12s 1/1/3 (IPv6)
                7   0
2001:db8::2:4
Def. Instance 64500      5   0 00h00m11s 1/1/3 (IPv6)
                7   0
2001:db8::12:1
Def. Instance 64501      5   0 00h00m13s 1/1/3 (IPv6)
                5   0
172.31.0.1
Svc: 1         64496      4   0 00h00m13s 0/0/0 (IPv4)
                4   0
172.31.0.3
Svc: 1         64496      5   0 00h00m12s 1/1/0 (IPv4)
                4   0
172.31.0.4
Svc: 1         64496      5   0 00h00m11s 1/1/0 (IPv4)
                4   0
2001:db8::31:0:1
Svc: 1         64496      4   0 00h00m13s 0/0/0 (IPv6)
                4   0
2001:db8::31:0:3
Svc: 1         64496      5   0 00h00m12s 1/1/0 (IPv6)
                4   0
2001:db8::31:0:4
Svc: 1         64496      5   0 00h00m11s 1/1/0 (IPv6)
                4   0
-----
```

As a result of this, BGP is in the "converged" state in the base router, both for IPv4 and IPv6, as follows:

```
[ ]
A:admin@PE-2# show router bgp convergence

=====
BGP IPv4 Convergence
=====
Min wait advertise timer          : 20
```

```

Established peers at min wait timer expiry : 3
Current established peers                  : 3
First session established time             : 00h00m01s
Last session established time              : 00h00m03s
Max Wait advertise timer                  : 30
Converged peers                           : 3
Converged state                          : converged
Converged time                            : 00h00m20s
=====

```

=====

BGP IPv6 Convergence

=====

```

Min wait advertise timer                  : 20
Established peers at min wait timer expiry : 3
Current established peers                  : 3
First session established time             : 00h00m01s
Last session established time              : 00h00m03s
Max Wait advertise timer                  : 30
Converged peers                           : 3
Converged state                          : converged
Converged time                            : 00h00m20s
=====

```

The converged time is only applicable in the "converged" state and is measured relative to BGP instance restart at time T=0.

BGP is still in the "partial" state within the VPRN 1 context, both for IPv4 and IPv6, as follows:

```

[]
A:admin@PE-2# show router 1 bgp convergence
=====
BGP IPv4 Convergence
=====
Min wait advertise timer                  : 60
Established peers at min wait timer expiry : 0
Current established peers                  : 3
First session established time             : 00h00m01s
Last session established time              : 00h00m03s
Max Wait advertise timer                  : 180
Converged peers                           : 3
Converged state                          : partial
Converged time                            : N/A
=====

```

=====

BGP IPv6 Convergence

=====

```

Min wait advertise timer                  : 60
Established peers at min wait timer expiry : 0
Current established peers                  : 3
First session established time             : 00h00m01s
Last session established time              : 00h00m03s
Max Wait advertise timer                  : 180
Converged peers                           : 3
Converged state                          : partial
Converged time                            : N/A
=====

```

After a while, PE-1 also advertises its routes for VPRN 1, followed by EORs for IPv4 and IPv6. BGP converges for VPRN 1, as follows:

```
[ ]
A:admin@PE-2# show router 1 bgp convergence

=====
BGP IPv4 Convergence
=====
Min wait advertise timer           : 60
Established peers at min wait timer expiry : 3
Current established peers          : 3
First session established time      : 00h00m01s
Last session established time       : 00h00m03s
Max Wait advertise timer           : 180
Converged peers                    : 3
Converged state                  : converged
Converged time                      : 00h01m00s
=====

=====
BGP IPv6 Convergence
=====
Min wait advertise timer           : 60
Established peers at min wait timer expiry : 3
Current established peers          : 3
First session established time      : 00h00m01s
Last session established time       : 00h00m03s
Max Wait advertise timer           : 180
Converged peers                    : 3
Converged state                  : converged
Converged time                      : 00h01m00s
=====
```

Conclusion

With BGP convergence tuning (by means of delaying route advertisements using two timers), less path churn and fewer advertisements can result in faster convergence. BGP convergence is mainly important in scaled environments (high number of BGP sessions and routes). As a result, the advertised paths are more optimal. The BGP convergence process can be monitored using a **show** command.

BGP Default Route Origination

This chapter describes BGP Default Route Origination.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information and MD-CLI configuration in this chapter are based on SR OS Release 20.7.R1. Advertising artificially generated IPv4 and IPv6 default routes is supported in SR OS Release 19.7.R1 and later.

Overview

It is common practice for a BGP router to send an IPv4 and/or IPv6 default route to certain peers rather than a number of more specific routes.

In SR OS releases earlier than 19.7.R1, a BGP router only advertises a default route that is installed in the Forwarding Information Base (FIB). This default route is either received from a BGP peer and re-advertised, or the default route is configured locally as a static route, with black-hole next-hop. The attributes of this default route can be modified by an export policy. The drawback of depending on a default route installed in the FIB is that when the BGP peer withdraws or modifies the default route, the BGP router must withdraw or re-advertise the default route.

In SR OS Release 19.7.R1 and later, the **send-default** command allows BGP routers to advertise artificially generated IPv4 (0.0.0.0/0) and/or IPv6 (::/0) default routes. These artificially generated default routes are unrelated to possible default routes installed in the FIB of the local router. If the local FIB contains a default route and a BGP export policy allows that installed default route to be advertised, the **send-default** command overrides the advertisement of the installed default route. If the default route in the FIB is withdrawn or modified, the artificially generated default route continues to be advertised.

The **send-default** command can be configured in the general BGP context, in the BGP group context, or in the BGP neighbor context, in both base router instance and VPRN router instances. The command can be used for IPv4, IPv6, or both. An optional send-default export policy can modify the attributes of the artificially generated default routes. Only the **default-action** part of this send-default export policy is parsed and applied, as follows:

```
*[ex:configure router "Base" bgp]
A:admin@PE-1# send-default

send-default

export-policy      - Export policy name
ipv4               - Enable IPv4 family type
ipv6               - Enable IPv6 family type
```

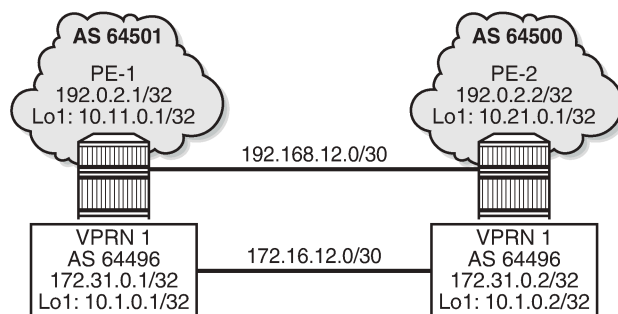
Before modification by a send-default export policy, the properties of the artificially generated default route are as follows:

- The origin is set to Incomplete.
- When advertised to an iBGP peer, the AS_PATH is empty.
- When advertised to an eBGP peer, the global Autonomous System Number (ASN) and/or local AS are prepended. If the send-default export policy specifies an **as-path-prepend** action, these modifications are made before prepending the ASN and/or local AS.
- The BGP next-hop is the local address used with the receiving peer or the local router ID (if the Network Layer Reachability Information (NLRI) is IPv6, and the local address is an IPv4 address or it refers to an IPv4-only interface).
- No Multi-Exit Discriminator (MED) attribute is added.
- When advertised to an iBGP peer, a local preference attribute is added and its value is taken from the configuration of the **local-preference** command or the value 100, the implicit default.
- No standard or large communities are attached. When a send-default export policy is applied to change this, confirm that **disable-communities** is not set.

Configuration

[Figure 68: Example topology with IPv4 addresses](#) shows the example topology with two routers. An eBGP session is established between the base routers (PE-1 in AS 64501 and PE-2 in AS 64500) and an iBGP session is established within VPRN 1 in AS 64496.

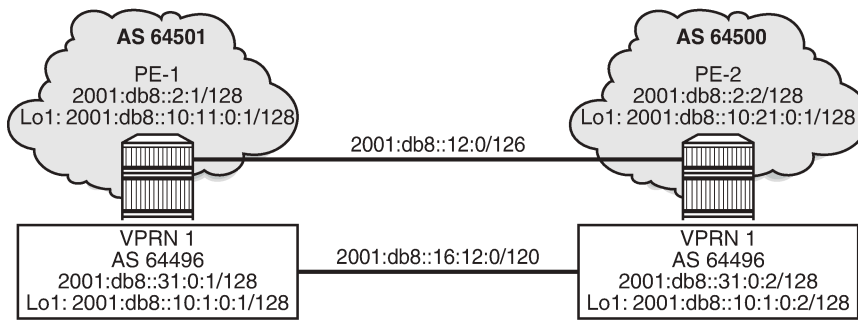
Figure 68: Example topology with IPv4 addresses



36515

[Figure 69: Example topology with IPv6 addresses](#) shows the same example topology with IPv6 addresses.

Figure 69: Example topology with IPv6 addresses



36516

The initial configuration includes:

- Cards, MDAs, ports
- Router interfaces

On PE-1, the BGP configuration in the base router is as follows:

```
# on PE-1:
configure {
  router "Base" {
    autonomous-system 64501
    bgp {
      router-id 192.0.2.1
      split-horizon true
      group "eBGP" {
        peer-as 64500
        local-as {
          as-number 64501
        }
        import {
          policy ["1:0"]          # accepts routes with community "1:0"
        }
      }
      neighbor "192.168.12.2" {
        group "eBGP"
        local-address "int-PE-1-PE-2"
        send-communities {
          large false           # no large communities sent to 192.168.12.2
        }
        family {
          ipv4 true
        }
      }
      neighbor "2001:db8::12:2" {
        group "eBGP"
        local-address 2001:db8::12:1
        family {
          ipv6 true
        }
      }
    }
  }
}
```

On PE-1, the BGP configuration in VPRN 1 is as follows:

```
# on PE-1:
configure {
  service {
    vprn "VPRN 1" {
      autonomous-system 64496
      ---snip---
      bgp
        router-id 172.31.0.1
        split-horizon true
        group "iBGP-VPRN1" {
          type internal
        }
        neighbor "172.31.0.2" {
          group "iBGP-VPRN1"
          local-address 172.31.0.1
          send-communities {
            large false      # no large communities sent to 172.31.0.2
          }
          family {
            ipv4 true
          }
        }
        neighbor "2001:db8::31:0:2" {
          group "iBGP-VPRN1"
          family {
            ipv6 true
          }
        }
      }
    }
  }
  ---snip---
}
```

The configuration is similar on PE-2.

No export policies are applied in BGP, so no routes will be advertised. The following BGP sessions are established on PE-2:

```
[ ]
A:admin@PE-2# show router bgp summary all

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId      AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
                PktSent OutQ
-----
192.168.12.1
Def. Instance  64501      8   0 00h01m45s 0/0/0 (IPv4)
                9   0
2001:db8::12:1
Def. Instance  64501      7   0 00h01m39s 0/0/0 (IPv6)
                7   0
172.31.0.1
Svc: 1         64496      7   0 00h01m33s 0/0/0 (IPv4)
                7   0
2001:db8::31:0:1
Svc: 1         64496      6   0 00h01m24s 0/0/0 (IPv6)
```

```

-----
6 0
-----

```

Initially, no default routes are installed in the route table of the base router or the VPRN; for example, on PE-2, as follows:

```

[]
A:admin@PE-2# show router route-table 0.0.0.0/0

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type  Proto  Age      Pref
  Next Hop[Interface Name]                       Metric
-----
No. of Routes: 0

```

```

[]
A:admin@PE-2# show router 1 route-table ipv6 ::/0

=====
IPv6 Route Table (Service: 1)
=====
Dest Prefix[Flags]                               Type  Proto  Age      Pref
  Next Hop[Interface Name]                       Metric
-----
No. of Routes: 0

```

The following use cases are shown in the following subsections:

- [Advertise default routes that are installed in the FIB](#)
- [Advertise artificially generated default routes](#)

Advertise default routes that are installed in the FIB

PE-1 has not received default routes from any other BGP peer, so black-holed default routes for IPv4 and IPv6 are configured locally in the base router and in VPRN 1 routing instances, as follows:

```

# on PE-1:
configure {
  router "Base" {
    static-routes {
      route 0.0.0.0/0 route-type unicast {
        blackhole {
          admin-state enable
        }
      }
      route ::/0 route-type unicast {
        blackhole {
          admin-state enable
        }
      }
    }
  }
  service {
    vprn "VPRN 1" {
      static-routes {

```

```

        route 0.0.0.0/0 route-type unicast {
            blackhole {
                admin-state enable
            }
        }
        route ::/0 route-type unicast {
            blackhole {
                admin-state enable
            }
        }
    }
}

```

The following export policies are configured for prefixes 0.0.0.0/0 and ::/0.

```

# on PE-1:
configure {
    policy-options {
        community "1:0" {
            member "1:0" { }
        }
        prefix-list "route_0/0" {
            prefix 0.0.0.0/0 type exact {
            }
        }
        prefix-list "route_::/0" {
            prefix ::/0 type exact {
            }
        }
    }
    policy-statement "export-route_0/0" {
        entry 10 {
            from {
                prefix-list ["route_0/0"]
            }
            action {
                action-type accept
                community {
                    add ["1:0"]
                }
                origin igp
            }
        }
    }
    policy-statement "export-route_::/0" {
        entry 10 {
            from {
                prefix-list ["route_::/0"]
            }
            action {
                action-type accept
                community {
                    add ["1:0"]
                }
                origin igp
            }
        }
    }
}
}

```

These export policies are applied in BGP group "eBGP" in the base router, as follows:

```

# on PE-1:
configure {
    router "Base" {
        bgp {

```

```

        group "eBGP" {
            export {
                policy ["export-route_0/0" "export-route_::/0"]
            }
        }
    }
}

```

The same export policies are applied in the general **bgp** context in VPRN 1, as follows:

```

# on PE-1:
configure {
    service {
        vprn "VPRN 1" {
            bgp {
                export {
                    policy ["export-route_0/0" "export-route_::/0"]
                }
            }
        }
    }
}

```

No default routes are configured on PE-2.

The following BGP summary on PE-2 shows that in each BGP session one BGP route is received and active:

```

[]
A:admin@PE-2# show router bgp summary all
=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId          AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
                PktSent OutQ
-----
192.168.12.1
Def. Instance     64501      22   0 00h08m21s 1/1/0 (IPv4)
                22   0
2001:db8::12:1
Def. Instance     64501      21   0 00h08m15s 1/1/0 (IPv6)
                20   0

172.31.0.1
Svc: 1            64496      21   0 00h08m10s 1/1/0 (IPv4)
                20   0
2001:db8::31:0:1
Svc: 1            64496      21   0 00h08m00s 1/1/0 (IPv6)
                20   0
-----

```

The following BGP route is received in the base router:

```

[]
A:admin@PE-2# show router bgp routes
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge

```

```

Origin codes : i - IGP, e - EGP, ? - incomplete

=====
BGP IPv4 Routes
=====
Flag Network LocalPref MED
      Nexthop (Router) Path-Id IGP Cost
      As-Path Label
-----
u*>i 0.0.0.0/0 100 None
      192.168.12.1 None 0
      64501 -
-----
Routes : 1
=====

```

Also, a BGP-IPv6 route for ::/0 is received in the base router, and VPRN 1 receives BGP-IPv4 route 0.0.0.0/0 and BGP-IPv6 route ::/0, as follows:

```

[]
A:admin@PE-2# show router 1 bgp routes ipv6
=====
BGP Router ID:172.31.0.2 AS:64496 Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete

=====
BGP IPv6 Routes
=====
Flag Network LocalPref MED
      Nexthop (Router) Path-Id IGP Cost
      As-Path Label
-----
u*>i ::/0 100 None
      2001:db8::31:0:1 None 10
      No As-Path -
-----
Routes : 1
=====

```

The default route 0.0.0.0/0 is installed in the route table for the base router, as follows:

```

[]
A:admin@PE-2# show router route-table 0.0.0.0/0
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags] Type Proto Age Pref
Next Hop[Interface Name] Metric
-----
0.0.0.0/0 Remote BGP 00h02m48s 170
192.168.12.1 0
-----
No. of Routes: 1

```

Similarly, the default route `::/0` is installed in the IPv6 route table for the base router (not shown here). For VPRN 1, default route `0.0.0.0/0` is installed in the IPv4 route table (not shown here), whereas default route `::/0` is installed in the IPv6, as follows:

```
[ ]
A:admin@PE-2# show router 1 route-table ipv6 ::/0
=====
IPv6 Route Table (Service: 1)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]                Metric
-----
::/0                               Remote BGP      00h02m52s  170
      fe80::21:88ab:d904:706f-"int-VPRN1-PE-2-PE-1"  10
-----
No. of Routes: 1
```

Advertise artificially generated default routes

With the **send-default** command, no default routes need to be installed in the FIB. However, the following example shows that both static default routes in PE-1 remain, but that this static default route will not be advertised anymore. With the **send-default** command, an artificially generated default route is advertised and overrules the static default route.

The following **send-default** command is configured on PE-1 and PE-2:

```
# on PE-1, PE-2:
configure {
  router "Base" {
    bgp {
      group "eBGP" {
        send-default {
          ipv4 true
          ipv6 true
        }
      }
    }
  }
  service {
    vprn "VPRN 1" {
      bgp {
        send-default {
          ipv4 true
          ipv6 true
        }
      }
    }
  }
}
```

The following BGP summary on PE-2 shows that in each iBGP session (VPRN 1 in AS 64496), one route is received and active, and one route is advertised. The BGP sessions in the base router are eBGP sessions. In MD-CLI, the default behavior is compliant with RFC 8212, so all BGP routes are rejected when no export policies are configured.

```
[ ]
A:admin@PE-2# show router bgp summary all
=====
BGP Summary
```

```

=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId          AS PktRcvd InQ Up/Down  State|Rcv/Act/Sent (Addr Family)
                   PktSent OutQ
-----
192.168.12.1
Def. Instance 64501      13   0 00h03m50s 0/0/0 (IPv4)
                   14   0
2001:db8::12:1
Def. Instance 64501      12   0 00h03m44s 0/0/0 (IPv6)
                   12   0
172.31.0.1
Svc: 1        64496      12   0 00h03m40s 1/1/1 (IPv4)
                   13   0
2001:db8::31:0:1
Svc: 1        64496      12   0 00h03m40s 1/1/1 (IPv6)
                   12   0
-----

```

Because no send-default export policy is configured to modify the attributes, the origin will remain Incomplete, which also proves that the received routes in VPRN 1 on PE-2 are different from the ones received before the **send-default** command was configured, as follows:

```

[]
A:admin@PE-2# show router 1 bgp routes
=====
BGP Router ID:172.31.0.2      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                LocalPref  MED
     Nexthop (Router)      Path-Id    IGP Cost
     As-Path                Label
-----
u*>? 0.0.0.0/0                100        None
     172.31.0.1            None        10
     No As-Path              -
-----
Routes : 1
=====

```

The following shows the details of the received and advertised BGP-IPv6 route ::/0 in VPRN 1 on PE-2:

```

[]
A:admin@PE-2# show router 1 bgp routes ::/0 hunt
=====
BGP Router ID:172.31.0.2      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete

```



```

=====
BGP IPv6 Routes
=====
-----
RIB In Entries
-----
Network      : ::/0
Nexthop      : 2001:db8::31:0:1
Path Id      : None
From         : 2001:db8::31:0:1
Res. Protocol : ISIS                      Res. Metric   : 10
Res. Nexthop  : fe80::21:88ab:d904:706f
Local Pref.   : 100                       Interface Name : int-VPRN1-PE-2-PE-1
Aggregator AS : None                       Aggregator    : None
Atomic Aggr.  : Not Atomic                 MED           : None
AIGP Metric   : None                       IGP Cost      : 10
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None                       Peer Router Id : 172.31.0.1
Fwd Class     : None                       Priority       : None
Flags       : Used Valid Best Incomplete
Route Source  : Internal
AS-Path       : No As-Path
Route Tag     : 0
Neighbor-AS   : n/a
Orig Validation: NotFound
Source Class  : 0                          Dest Class    : 0
Add Paths Send : Default
RIB Priority   : Normal
Last Modified  : 00h01m07s
-----
RIB Out Entries
-----
Network      : ::/0
Nexthop      : 2001:db8::31:0:2
Path Id      : None
To           : 2001:db8::31:0:1
Res. Protocol : INVALID                    Res. Metric   : 0
Res. Nexthop  : n/a
Local Pref.   : 100                       Interface Name : NotAvailable
Aggregator AS : None                       Aggregator    : None
Atomic Aggr.  : Not Atomic                 MED           : None
AIGP Metric   : None                       IGP Cost      : 10
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None                       Peer Router Id : 172.31.0.1
Origin     : Incomplete
AS-Path       : No As-Path
Route Tag     : 0
Neighbor-AS   : n/a
Orig Validation: NotFound
Source Class  : 0                          Dest Class    : 0
-----
Routes : 2
=====

```

The origin attribute can be modified by the following export policy that sets the origin to IGP, the MED value to 50, and adds the communities "64496:1:1" and "1:0":

```
# on PE-1, PE-2:
configure {
  policy-options {
    community "1:0" {
      member "1:0" { }
    }
    community "large1" {
      member "64496:1:1" { }
    }
  }
  policy-statement "1:0" { # import policy for eBGP sessions (base)
    entry 10 {
      from {
        community {
          name "1:0"
        }
      }
      action {
        action-type accept
      }
    }
  }
  policy-statement "export-default" { # send-default export policy
    default-action {
      action-type accept
      origin igp
      bgp-med {
        set 50
      }
      community {
        add ["large1" "1:0"]
      }
    }
  }
}
```

The export policy is included in the **send-default** command, as follows:

```
# on PE-1, PE-2:
configure {
  router "Base" {
    bgp {
      group "eBGP" {
        send-default {
          ipv4 true
          ipv6 true
          export-policy "export-default"
        }
        import {
          policy ["1:0"]
        }
      }
    }
  }
  service {
    vprn "VPRN 1" {
      bgp {
        send-default {
          ipv4 true
          ipv6 true
          export-policy "export-default"
        }
      }
    }
  }
}
```

```
}

```

The export policy sets the origin to IGP, sets the MED to a value of 50, and adds communities "large1" and "1:0". The import policy in the base router accepts routes with community "1:0". PE-2 receives and accepts the BGP-IPv4 default route with origin IGP and MED 50, as follows:

```
[ ]
A:admin@PE-2# show router bgp routes
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  0.0.0.0/0                  None       50
      192.168.12.1           None       0
      64501                   -
-----
Routes : 1
=====
```

```
[ ]
A:admin@PE-2# show router bgp routes 0.0.0.0/0 hunt | match Flags
Flags          : Used Valid Best IGP
```

The other artificially generated default routes also have origin IGP and MED 50. In this example, the **send-communities large false** command is configured on PE-1 and PE-2 for the IPv4 neighbors in the base router and in VPRN 1, so no large community is sent for IPv4; only for IPv6. On PE-2, the default IPv4 routes in the RIB-IN and the RIB-OUT of the base router only contain community "1:0", not the large community "64496:1:1", as follows:

```
[ ]
A:admin@PE-2# show router bgp routes 0.0.0.0/0 hunt | match Community
Community      : 1:0          # RIB-IN
Community      : 1:0          # RIB-OUT
```

On PE-2, the details of the received default IPv6 route ::/0 in VPRN 1 are as follows:

```
[ ]
A:admin@PE-2# show router 1 bgp routes ::/0 hunt
=====
BGP Router ID:172.31.0.2    AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv6 Routes
```

```

=====
-----
RIB In Entries
-----
Network      : ::/0
Nexthop      : 2001:db8::31:0:1
Path Id      : None
From         : 2001:db8::31:0:1
Res. Protocol : ISIS                      Res. Metric   : 10
Res. Nexthop  : fe80::10:1ff:fe01:1
Local Pref.   : 100                       Interface Name : int-VPRN1-PE-2-PE-1
Aggregator AS : None                      Aggregator    : None
Atomic Aggr.  : Not Atomic                MED           : 50
AIGP Metric   : None                      IGP Cost      : 10
Connector     : None
Community     : 1:0 64496:1:1
Cluster       : No Cluster Members
Originator Id : None                      Peer Router Id : 172.31.0.1
Fwd Class     : None                      Priority      : None
Flags         : Used Valid Best IGP
Route Source  : Internal
AS-Path       : No As-Path
Route Tag     : 0
Neighbor-AS   : n/a
Orig Validation: NotFound
Source Class  : 0                          Dest Class    : 0
Add Paths Send : Default
RIB Priority   : Normal
Last Modified : 00h02m50s
---snip---

```

The artificially generated default routes are only modified by the send-default export policy, not involving other export BGP policies.

Conclusion

With the **send-default** command, BGP routers can advertise artificially generated default routes for IPv4, IPv6, or both. The artificially generated default routes are always advertised, regardless of the presence of default routes installed in the local FIB.

BGP Fast Reroute

This chapter provides information about BGP Fast Reroute.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 14.0.R7, but the MD-CLI in the current edition is based on SR OS Release 20.10.R1.

Overview

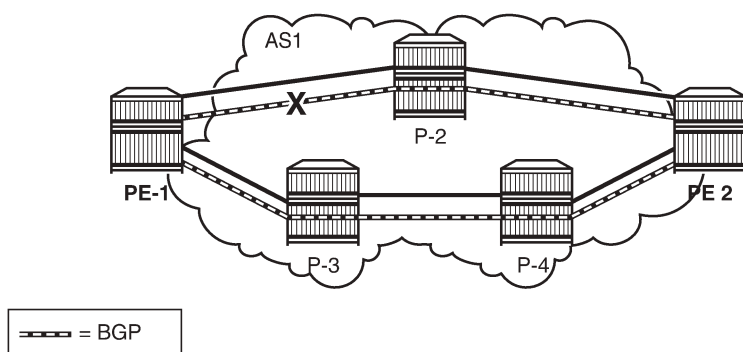
Border Gateway Protocol (BGP) is a key protocol for ISPs, supporting inter-Autonomous System (inter-AS) and intra-Autonomous System (intra-AS) applications with many address families. Additionally, ISPs need to maintain the service level agreements with their customers, even in case of network failures.

MPLS Fast Reroute (FRR) is often used to provide resiliency to intra-AS services, and relies on alternate label switched paths being established through the network. Traffic is switched to the alternate path in case of a failure of the primary path.

However, the traffic for inter-AS services crosses the boundaries of multiple ASs, so to provide resiliency, BGP FRR can be used. Before a network failure occurs, multiple paths must be received for a prefix to take advantage of this feature. When a prefix has a backup path and its primary paths fail, the affected traffic is rapidly diverted to the backup path without waiting for the control plane to reconverge. When many prefixes share the same primary paths, and in some cases also the same backup path, the time to divert traffic to the backup path is independent of the number of prefixes; this is also known as Prefix Independent Convergence (PIC). The traffic goes back to the primary paths when those paths are restored. Multiple primary paths can be active simultaneously when Equal Cost Multi Path (ECMP) applies.

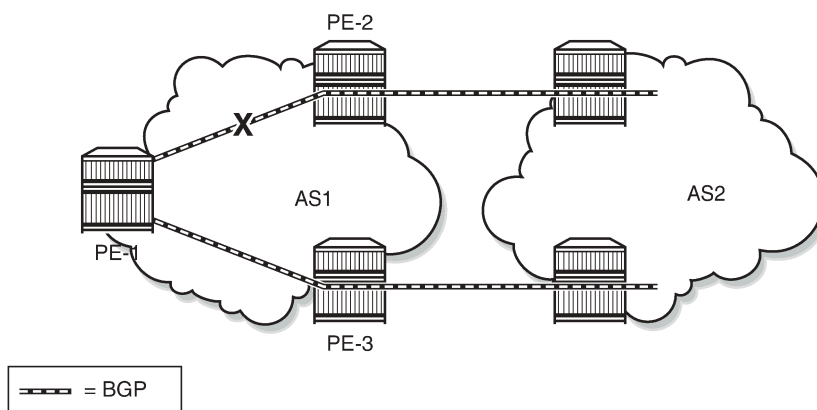
Within SR OS, two BGP FRR functions are supported: Core PIC and Edge PIC. Core PIC describes a scenario where a link or node on the path to the BGP next-hop fails, but the BGP next-hop remains reachable; see [Figure 70: Core PIC](#). Edge PIC describes a scenario where an edge node or edge link fails, which results in a change of the BGP next-hop; see [Figure 71: Edge PIC](#).

Figure 70: Core PIC



26255

Figure 71: Edge PIC



26256

Within SR OS, Core PIC is enabled by default and cannot be disabled. Therefore, this chapter will describe the use of Edge PIC.

BGP FRR is supported for different BGP address families in the **base router** context or within a specific **vprn** context. This chapter will focus on the IPv4 address family within the base router context.

The following SR OS supported features can be used to allow BGP to maintain multiple paths through an autonomous system:

- BGP best external
- BGP add-paths

Convergence goes through several phases, which also apply to BGP:

- detect the network failure
- distribute updated routing information, and update the network topology
- calculate new routes, and optionally change next-hops
- update the forwarding plane

Several mechanisms are available to enhance BGP network convergence, such as:

- Bidirectional Forwarding Detection (BFD)
- Minimum Router Advertisement Interval (MRAI)
- BGP peer tracking

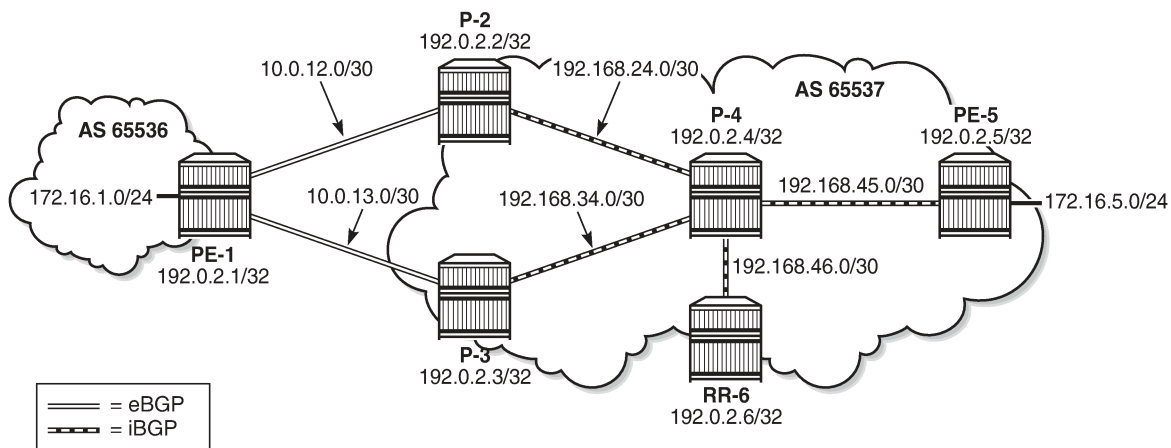
This chapter describes the use of BFD and MRAI for faster network convergence.

Configuration

The example topology used in this chapter is shown in [Figure 72: BGP FRR topology](#), and has the following characteristics:

- iBGP sessions are established between AS 65537 routers using RR-6 as route reflector with P-2, P-3, P-4, and PE-5 as route reflector clients.
- eBGP sessions are established between P-2 and P-3 of AS 65537 and PE-1 of AS 65536.
- PE-1 advertises a BGP route for prefix 172.16.1.0/24 with community "1:0" to P-2 and P-3.
- P-2 changes the local preference to 150 for the route advertised to its route reflector RR-6. Import policy "Import_LP150" on P-2 accepts routes with community "1:0" and modifies the local preference to 150; import policy "1:0" on P-3 accepts routes with community "1:0" without modifications.
- PE-5 advertises a BGP route for prefix 172.16.5.0/24 with community "1:0" to RR-6. This route is re-advertised to the other RR clients and eventually, by P-2 and P-3 to PE-1.

Figure 72: BGP FRR topology



26257

These characteristics enforce traffic for destination 172.16.1.0/24 to leave AS 65537 via P-2. P-2 (and also PE-5) learns the destination and the local preference via route reflector RR-6. But because P-3's own local preference is lower (default LP=100), it stops advertising prefix 172.16.1.0/24 toward RR-6, so that P-4 is aware of the path via P-2 only.

The objective is for P-4 to receive multiple copies of the 172.16.1.0/24 prefix with redundant next-hops, to provide for faster convergence under failure. Considering the characteristics previously listed for the topology, two features contribute for achieving this goal:

1. Using BGP best external

2. Using BGP add-paths

The BGP add-paths feature is required in scenarios with route-reflectors, possibly combined with the BGP best external feature. The BGP best external feature can be used without BGP add-paths in scenarios when the BGP peers are in a full mesh.

As a result, multiple exit paths for prefix 172.16.1.0/24 leaving AS 65537 are available, improving convergence time on the iBGP peers because they only need to update their FIBs if they lose the primary route.

BGP best external

P-3 is configured with the BGP best external feature, as follows:

```
# on P-3:
configure {
  router "Base" {
    autonomous-system 65537
    bgp {
      loop-detect discard-route
      advertise-inactive true
      split-horizon true
      advertise-external {
          ipv4 true
      }
      group "eBGP_AS65536" {
        peer-as 65536
        import {
          policy ["1:0"]
        }
        export {
          policy ["1:0"]
        }
      }
      group "iBGP_AS65537" {
        next-hop-self true
        peer-as 65537
      }
      neighbor "10.0.13.1" {
        group "eBGP_AS65536"
      }
      neighbor "192.0.2.6" {
        group "iBGP_AS65537"
      }
    }
  }
}
```

In this output, advertise-external is activated for the IPv4 address family only. It can also be activated for the IPv6, label-IPv4, and label-IPv6 address families.

Although it is not necessary to also enable BGP best external on P-2, it is not uncommon to also configure this feature on other autonomous system border routers.

P-3 advertises prefix 172.16.1.0/24 toward the route reflector RR-6, as follows:

```
[ ]
A:admin@P-3# show router bgp neighbor 192.0.2.6 advertised-routes
=====
BGP Router ID:192.0.2.3      AS:65537      Local AS:65537
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
```



```

Origin codes      l - leaked, x - stale, > - best, b - backup, p - purge
                  : i - IGP, e - EGP, ? - incomplete

=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                    Path-Id    IGP Cost
      As-Path                               -          Label
-----
i     172.16.1.0/24                         100        None
      192.0.2.3                             None        0
      65536                                  -           -
-----
Routes : 1
=====

```

The BGP best external feature is sufficient for providing alternate paths in a fully meshed autonomous system, and could be used in conjunction with the BGP add-paths feature. The BGP add-paths feature is a requirement in scenarios with route reflectors.

BGP add-paths

P-2, P-3, P-4 and RR-6 are configured with the BGP add-paths feature. PE-5 does not require the add-paths feature, because the alternate path to 172.16.1.0/24 starts in P-4.

```

# on P-2, P-3, P-4, RR-6:
configure {
  router "Base" {
    bgp {
      group "iBGP_AS65537" {
        add-paths {
          ipv4 {
            send 2
            receive true
          }
        }
      }
    }
  }
}

```

The BGP configuration on P-2 is as follows:

```

# on P-2:
configure {
  router "Base" {
    bgp {
      loop-detect discard-route
      advertise-inactive true
      split-horizon true
      group "eBGP_AS65536" {
        peer-as 65536
        import {
          policy ["Import_LP150"]
        }
        export {
          policy ["1:0"]
        }
      }
      group "iBGP_AS65537" {
        next-hop-self true
        peer-as 65537
      }
    }
  }
}

```

```

        add-paths {
            ipv4 {
                send 2
                receive true
            }
        }
    }
    neighbor "10.0.12.1" {
        group "eBGP_AS65536"
    }
    neighbor "192.0.2.6" {
        group "iBGP_AS65537"
    }
}

```

The BGP configuration for P-3 and P-4 is very similar and is not shown here.

The BGP configuration on RR-6 then is as follows:

```

# on RR-6:
configure {
    router "Base" {
        autonomous-system 65537
        bgp {
            loop-detect discard-route
            split-horizon true
            group "iBGP_AS65537" {
                peer-as 65537
                advertise-inactive true
                cluster {
                    cluster-id 6.6.6.6
                }
                add-paths {
                    ipv4 {
                        send 2
                        receive true
                    }
                }
            }
            neighbor "192.0.2.2" {
                group "iBGP_AS65537"
            }
            neighbor "192.0.2.3" {
                group "iBGP_AS65537"
            }
            neighbor "192.0.2.4" {
                group "iBGP_AS65537"
            }
            neighbor "192.0.2.5" {
                group "iBGP_AS65537"
            }
        }
    }
}

```

The default behavior of a route reflector is to only consider the best path. By enabling the add-paths feature on RR-6, multiple paths are considered.

Both P-2 and P-3 advertise route 172.16.1.0/24 to RR-6, as follows:

```

[]
A:admin@P-2# show router bgp neighbor 192.0.2.6 advertised-routes
=====
BGP Router ID:192.0.2.2      AS:65537      Local AS:65537
=====
Legend -

```

```
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
```

```
=====
BGP IPv4 Routes
=====
```

Flag	Network NextHop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
i	172.16.1.0/24 192.0.2.2 65536	150 1	None 0 -

```
-----
Routes : 1
=====
```

```
[ ]
A:admin@P-3# show router bgp neighbor 192.0.2.6 advertised-routes
```

```
=====
BGP Router ID:192.0.2.3      AS:65537      Local AS:65537
=====
```

```
Legend -
```

```
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
```

```
=====
BGP IPv4 Routes
=====
```

Flag	Network NextHop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
i	172.16.1.0/24 192.0.2.3 65536	100 1	None 0 -

```
-----
Routes : 1
=====
```

For more examples of the BGP add-paths feature, see [BGP Add-Path](#) and [BGP Multipath](#) in this guide.

Backup path

P-4 is the place in the topology where an alternate path is created. The data plane part of the Edge PIC configuration is performed by enabling the **backup-path** command within the **bgp** context. In the following, backup-paths are considered for the IPv4 address family only, but the IPv6, label-IPv4, and label-IPv6 address families are allowed too.

```
# on PE-1, P-4:
configure {
  router "Base" {
    bgp {
      backup-path {
        ipv4 true
      }
    }
  }
}
```

In this way, BGP considers all alternate paths which are present through the BGP best external and BGP add-paths feature. The BGP configuration on P-4 is as follows:

```
# on P-4:
configure {
  router "Base" {
    autonomous-system 65537
    bgp {
      loop-detect discard-route
      split-horizon true
      backup-path {
        ipv4 true
      }
      group "iBGP_AS65537" {
        peer-as 65537
        add-paths {
          ipv4 {
            send 2
            receive true
          }
        }
      }
      neighbor "192.0.2.6" {
        group "iBGP_AS65537"
      }
    }
  }
}
```

In the default BGP behavior, without the **backup-path** command, two BGP routes exist. Both routes are valid, but only the first one is the best path (indicated by ">"), as follows:

```
[ ]
A:admin@P-4# show router bgp routes 172.16.1.0/24
=====
BGP Router ID:192.0.2.4      AS:65537      Local AS:65537
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  172.16.1.0/24             150        None
      192.0.2.2             4          10
      65536                  -
*i    172.16.1.0/24             100        None
      192.0.2.3             5          10
      65536                  -
-----
Routes : 2
=====
```

The routing table then is as follows:

```
[ ]
A:admin@P-4# show router route-table protocol bgp
=====
```

```
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age           Pref
  Next Hop[Interface Name]                Metric
-----
172.16.1.0/24              Remote BGP     00h17m19s  170
  192.168.24.1                0
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

With the **backup-path** command, again both BGP routes are valid; the first route is the best path, and now the second route is explicitly marked to be a backup path (indicated by "b"), as follows:

```
[ ]
A:admin@P-4# show router bgp routes 172.16.1.0/24
=====
BGP Router ID:192.0.2.4      AS:65537      Local AS:65537
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  172.16.1.0/24            150        None
      192.0.2.2              4          10
      65536                   -
ub*i  172.16.1.0/24            100        None
      192.0.2.3              5          10
      65536                   -
-----
Routes : 2
=====
```

Now the routing table is as follows. The "B" flag indicates that a BGP backup path is available.

```
[ ]
A:admin@P-4# show router route-table protocol bgp
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age           Pref
  Next Hop[Interface Name]                Metric
-----
172.16.1.0/24 [B]          Remote BGP     00h00m16s  170
  192.168.24.1                0
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
=====
```

S = Sticky ECMP requested

To show both routes, use the following command:

```
[ ]
A:admin@P-4# show router route-table protocol bgp alternative
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age           Pref
Next Hop[Interface Name]          Metric
Alt-NextHop                       Alt-
Metric
-----
172.16.1.0/24                     Remote BGP     00h01m36s  170
192.168.24.1                       0
172.16.1.0/24 (Backup)          Remote BGP     00h01m36s  170
192.168.34.1                       0
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
      Backup = BGP backup route
      LFA = Loop-Free Alternate nexthop
      S = Sticky ECMP requested
=====
```

The currently active next-hop in the forwarding path is 192.168.24.1, as follows:

```
[ ]
A:admin@P-4# show router fib 1 ip-prefix-prefix-length 172.16.1.0/24 all
=====
FIB Display
=====
Prefix [Flags]                    Protocol  Installed
NextHop
-----
172.16.1.0/24                     BGP      Y
192.168.24.1 (int-P-4-P-2)
-----
Total Entries : 1
=====
```

The active and standby next-hops are also programmed into the forwarding path, as follows:

```
[ ]
A:admin@P-4# show router fib 1 ip-prefix-prefix-length 172.16.1.0/24 extensive
=====
FIB Display (Router: Base)
=====
Dest Prefix      : 172.16.1.0/24
Protocol         : BGP
Installed        : Y
Indirect Next-Hop : 192.0.2.2
QoS              : Priority=n/c, FC=n/c
Source-Class     : 0
Dest-Class       : 0
ECMP-Weight      : 1
Resolving Next-Hop : 192.168.24.1
Interface        : int-P-4-P-2
ECMP-Weight      : 1
=====
```

```

Indirect Next-Hop      : 192.0.2.3
QoS                   : Priority=n/c, FC=n/c
Source-Class          : 0
Dest-Class             : 0
ECMP-Weight           : 1
Backup-Path           : Yes
Resolving Next-Hop    : 192.168.34.1
  Interface            : int-P-4-P-3
  ECMP-Weight          : 1
=====
Total Entries : 1
=====

```

In summary, two paths are available out of P-4 and leading to 172.16.1.0/24 in the remote AS, but only one is installed in the forwarding plane. The active route is P-4-P-2-PE-1; the backup route is P-4-P-3-PE-1. A **traceroute** command confirms the active path, as follows:

```

[]
A:admin@PE-5# traceroute 172.16.1.1 source-address 172.16.5.1 numeric
traceroute to 172.16.1.1 from 172.16.5.1, 30 hops max, 40 byte packets
 1 192.168.45.1    0.722 ms  0.662 ms  0.646 ms
 2 192.168.24.1   1.22 ms  1.21 ms  1.21 ms
 3 172.16.1.1     3.09 ms  1.78 ms  1.74 ms

```

Faster convergence through BFD

As already described, BFD can help speed up BGP convergence, mainly when detecting network failure. In the following, BFD is enabled on the eBGP sessions, and on the IS-IS protocol.

The BFD parameters are defined at interface level, enabling BFD for an application is done in the application context. Because PE-1 only has eBFD sessions toward P-2 and P-3, it is enabled at the global BGP level, but it can also be enabled at the group or neighbor level.

```

# on PE-1:
configure {
  router "Base" {
    interface "int-PE-1-P-2" {
      ipv4 {
        bfd {
          admin-state enable
        }
      }
    }
    interface "int-PE-1-P-3" {
      ipv4 {
        bfd {
          admin-state enable
        }
      }
    }
  }
  bgp {
    bfd-liveness true
  }
}

```

Because the BFD configuration for P-2 and P-3 is very similar, it is only shown for P-2, as follows:

```

# for P-2:
configure {
  router "Base" {

```

```

interface "int-P-2-PE-1" {
  ipv4 {
    bfd {
      admin-state enable
    }
  }
}
interface "int-P-2-P-4" {
  ipv4 {
    bfd {
      admin-state enable
    }
  }
}

```

BFD is enabled for group eBGP_AS65536 only, at group level, as follows:

```

# on P-2:
configure {
  router "Base" {
    bgp {
      group "eBGP_AS65536" {
        bfd-liveness true
      }
    }
  }
}

```

BFD for IS-IS is enabled at the IS-IS interface level, and is enabled for IPv4 only, as follows.

```

# on P-2:
configure {
  router "Base" {
    isis 0 {
      interface "int-P-3-P-4" {
        bfd-liveness {
          ipv4 {
            }
          }
        }
      }
    }
  }
}

```

Faster convergence through MRAI

Adjusting the BGP MRAI also can help speed up network convergence, using the following command:

```

*[ex:configure router "Base" bgp]
A:admin@P-2# min-route-advertisement

min-route-advertisement <number>
<number> - <1..255>
Default - 30

```

Minimum time before a prefix can be advertised to peer

Lowering the MRAI puts a higher load on the CPM, so a trade-off must be made between convergence time and processing load.

Switchover

To demonstrate a switchover scenario, a failure is introduced by disabling port 1/1/1 on PE-1, as follows:

```
# on PE-1:
configure {
  port 1/1/1 {
    admin-state disable
  }
}
```

The path through the network is PE-5-P-4-P-3-PE-1, as follows:

```
[ ]
A:admin@PE-5# traceroute 172.16.1.1 source-address 172.16.5.1 numeric
traceroute to 172.16.1.1 from 172.16.5.1, 30 hops max, 40 byte packets
 1 192.168.45.1  0.698 ms  0.695 ms  0.698 ms
 2 192.168.34.1  1.21 ms  1.21 ms  1.15 ms
 3 172.16.1.1   1.73 ms  1.71 ms  1.70 ms
```

On P-4, traffic is now diverted to P-3, and the BGP routes are as follows:

```
[ ]
A:admin@P-4# show router bgp routes 172.16.1.0/24
=====
BGP Router ID:192.0.2.4      AS:65537      Local AS:65537
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
      Nexthop (Router)                     Path-Id   IGP Cost
      As-Path                               Path-Id   Label
-----
u*>i 172.16.1.0/24                           100      None
      192.0.2.3                             1        10
      65536                                  -
-----
Routes : 1
=====
```

The route table is as follows:

```
[ ]
A:admin@P-4# show router route-table protocol bgp
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age           Pref
      Next Hop[Interface Name]      Metric
-----
172.16.1.0/24                      Remote BGP    00h01m41s  170
      192.168.34.1                   0
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
```

S = Sticky ECMP requested

The forwarding plane is reprogrammed to send traffic for the 172.16.1.0/24 subnet to P-3, as follows:

```
[ ]
A:admin@P-4# show router fib 1 ip-prefix-prefix-length 172.16.1.0/24 extensive
=====
FIB Display (Router: Base)
=====
Dest Prefix           : 172.16.1.0/24
Protocol              : BGP
Installed             : Y
Indirect Next-Hop    : 192.0.2.3
  QoS                 : Priority=n/c, FC=n/c
  Source-Class       : 0
  Dest-Class         : 0
  ECMP-Weight        : 1
  Resolving Next-Hop : 192.168.34.1
    Interface        : int-P-4-P-3
    ECMP-Weight      : 1
=====
Total Entries : 1
=====
```

Bringing port 1/1/1 on PE-1 up again will result in the path PE-5-P-4-P-2-PE-1 being reactivated. Switchback takes longer, because the external BGP session needs to be re-established, and routes have to be relearned.

Conclusion

BGP FRR provides ISPs the means to offer backup paths with fast switchover times when used in combination with short failure detection times and short advertisement intervals. By guaranteeing service in case of network failures, ISPs can provide enhanced service offerings to their customers.

BGP Fast Reroute Policy Control

This chapter provides information about BGP Fast Reroute Policy Control.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially based on SR OS Release 15.0.R4, but the CLI in the current edition is based on SR OS Release 22.10.R2.

Overview

BGP Fast Reroute (FRR) allows for precomputing multiple redundant BGP paths in the control plane and installing backup routes in the forwarding plane via indirection techniques. See the [BGP Fast Reroute](#) chapter for more information.

The BGP FRR Policy Control feature allows for selectively applying FRR for designated BGP prefixes. This allows an operator to develop separate service and redundancy models for different customers or services. It also allows for using data path resources required for BGP FRR in a more efficient way.

The BGP FRR policy control feature includes the **install-backup-path** policy action command. This command is supported in the following configuration contexts:

```
[/]
A:admin@PE-3# tree flat detail | match install-backup-path
configure groups group <string> policy-options policy-statement <string> default-action
  install-backup-path <boolean>
configure groups group <string> policy-options policy-statement <string> entry <string |
  number> action install-backup-path <boolean>
configure groups group <string> policy-options policy-statement <string> named-entry <string>
  action install-backup-path <boolean>
configure policy-options policy-statement <string> default-action install-backup-path <boolean>
configure policy-options policy-statement <string> entry <number> action install-backup-path
  <boolean>
configure policy-options policy-statement <string> named-entry <string> action install-backup-
  path <boolean>
```

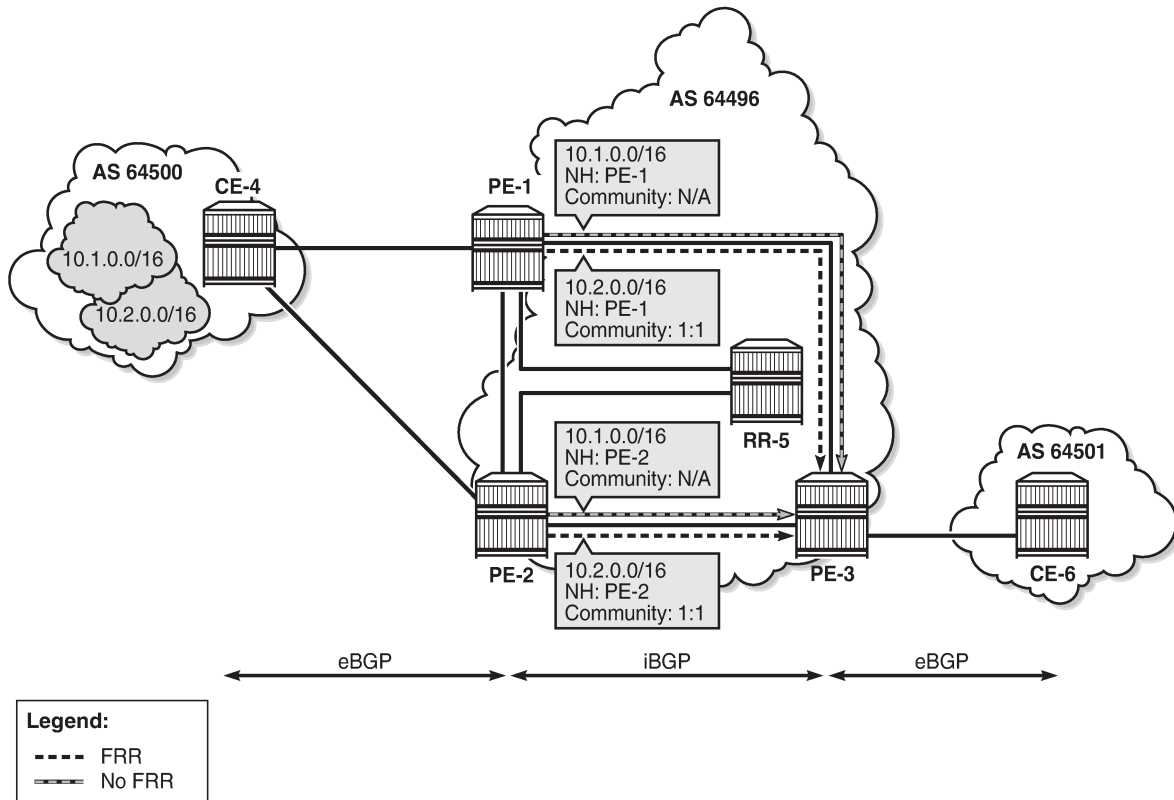
The **install-backup-path** command is effective when configured in BGP-import or VRF-import policies. In cases where this command is configured in an import policy applied in the global **bgp** context, the command applies to the following types of routes:

- IPv4
- IPv6
- Label-IPv4
- 6PE

- VPN-IPv4 (only if **vpn-apply-import** is configured in BGP)
- VPN-IPv6 (only if **vpn-apply-import** is configured in BGP)

Figure 73: Community addition on PE-1 and PE-2 shows an example of community addition. Two prefixes, 10.1.0.0/16 and 10.2.0.0/16, are advertised by CE-4 to both of its peers, PE-1 and PE-2. The administrator of AS 64496 wants to apply FRR only for the 10.2.0.0/16 prefix that will eventually be advertised to and used on PE-3, and not for 10.1.0.0/16. To facilitate this procedure, an import policy is applied on both PE-1 and PE-2 for routes advertised by CE-4 in AS 64500. The import policy selects and adds a community value of "1:1" to the 10.2.0.0/16 prefix. No community is applied to 10.1.0.0/16.

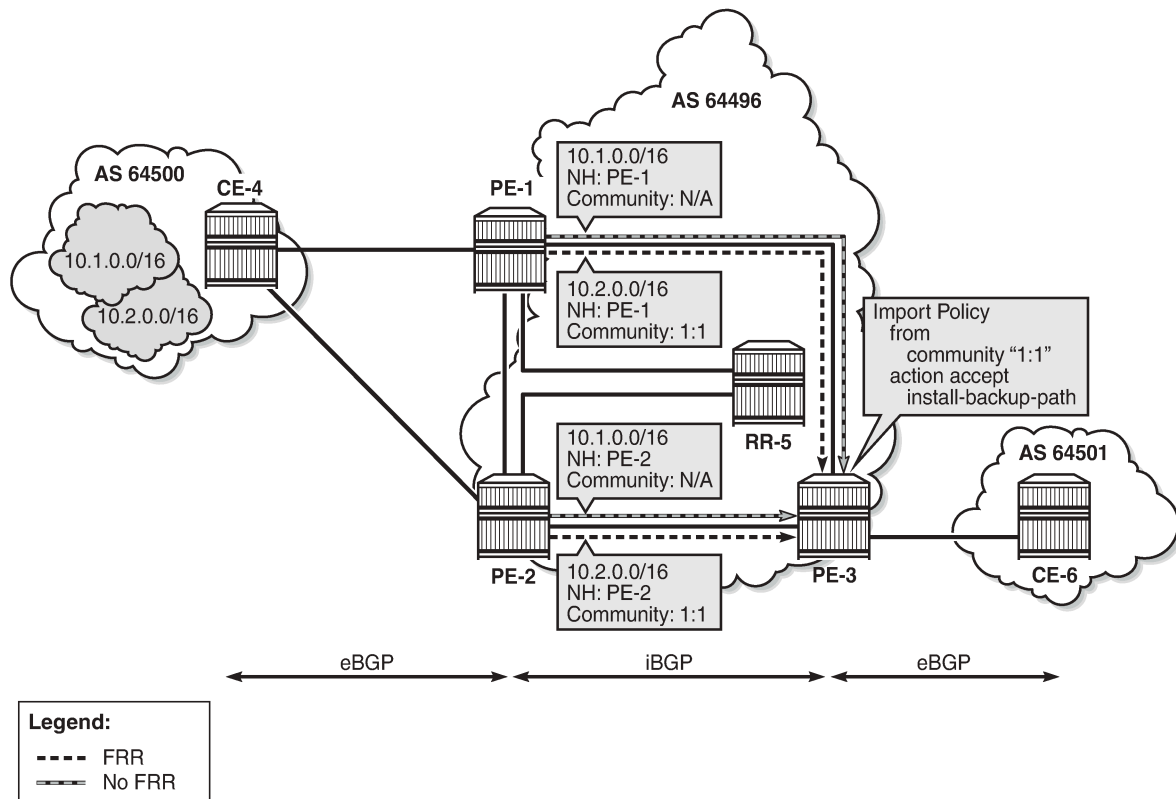
Figure 73: Community addition on PE-1 and PE-2



26770

Figure 74: FRR policy on PE-3 shows the FRR import policy applied on PE-3 for the routes received from PE-1 and PE-2. The policy matches routes with a community value of "1:1" and instructs the router to calculate and install a backup path for those matching routes.

Figure 74: FRR policy on PE-3



26771

Configuration

The following configuration examples are in this section:

- BGP FRR for address family IPv4 without FRR policy
- BGP FRR for address family IPv4 with FRR policy
- BGP with FRR policy for address family VPN-IPv4 using global BGP policy and **vpn-apply-import**
- BGP with FRR policy for address family VPN-IPv4 using VRF-import policy

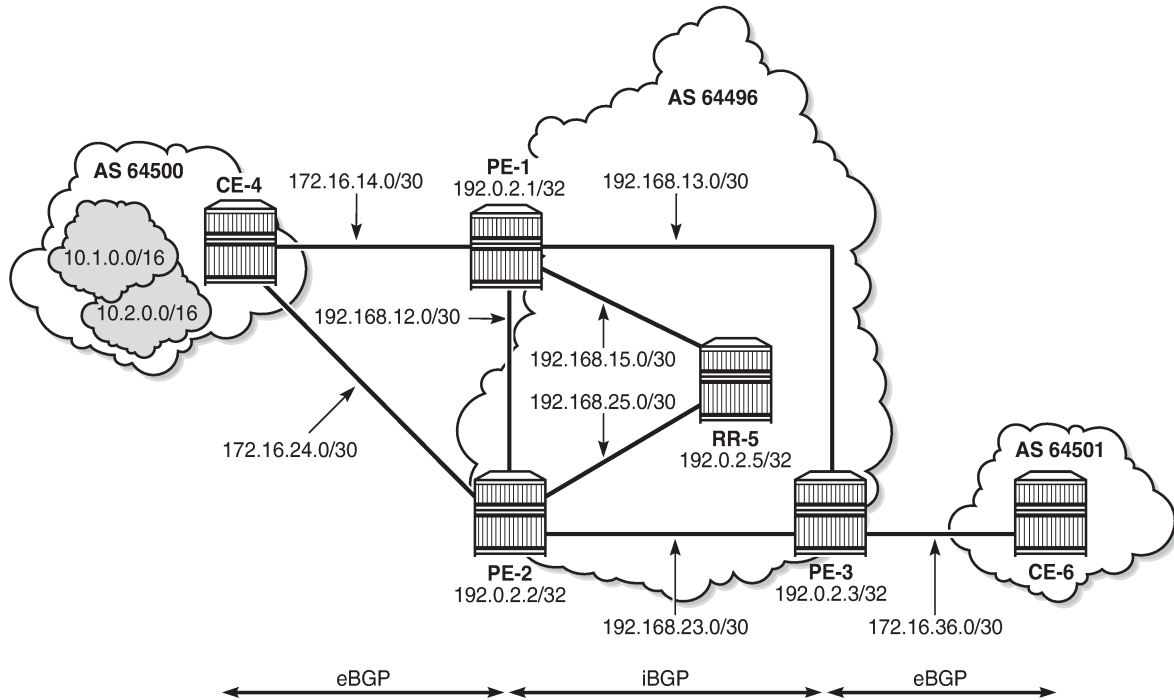
BGP FRR policy control feature for address family IPv4

Figure 3 shows the example topology used for the BGP FRR Policy Control feature for the IPv4 address family. The topology is similar to the one in the [BGP Add-Path](#) chapter, with the following characteristics:

- CE-4 in AS 64500 advertises both prefixes 10.1.0.0/16 and 10.2.0.0/16 to its eBGP peers PE-1 and PE-2 in AS 64496.
- RR-5 is route reflector for all PEs in AS 64496.
- Add-path is configured on all PE routers and RR-5 with a sending limit of 2.

- CE-6 in AS 64501 peers with PE-3 in AS 64496 and can send traffic to CE-4 in 64500.

Figure 75: Example topology - IPv4



26772

Initial configuration

The initial configuration on all nodes includes:

- Cards, MDAs, ports
- Router interfaces
- IS-IS as IGP on all interfaces within AS 64496 (alternatively, OSPF can be used)
- LDP on all interfaces between the PEs in AS 64496, but not toward RR-5. LDP is used to create the transport tunnels that are bound to the VPRN services in the VPN-IPv4 address family section.

BGP is configured on all the nodes. CE-4 peers with PE-1 and PE-2 and exports prefixes 10.1.0.0/16 and 10.2.0.0/16 to both eBGP peers, as follows:

```
# on CE-4:
configure {
  policy-options {
    prefix-list "10.1.0.0/16" {
      prefix 10.1.0.0/16 type longer {
      }
    }
  }
  prefix-list "10.2.0.0/16" {
    prefix 10.2.0.0/16 type longer {
    }
  }
}
```

```
    policy-statement "export-bgp" {
      entry 10 {
        from {
          prefix-list ["10.1.0.0/16"]
        }
        action {
          action-type accept
        }
      }
      entry 20 {
        from {
          prefix-list ["10.2.0.0/16"]
        }
        action {
          action-type accept
        }
      }
    }
  }
router "Base" {
  autonomous-system 64500
  bgp {
    rapid-withdrawal true
    split-horizon true
    ebgp-default-reject-policy {
      import false
      export false
    }
    group "eBGP" {
      export {
        policy ["export-bgp"]
      }
      peer-as 64496
    }
    neighbor "172.16.14.1" {
      group "eBGP"
    }
    neighbor "172.16.24.1" {
      group "eBGP"
    }
  }
}
```

CE-4 also has configured the following loopback interfaces:

```
# on CE-4:
configure {
  router "Base" {
    interface "int-loopback-1" {
      ipv4 {
        primary {
          address 10.1.1.1
          prefix-length 16
        }
      }
    }
    loopback
  }
  interface "int-loopback-2" {
    ipv4 {
      primary {
        address 10.2.1.1
        prefix-length 16
      }
    }
  }
}
```

```

    }
    loopback
}

```

The BGP configuration on CE-6 is similar, except for the export policy.

PE-1 peers with CE-4 in AS 65400 and RR-5 in AS 64496. The BGP configuration on PE-1 is as follows:

```

# on PE-1:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      rapid-withdrawal true
      split-horizon true
      ebgp-default-reject-policy {
        import false
        export false
      }
      group "eBGP" {
        peer-as 64500
      }
      neighbor "172.16.14.2" {
        group "eBGP"
      }
      group "iBGP" {
        next-hop-self true
        peer-as 64496
        add-paths {
          ipv4 {
            send 2
            receive true
          }
        }
      }
      neighbor "192.0.2.5" {
        group "iBGP"
      }
    }
  }
}

```

The BGP configuration on PE-2 and PE-3 is similar to PE-1.

RR-5 acts as a route reflector to all the PEs in AS 64500 with a cluster ID of 5.5.5.5. The BGP configuration on RR-5 is as follows:

```

# on RR-5:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      rapid-withdrawal true
      split-horizon true
      ebgp-default-reject-policy {
        import false
        export false
      }
      group "iBGP" {
        cluster {
          cluster-id 5.5.5.5
        }
        peer-as 64496
        add-paths {
          ipv4 {
            send 2
          }
        }
      }
    }
  }
}

```



```

        receive true
    }
}
neighbor "192.0.2.1" {
    group "iBGP"
}
neighbor "192.0.2.2" {
    group "iBGP"
}
neighbor "192.0.2.3" {
    group "iBGP"
}

```

BGP FRR for address family IPv4 without FRR policy

PE-3 receives both prefixes from PE-1 and PE-2 via RR-5, but only uses the one from PE-1 (Nexthop: 192.0.2.1).

```

[/]
A:admin@PE-3# show router bgp routes
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  10.1.0.0/16                100        None
      192.0.2.1                10         10
      64500                      -
*i    10.1.0.0/16                100        None
      192.0.2.2                3          10
      64500                      -
u*>i  10.2.0.0/16                100        None
      192.0.2.1                11         10
      64500                      -
*i    10.2.0.0/16                100        None
      192.0.2.2                4          10
      64500                      -
-----
Routes : 4
=====

```

The following configuration is applied on PE-3 to enable BGP FRR:

```

# on PE-3:
configure {
    router "Base" {
        bgp {
            backup-path ipv4
        }
    }
}

```

PE-3 calculates and marks BGP routes from PE-2 as backup routes in the BGP routing table:

```
[/]
A:admin@PE-3# show router bgp routes
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
u*>i  10.1.0.0/16                100        None
      192.0.2.1              10         10
      64500                   -
ub*i 10.1.0.0/16              100        None
      192.0.2.2              3          10
      64500                   -
u*>i  10.2.0.0/16                100        None
      192.0.2.1              11         10
      64500                   -
ub*i 10.2.0.0/16              100        None
      192.0.2.2              4          10
      64500                   -
-----
Routes : 4
=====
```

PE-3 installs BGP routes from PE-2 as backup routes in its route table:

```
[/]
A:admin@PE-3# show router route-table 10.0.0.0/8 longer alternative
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age          Pref
Next Hop[Interface Name]   Metric
Alt-NextHop                 Alt-
Metric
-----
10.1.0.0/16                 Remote BGP    00h02m47s  170
      192.168.13.1          10
10.1.0.0/16 (Backup)      Remote BGP  00h02m47s  170
      192.168.23.1          10
10.2.0.0/16                 Remote BGP    00h02m47s  170
      192.168.13.1          10
10.2.0.0/16 (Backup)      Remote BGP  00h02m47s  170
      192.168.23.1          10
-----
No. of Routes: 4
Flags: n = Number of times nexthop is repeated
      Backup = BGP backup route
      LFA = Loop-Free Alternate nexthop
      S = Sticky ECMP requested
```

BGP FRR for address family IPv4 with FRR policy

The global BGP FRR activation command enabled on PE-3 in the previous step is removed from the configuration:

```
# on PE-3:
configure {
  router "Base" {
    bgp {
      backup-path delete ipv4
    }
  }
}
```

The following command output on PE-3 shows no community values attached to the prefix 10.2.0.0/16 advertised by PE-1 and PE-2:

```
[/]
A:admin@PE-3# show router bgp routes 10.2.0.0/16 detail | match '^NextHop|Community'
NextHop      : 192.0.2.1
Community    : No Community Members
NextHop      : 192.0.2.1
Community    : No Community Members
NextHop      : 192.0.2.2
Community    : No Community Members
NextHop      : 192.0.2.2
Community    : No Community Members
```

The following policy is configured on PE-1 and PE-2 to add the BGP community "1:1" to the prefix 10.2.0.0/16 advertised by CE-4:

```
# on PE-1 and PE-2:
configure {
  policy-options {
    community "1:1" {
      member "1:1" { }
    }
  }
  prefix-list "10.2.0.0/16" {
    prefix 10.2.0.0/16 type longer {
    }
  }
  policy-statement "add-bgp-community" {
    entry 10 {
      from {
        prefix-list ["10.2.0.0/16"]
      }
      action {
        action-type accept
        community {
          add ["1:1"]
        }
      }
    }
  }
}
```

The policy is applied as a BGP-import policy on PE-1 and PE-2 for the eBGP group:

```
# on PE-1, PE-2:
configure {
```

```
router "Base" {
  bgp {
    group "eBGP" {
      import {
        policy ["add-bgp-community"]
      }
    }
  }
}
```

PE-3 now shows the community value associated with prefix 10.2.0.0/16 as applied and advertised by PE-1 and PE-2:

```
[/]
A:admin@PE-3# show router bgp routes 10.2.0.0/16 detail | match '^NextHop|Community'
NextHop      : 192.0.2.1
Community    : 1:1
NextHop      : 192.0.2.1
Community    : 1:1
NextHop      : 192.0.2.2
Community    : 1:1
NextHop      : 192.0.2.2
Community    : 1:1
```

The following policy is configured on PE-3 to selectively install a backup path only for prefixes with a community value equal to "1:1":

```
# on PE-3:
configure {
  policy-options {
    community "1:1" {
      member "1:1" { }
    }
  }
  policy-statement "policy-bgp-frr-import" {
    entry 10 {
      from {
        community {
          name "1:1"
        }
      }
      action {
        action-type accept
        install-backup-path true
      }
    }
  }
}
```

The policy is applied on PE-3 to selectively install a backup path only for prefixes with a community value equal to "1:1":

```
# on PE-3:
configure {
  router "Base" {
    bgp {
      group "iBGP" {
        import {
          policy ["policy-bgp-frr-import"]
        }
      }
    }
  }
}
```

The following command output shows PE-3 has calculated a BGP FRR path only for prefix 10.2.0.0/16 indicated by the "b" (backup) flag:

```
[/]
A:admin@PE-3# show router bgp routes
=====
```

```

BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  10.1.0.0/16                             100        None
      192.0.2.1                             10         10
      64500                                  -
*i    10.1.0.0/16                             100        None
      192.0.2.2                             3          10
      64500                                  -
u*>i  10.2.0.0/16                             100        None
      192.0.2.1                             11         10
      64500                                  -
ub*i  10.2.0.0/16                             100        None
      192.0.2.2                             4          10
      64500                                  -
-----
Routes : 4
=====

```

The following command output shows PE-3 has installed a backup route only for prefix 10.2.0.0/16 in its route table:

```

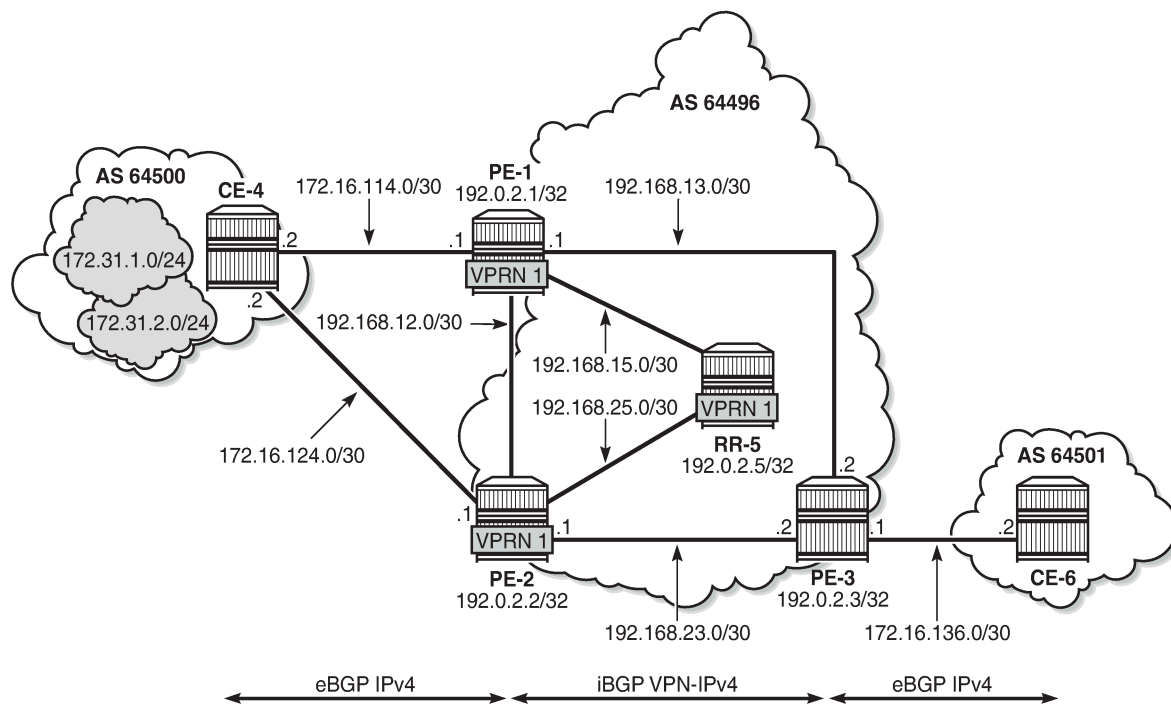
[/]
A:admin@PE-3# show router route-table 10.0.0.0/8 longer alternative
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age          Pref
  Next Hop[Interface Name]        Metric
  Alt-NextHop                      Alt-
                                   Metric
-----
10.1.0.0/16                        Remote BGP    00h09m48s  170
  192.168.13.1                      10
10.2.0.0/16                        Remote BGP    00h01m12s  170
  192.168.13.1                      10
10.2.0.0/16 (Backup)             Remote BGP 00h01m12s  170
  192.168.23.1                    10
-----
No. of Routes: 3
Flags: n = Number of times nexthop is repeated
      Backup = BGP backup route
      LFA = Loop-Free Alternate nexthop
      S = Sticky ECMP requested
=====

```

BGP with FRR policy for address family VPN-IPv4 using global BGP policy

Figure 76: Example topology - VPN-IPv4 shows the example topology used to illustrate the BGP FRR policy control feature for the VPN-IPv4 route family. CE-4 exports both prefixes 172.31.1.0/24 and 172.31.2.0/24 to VPRN 1 on PE-1 and PE-2.

Figure 76: Example topology - VPN-IPv4



26773

VPRN 1 is configured on all PEs in AS 64496. The configuration of VPRN 1 is similar on all PEs; for example, for PE-1, the VPRN configuration is as follows:

```
# on PE-1:
configure {
  service {
    vprn "VPRN 1" {
      customer "1"
      service-id 1
      autonomous-system 64496
      interface "int-PE-1-CE-4-VPRN1" {
        ipv4 {
          primary {
            address 172.16.114.1
            prefix-length 30
          }
        }
        sap 1/1/c1/2:1 {
        }
      }
    }
  }
  bgp {
    split-horizon true
    ebgp-default-reject-policy {
      import false
    }
  }
}
```

```

        export false
    }
    group "eBGP-1" {
        peer-as 64500
    }
    neighbor "172.16.114.2" {
        group "eBGP-1"
    }
}
bgp-ipvpn {
    mpls {
        admin-state enable
        route-distinguisher "64496:1"
        vrf-target {
            community "target:64496:1"
        }
        auto-bind-tunnel {
            resolution any
        }
    }
}
admin-state enable

```

On the CEs, the configuration is either in the base routing instance, with additional router interfaces and BGP neighbors, or in a VPRN. In this example, the following VPRN is configured on CE-4:

```

# on CE-4:
service {
    vprn "VPRN 1" {
        customer "1"
        service-id 1
        autonomous-system 64500
        interface "int-CE-4-PE-1-VPRN1" {
            ipv4 {
                primary {
                    address 172.16.114.2
                    prefix-length 30
                }
            }
            sap 1/1/c1/1:1 {
            }
        }
        interface "int-CE-4-PE-2-VPRN1" {
            ipv4 {
                primary {
                    address 172.16.124.2
                    prefix-length 30
                }
            }
            sap 1/1/c1/2:1 {
            }
        }
        interface "loopback1-VPRN1" {
            ipv4 {
                primary {
                    address 172.31.1.1
                    prefix-length 24
                }
            }
            loopback true
        }
        interface "loopback2-VPRN1" {
            ipv4 {
            }
        }
    }
}

```

```

        primary {
            address 172.31.2.1
            prefix-length 24
        }
    }
    loopback true
}
bgp {
    split-horizon true
    ebgp-default-reject-policy {
        import false
        export false
    }
    group "eBGP-1" {
        export {
            policy ["export-VPRN1"]
        }
        peer-as 64496
    }
    neighbor "172.16.114.1" {
        group "eBGP-1"
    }
    neighbor "172.16.124.1" {
        group "eBGP-1"
    }
}
bgp-ipvpn {
    mpls {
        admin-state enable
        route-distinguisher "64500:1"
    }
}
admin-state enable

```

The export policy to export prefixes 172.31.1.0/24 and 172.31.2.0/24 is defined as follows:

```

# on CE-4:
configure {
    policy-options {
        prefix-list "172.31.0.0/16" {
            prefix 172.31.0.0/16 type longer {
            }
        }
    }
    policy-statement "export-VPRN1" {
        entry 10 {
            from {
                prefix-list ["172.31.0.0/16"]
            }
            action {
                action-type accept
            }
        }
    }
}

```

The VPRN configuration on CE-6 is similar, but no prefix is exported from CE-6.

For all BGP speakers in AS 64496, BGP must be configured for address family VPN-IPv4 as well as for IPv4, as follows:

```

# on PE-1, PE-2, PE-3, RR-5:
configure {
    router "Base" {
        bgp {

```



```

group "iBGP" {
  family {
    ipv4 true
    vpn-ipv4 true
  }
}

```

BGP add-path cannot be enabled in the **bgp** context within a VPRN. However, it can be enabled in the base routing instance for address family VPN-IPv4. This is done on all PEs and RR-5 at group level with the following command:

```

# on PE-1, PE-2, PE-3, RR-5:
configure {
  router "Base" {
    bgp {
      group "iBGP" {
        add-paths {
          vpn-ipv4 {
            send 2
            receive true
          }
        }
      }
    }
  }
}

```

The BGP configuration on PE-1 is as follows:

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      rapid-withdrawal true
      split-horizon true
      ebgp-default-reject-policy {
        import false
        export false
      }
      group "eBGP" {
        peer-as 64500
        import {
          policy ["add-bgp-community"]
        }
      }
      group "iBGP" {
        next-hop-self true
        peer-as 64496
        family {
          ipv4 true
          vpn-ipv4 true
        }
        add-paths {
          ipv4 {
            send 2
            receive true
          }
          vpn-ipv4 {
            send 2
            receive true
          }
        }
      }
    }
    neighbor "172.16.14.2" {
      group "eBGP"
    }
    neighbor "192.0.2.5" {
      group "iBGP"
    }
  }
}

```

```
}

```

With add-path enabled for address family VPN-IPv4, PE-1 and PE-2 will advertise their routes for prefixes 172.31.1.0/24 and 172.31.2.0/24 as VPN-IPv4 routes to RR-5. RR-5 will advertise both routes to its other RR clients. PE-3 receives two VPN-IPv4 routes for each of the prefixes 172.31.1.0/24 and 172.31.2.0/24, as follows:

```
[/]
A:admin@PE-3# show router bgp routes 172.31.0.0/16 vpn-ipv4 longer
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                       Path-Id    IGP Cost
      As-Path
-----
u*>i  64496:1:172.31.1.0/24                    100        None
      192.0.2.1                               21         10
      64500                                    524283
*>i   64496:1:172.31.1.0/24                    100        None
      192.0.2.2                               9          10
      64500                                    524283
u*>i  64496:1:172.31.2.0/24                    100        None
      192.0.2.1                               20         10
      64500                                    524283
*>i   64496:1:172.31.2.0/24                    100        None
      192.0.2.2                               10         10
      64500                                    524283
-----
Routes : 4
=====
```

The following policy is configured on PE-1 and PE-2 to include the community value "1:1" to prefix 172.31.2.0/24, as well as to the VPRN route target 64496:1 within entry 10. All the other routes are tagged with only the VPRN route target 64496:1 in entry 20.

```
# on PE-1 and PE-2:
configure {
  policy-options {
    community "1:1" {
      member "1:1" { }
    }
    community "target:64496:1" {
      member "target:64496:1" { }
    }
    prefix-list "172.31.2.0/24" {
      prefix 172.31.2.0/24 type longer {
    }
  }
  }
  policy-statement "policy-export-VPRN1" {
    entry 10 {
      from {
        prefix-list ["172.31.2.0/24"]

```

```

    }
    action {
      action-type accept
      community {
        add ["1:1" "target:64496:1"]
      }
    }
  }
  entry 20 {
    from {
    }
    action {
      action-type accept
      community {
        add ["target:64496:1"]
      }
    }
  }
}

```

The policy is applied as a VRF-export policy in VPRN 1 on PE-1 and PE-2:

```

# on PE-1, PE-2:
configure {
  service {
    vprn "VPRN 1" {
      bgp-ipvpn {
        mpls {
          admin-state enable
          vrf-export {
            policy ["policy-export-VPRN1"]
          }
        }
      }
    }
  }
}

```

On PE-3, prefix 172.31.1.0/24 is received with the community value of the VPRN route target only:

```

[/]
A:admin@PE-3# show router bgp routes 172.31.1.0/24 vpn-ipv4 hunt | match "Community"
Community      : target:64496:1
Community      : target:64496:1

```

However, prefix 172.31.2.0/24 is received with both community values "1:1" and "target:64496:1" from PE-1 and PE-2:

```

[/]
A:admin@PE-3# show router bgp routes 172.31.2.0/24 vpn-ipv4 hunt | match "Community"
Community      : 1:1 target:64496:1
Community      : 1:1 target:64496:1

```

The following command is applied on PE-3 to make the policy named "policy-bgp-frr-import", configured in the previous section for IPv4 routes, effective also on VPN-IPv4 routes:

```

# on PE-3:
configure {
  router "Base" {
    bgp {
      vpn-apply-import true
    }
  }
}

```

PE-3 now has a BGP backup path only for prefix 172.31.2.0/24, as indicated by the "b" (backup) flag:

```

[/]
A:admin@PE-3# show router bgp routes 172.31.0.0/16 vpn-ipv4 longer

```

```

=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  64496:1:172.31.1.0/24                 100        None
      192.0.2.1                             21         10
      64500                                   524283
*>i   64496:1:172.31.1.0/24                 100        None
      192.0.2.2                             9          10
      64500                                   524283
u*>i  64496:1:172.31.2.0/24                 100        None
      192.0.2.1                             20         10
      64500                                   524283
ub*>i 64496:1:172.31.2.0/24                 100        None
      192.0.2.2                             10         10
      64500                                   524283
-----
Routes : 4
=====

```

PE-3 has installed a backup route only for prefix 172.31.2.0/24 in its VPRN route table:

```

[/]
A:admin@PE-3# show router 1 route-table 172.31.0.0/16 longer alternative
=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
  Next Hop[Interface Name]        Metric
  Alt-NextHop                      Alt-
                                   Metric
-----
172.31.1.0/24                      Remote BGP VPN 00h01m47s 170
  192.0.2.1 (tunneled)              10
172.31.2.0/24                      Remote BGP VPN 00h01m47s 170
  192.0.2.1 (tunneled)              10
172.31.2.0/24 (Backup)             Remote BGP VPN 00h01m47s 170
  192.0.2.2 (tunneled)             10
-----
No. of Routes: 3
Flags: n = Number of times nexthop is repeated
      Backup = BGP backup route
      LFA = Loop-Free Alternate nexthop
      S = Sticky ECMP requested
=====

```

BGP with FRR policy for address family VPN-IPv4 using VRF-import policy

The `vpn-apply-import` command enabled in the previous section is removed from the BGP configuration on PE-3:

```
# on PE-3:
configure {
  router "Base" {
    bgp {
      delete vpn-apply-import
    }
  }
}
```

PE-3 removes the backup path for prefix 172.31.2.0/24:

```
[/]
A:admin@PE-3# show router 1 route-table 172.31.0.0/16 longer alternative

=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]                               Type  Proto  Age           Pref
  Next Hop[Interface Name]                       Metric
  Alt-NextHop                                     Alt-
                                                Metric
-----
172.31.1.0/24                                     Remote BGP VPN 00h01m36s 170
      192.0.2.1 (tunneled)                        10
172.31.2.0/24                                     Remote BGP VPN 00h01m36s 170
      192.0.2.1 (tunneled)                        10
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
      Backup = BGP backup route
      LFA = Loop-Free Alternate nexthop
      S = Sticky ECMP requested
=====
```

The following policy is configured to selectively apply FRR for prefixes with a matching community value equal to "1:1" and "target:64496:1" on PE-3:

```
# on PE-3:
configure {
  policy-options {
    community "1:1" {
      member "1:1" { }
    }
    community "target:64496:1" {
      member "target:64496:1" { }
    }
  }
  policy-statement "policy-import-VPRN1" {
    entry 10 {
      from {
        community {
          expression "[target:64496:1] AND [1:1]"
        }
      }
      action {
        action-type accept
        install-backup-path true
      }
    }
  }
  default-action {
```

```

        action-type accept
    }
}

```

The policy is applied as a VRF-import policy in VPRN 1 on PE-3:

```

# on PE-3:
configure {
  service {
    vprn "VPRN 1" {
      bgp-ipvpn {
        mpls {
          admin-state enable
          vrf-import {
            policy ["policy-import-VPRN1"]
          }
        }
      }
    }
  }
}

```

PE-3 again installs a backup path only for prefix 172.31.2.0/24 and not for 172.31.1.0/24:

```

[/]
A:admin@PE-3# show router 1 route-table 172.31.0.0/16 longer alternative
=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
Next Hop[Interface Name]          Metric
Alt-NextHop                        Alt-
Metric
-----
172.31.1.0/24                      Remote BGP VPN 00h01m42s 170
192.0.2.1 (tunneled)                10
172.31.2.0/24                      Remote BGP VPN 00h01m42s 170
192.0.2.1 (tunneled)                10
172.31.2.0/24 (Backup)             Remote BGP VPN 00h01m42s 170
192.0.2.2 (tunneled)             10
-----
No. of Routes: 3
Flags: n = Number of times nexthop is repeated
Backup = BGP backup route
LFA = Loop-Free Alternate nexthop
S = Sticky ECMP requested
=====

```

Conclusion

The BGP FRR policy control feature allows for selectively applying FRR for designated prefixes. The feature brings more flexibility and granularity to the BGP FRR implementation.

BGP FlowSpec for IPv4 and IPv6

This chapter provides information about BGP FlowSpec for IPv4 and IPv6.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The configuration and information in this chapter are based on SR OS Release 22.7.R1.

Overview

The base BGP Flow Specification (FlowSpec) is defined in RFC 5575 (Flow Specification) and describes a method of encoding IPv4 flow specification information into Network Layer Reachability Information (NLRI). RFC 8955 updates RFC 5575 and RFC 8956 includes the IPv6 address family. The flow specification is an n-tuple consisting of one or more matching criteria, which can be applied to IP traffic. The FlowSpec NLRI is encoded into Multiprotocol BGP using MP_REACH_NLRI and MP_UNREACH_NLRI attributes.

As well as the flow specification defining match criteria, extended community attributes are defined to provide traffic filtering actions for the specified flow specification. Therefore, a FlowSpec route (MP_REACH_NLRI) contains a description of the traffic to be matched (using FlowSpec NLRI), and the filtering action to be taken with that traffic (using traffic filtering action extended communities). RFC 7674 provided an update to the original RFC 5575 specification to clarify the formatting of some of these traffic actions, notably redirect to VRF.

The use of FlowSpec is to dynamically distribute traffic filtering rules for mitigating distributed denial of service (DDoS) attacks. A router receiving a FlowSpec update can dynamically create IP filters to mitigate both intra-AS and inter-AS DDoS attacks. Mitigation is implemented by dropping traffic at the ingress point of the network (or nearest possible point toward the source of the DDoS attack) or by redirecting traffic to a separate routing context for forwarding (off-ramping) to a traffic-cleansing device. The ability to redirect traffic led to FlowSpec being considered for software defined networking (SDN)-driven applications or network re-optimization tools. In those cases, a subset of traffic needs to be forced (redirected) into a specific routing context or tunnel/label switched path (LSP) for network capacity optimization or to meet a service level agreement (SLA).

BGP FlowSpec uses AFI 1 (IPv4) or AFI 2 (IPv6) with SAFI 133 (IPv4 dissemination of flow specification rules) or SAFI 134 (VPNv4 dissemination of flow specification rules). SR OS supports IPv4 and IPv6. In SR OS Release 22.7.R1 and later, VPN-IPv4 and VPN-IPv6 are also supported.

The FlowSpec NLRI may consist of several components that form the flow specification. A packet only matches the flow specification when it matches all of the components in the NLRI. In the *BGP FlowSpec* section of the *7450 ESS*, *7750 SR*, *7950 XRS*, and *VSR Unicast Routing Protocols Guide*, tables *Subcomponents of FlowSpec IPv4 and FlowSpec-VPN IPv4 NLRI* and *Subcomponents of FlowSpec IPv6 and FlowSpec-VPN IPv6 NLRI* list the component types that are currently defined, their type values,

and their support in SR OS. Flow specification components must follow strict ordering. If present in the specification, a component must precede any other component of higher type value.

The traffic filtering action for a flow specification uses a number of extended community attributes. The attributes standardized in RFC 5575 are listed in the tables *IPv4 FlowSpec actions* and *IPv6 FlowSpec actions* in the *BGP FlowSpec* section of the *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide*. The traffic rate extended community specifies the rate in bytes per second, where a rate of zero specifies a drop action. The traffic action extended community consists of six bytes; only the two least significant bits of the last byte are currently defined. The terminal action (T-bit), when set to 1, indicates that subsequent filtering rules should be applied (like a next-entry action). When this bit is set to zero, and this action is applied, the evaluation of the traffic filter stops. The sample bit (S-bit), when set to 1, enables traffic sampling and logging for this flow specification. The **redirect-to-vrf** and mark traffic class extended communities are self-explanatory, with a route-target value being used to define the target redirect VRF.

FlowSpec routes are typically originated and contained within the administrative domain of an operator; particularly when used for DDoS mitigation purposes. This approach means applying ingress filters at the point where traffic enters the autonomous system (AS), such as an external peering point.

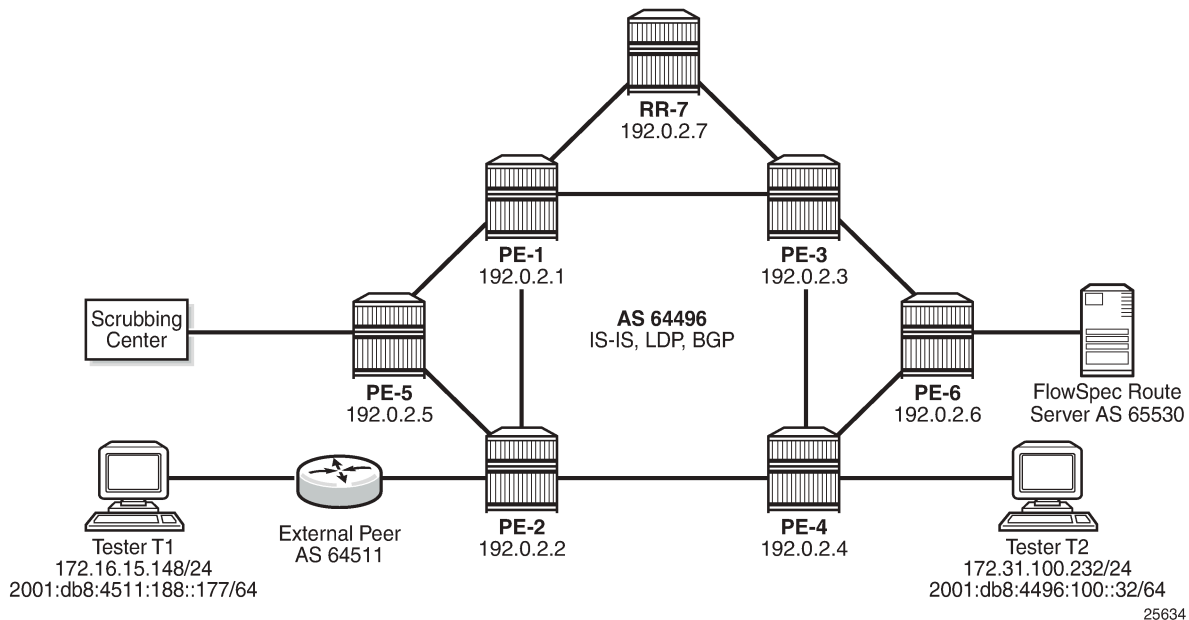
These filters should be instantiated as close as possible to the source of the attack traffic, even if that means applying filters within another operator's domain. This means that FlowSpec routes must be exchanged between ASs, requiring a trust relationship between the ASs, and a method for validating FlowSpec routes exchanged across AS boundaries. This is covered in the *BGP FlowSpec Route Validation* chapter.

Example topology

The example topology used in this chapter is shown in [Figure 77: Example topology](#). PE-1 through PE-6 and RR-7 participate in IS-IS Level-2 and LDP. All these devices are part of network AS 64496, with all PE routers peering in IBGP with the Route Reflector RR-7 for address families IPv4, IPv6, VPN-IPv4, VPN-IPv6, Label-IPv4, Label-IPv6, Flow-IPv4, and Flow-IPv6.

By including the Label-IPv4 and Label-IPv6 address families, generating labeled routes, and resolving these labeled routes to LDP tunnels on all PEs in the topology, IPv4 and IPv6 traffic is tunneled in MPLS.

Figure 77: Example topology



To demonstrate FlowSpec, the following items are connected to AS 64496:

- PE-2 is connected to an external peer in AS 64511, which advertises the IPv4 prefix 172.16.0.0/20 and the IPv6 prefix 2001:db8:4511::/48 in EBGP. Both prefixes are advertised within AS 64496 by PE-2 as labeled routes.
- PE-4 advertises IPv4 prefix 172.31.100.0/24 and IPv6 prefix 2001:db8:4496::/48 into IBGP, which PE-2 subsequently advertises in EBGP to AS 64511.
- Tester T1 is connected to the external peer in AS 64511 and sources and sinks traffic from IPv4 address 172.16.15.148 and IPv6 address 2001:db8:4511:188::177. Tester T2 is connected to PE-4 and sources and sinks traffic from IPv4 address 172.31.100.232 and IPv6 address 2001:db8:4496:100::32.
- PE-6 externally peers with a FlowSpec route server belonging to AS 65530.
- PE-5 connects to a DDoS scrubbing center with two interfaces:
 - A "dirty" interface for forwarding of mitigated traffic toward the scrubbing center for cleansing. This interface is connected to an off-ramp VPRN configured on PE-5 and PE-2. PE-5 has static IPv4/IPv6 default routes toward the scrubbing center, which are subsequently advertised into the off-ramp VPRN. This provides sufficient routing information to attract redirected traffic from PE-2 toward the scrubbing center for cleansing.
 - A "clean" interface for traffic received from the scrubbing center after it has been cleansed. This interface is connected to an IES service and is therefore routed toward its destination using the Global Routing Table (GRT).

Configuration

As an example of FlowSpec configuration, the following output shows the BGP configuration on PE-1. Similar configurations are applied to all other PE routers. All PE routers within AS 64496 peer as clients

with RR-7 for the address families IPv4, IPv6, VPN-IPv4, VPN-IPv6, Label-IPv4, Label-IPv6, Flow-IPv4, and Flow-IPv6. The Label-IPv4 and Label-IPv6 address families are required for labeled routes, and the resolution filter enables IPv4 and IPv6 traffic to pass through the MPLS/LDP transport tunnels. The Flow-IPv4 and Flow-IPv6 address families are required for propagating the FlowSpec routes, and represent the only part of the BGP configuration required by FlowSpec.

```
A:admin@PE-1# admin show configuration /configure router bgp
admin-state enable
path-mtu-discovery false
split-horizon true
group "IBGP" {
  type internal
  family {
    ipv4 true
    vpn-ipv4 true
    ipv6 true
    vpn-ipv6 true
    flow-ipv4 true
    flow-ipv6 true
    label-ipv4 true
    label-ipv6 true
  }
}
neighbor "192.0.2.7" {
  admin-state enable
  group "IBGP"
}
```

PE-2 peers with AS 64511 through an IES service interface using the IPv4 and IPv6 address families, with a dedicated BGP session for each family. This external peering point is the point where the IPv4 and IPv6 filters embedding the flowspec filters are applied. In the following output, these filters are applied in the SAP ingress context, to enable FlowSpec for IPv4 and IPv6, respectively. Such filters can also be enabled on spoke-SDPs within routed interfaces, and is supported within the base and VPRN routing instances.

```
A:admin@PE-2# admin show configuration /configure service ies 10
admin-state enable
description "Flowspec-test"
customer "1"
interface "to-AS64511" {
  sap 1/1/c4/1:10 {
    ingress {
      filter {
        ip "104"
        ipv6 "106"
      }
    }
  }
  ipv4 {
    primary {
      address 192.168.2.1
      prefix-length 30
    }
  }
  ipv6 {
    address 2001:db8:1b0c:2121::2 {
      prefix-length 127
    }
  }
}
```

FlowSpec operation

With FlowSpec enabled and configured as in previous section, FlowSpec routes can be advertised to dynamically trigger the instantiation of embedded filters. When valid FlowSpec routes are received, the FlowSpec filters are created. These FlowSpec filters must be referenced from the operator-defined IPv4 or IPv6 filters, for example as follows. These operator-defined filters must be applied to the interfaces in the ingress context for FlowSpec to work.

```
A:admin@PE-2# admin show configuration /configure filter
ip-filter "104" {
  default-action accept
  embed {
    flowspec offset 10000 {
      router-instance "Base"
    }
  }
}
ipv6-filter "106" {
  default-action accept
  embed {
    flowspec offset 10000 {
      router-instance "Base"
    }
  }
}
```

This section demonstrates the use of FlowSpec for traffic black-holing and traffic redirection for both IPv4 and IPv6.

IPv4 FlowSpec

To validate the instantiation of ingress filters based on IPv4 FlowSpec routes, a bidirectional traffic stream is started between T1 (172.16.15.148) in AS 64511 and T2 (172.31.100.232) in AS 64496. In the T1 to T2 direction, the destination port is TCP port 4191.

An IPv4 FlowSpec route is generated to black-hole/drop traffic with a source address of 172.16.15.148 (T1) and a destination address of 172.31.100.232 (T2), for any destination ports in the range 4191-4198. The following output shows the route as received at PE-2.

```
<timestamp> MINOR: DEBUG #2001 Base Peer 1: 192.0.2.7
"Peer 1: 192.0.2.7: UPDATE
Peer 1: 192.0.2.7 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 77
  Flag: 0x90 Type: 14 Len: 28 Multiprotocol Reachable NLRI:
    Address Family FLOW_IPV4
    NLRI len: 22
      dest_pref  172.31.100.232/32
      src_pref   172.16.15.148/32
      ip_proto   [ == 6 ]
      dest_port  [ >4190 ] and [ <4199 ]
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 6 AS Path:
    Type: 2 Len: 1 < 65530 >
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
  Flag: 0x80 Type: 9 Len: 4 Originator ID: 192.0.2.6
  Flag: 0x80 Type: 10 Len: 4 Cluster ID:
```

```
192.0.2.7
Flag: 0xc0 Type: 16 Len: 8 Extended Community:
rate-limit: 0 kbps
"
```

The route is shown as an MP_REACH_NLRI for address family Flow-IPv4 (AFI 1 SAFI 133). The NLRI uses the source and destination prefixes, the IP protocol, and the destination-port components to describe the flow and create the filter match criteria. The traffic rate extended community is then used to define a rate of 0, which is the filter drop action.

Unlike other address families, there is no strict requirement for the Next-Hop attribute to be present in the MP_REACH_NLRI. The Length of Next-Hop in the Address field can optionally be set to zero and should be ignored on receipt.

The received FlowSpec route can also be verified in the RIB, which provides a concise output of the flow attributes and traffic filtering function, as follows:

```
A:admin@PE-2# /show router bgp routes flow-ipv4
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP FLOW IPV4 Routes
=====
Flag  Network          Nexthop          LocalPref      MED
     As-Path
-----
u*>?  --                0.0.0.0         100            None
     65530

Community Action:  redirect-to-vrf:64496:2
Flowspec Components:
Dest Pref  : 172.31.100.232/32
Src Pref   : 172.16.15.148/32
Ip Proto   : [ == 6 ]
Port       : [ >4190 ] or [ <4199 ]
-----
Routes : 1
=====
```

The dynamically created FlowSpec IPv4 ingress filter is identified as *fSpec-0*, as follows. The origin indicates entry 256 has been added by BGP Flowspec.

```
A:admin@PE-2# /show filter ip "fSpec-0" detail
=====
IP Filter
=====
Filter Id       : fSpec-0
Scope           : Embedded
Type            : Normal
Shared Policer  : Off
Entries         : 1 (insert By Bgp)
Sub-Entries     : 4 (insert By Bgp)
Description     : IPv4 BGP FlowSpec filter for the Base router
-----
```

```

Filter Match Criteria : IP
-----
Entry                : 256
Origin               : Inserted by BGP FlowSpec
Description          : (Not Specified)
Log Id               : n/a
Src. IP              : 172.16.15.148/32
Dest. IP             : 172.31.100.232/32
Port                 : port-list "_tmnx_fSpec_ipv4_14_both"
Protocol             : 6
Dscp                 : Undefined
ICMP Type            : Undefined          ICMP Code      : Undefined
Fragment            : Off                 Src Route Opt  : Off
Sampling             : Off                 Int. Sampling  : On
IP-Option            : 0/0                 Multiple Option: Off
Tcp-flag             : (Not Specified)
Option-pres          : Off
Egress PBR           : Disabled
Primary Action       : Drop
Ing. Matches         : 0 pkts
Egr. Matches         : 0 pkts

-----
Filter Match IP Prefix Lists
-----
No IP Prefix Lists
-----
Filter Match Port Lists
-----
Port list "_tmnx_fSpec_ipv4_14_both"
  0-4198      4191-65535
  NUM ports/ranges: 2

  References:
    IP-filter 104 entry 10256 (Both)
    IP-filter fSpec-0 entry 256 (Both)
    NUM references: 2

NUM Port Lists: 1
-----
Filter Match Protocol Lists
-----
No Protocol Lists
=====

```

The configuration of filter 104 (embedding the *fSpec-0* filter) is as follows, and shows a count of ingress matches, which are dropped. This is verified with the loss of traffic in the direction from T1 to T2, but not in the reverse direction.

```

A:admin@PE-2# /show filter ip 104 detail

=====
IP Filter
=====
Filter Id           : 104                Applied           : Yes
Scope               : Template           Def. Action       : Forward
Type                : Normal
Shared Policer      : Off
System filter       : Unchained
Radius Ins Pt       : n/a
CrCtl. Ins Pt       : n/a
RadSh. Ins Pt       : n/a
PccRl. Ins Pt       : n/a

```

```

Entries          : 0/0/0/1 (Fixed/Radius/Cc/Embedded)
Sub-Entries      : 0/0/0/4
Description      : (Not Specified)
Filter Name      : 104
-----
Filter Match Criteria : IP
-----
Entry            : 10256
Origin           : Inserted by embedded filter fSpec-0 entry 256
Description      : (Not Specified)
Log Id           : n/a
Src. IP          : 172.16.15.148/32
Dest. IP         : 172.31.100.232/32
Port             : port-list "_tmnx_fSpec_ipv4_14_both"
Protocol         : 6
Dscp             : Undefined
ICMP Type        : Undefined          ICMP Code      : Undefined
Fragment         : Off                Src Route Opt  : Off
Sampling         : Off                Int. Sampling  : On
IP-Option        : 0/0                Multiple Option: Off
Tcp-flag         : (Not Specified)
Option-pres     : Off
Egress PBR       : Disabled
Primary Action   : Drop
Ing. Matches     : 0 pkts
Egr. Matches     : 0 pkts
-----
Filter Match IP Prefix Lists
-----
No IP Prefix Lists
-----
Filter Match Port Lists
-----
Port list "_tmnx_fSpec_ipv4_14_both"
  0-4198      4191-65535
  NUM ports/ranges: 2

  References:
    IP-filter 104 entry 10256 (Both)
    IP-filter fSpec-0 entry 256 (Both)
    NUM references: 2

NUM Port Lists: 1
-----
Filter Match Protocol Lists
-----
No Protocol Lists
=====

```

When the route is withdrawn and PE-2 receives an MP_UNREACH_NLRI for the same FlowSpec NLRI, the dynamically created filter entries are removed and all associated hardware resources (TCAM entries) are released.

Instead of dropping traffic at the ingress point to the network, an alternative option is to redirect the mitigated traffic to a traffic-cleansing device, if this infrastructure exists. FlowSpec has the redirect-to-vrf extended community for this purpose, with the process of forwarding traffic toward a scrubbing center frequently referred to as off-ramping. At PE-2, a VPRN is configured to off-ramp traffic toward the scrubbing center connected to PE-5, as shown in the following output.

In the case of FlowSpec, traffic redirection is half-duplex. That is, traffic is forwarded from PE-2 toward PE-5, but not from PE-5 toward PE-2. This is because when the traffic has been cleansed, it re-enters the

network at PE-5 within an IES, and is therefore routed toward its destination using the GRT. This process is frequently referred to as on-ramping. As a result of this half-duplex traffic flow, only a vrf-target import statement is required. There is no requirement to export any routes from PE-2.

```
A:admin@PE-2# admin show configuration /configure service vprn "2"
admin-state enable
description "FlowSpec-OffRamp-VRF"
customer "1"
bgp-ipvpn {
  mpls {
    admin-state enable
    route-distinguisher "64496:2"
    vrf-target {
      import-community "target:64496:2"
    }
    auto-bind-tunnel {
      resolution any
    }
  }
}
```

Off-ramping traffic also requires a VPRN service instance in PE-5 with a single SAP toward the scrubbing center, as shown in the following output. Static IPv4 and IPv4 default routes are configured with next hops of the scrubbing center and these are advertised into VPN-IPv4/VPN-IPv6 using route-policy. There is no requirement for PE-5 to import any BGP-VPN routes.

```
A:admin@PE-5# admin show configuration /configure service vprn "2"
admin-state enable
description "FlowSpec-OffRamp-VRF"
customer "1"
bgp-ipvpn {
  mpls {
    admin-state enable
    route-distinguisher "64496:2"
    vrf-export {
      policy ["vrf2-export"]
    }
  }
}
interface "OffRamp-to-Scrubbing-Center" {
  ipv4 {
    primary {
      address 192.168.2.5
      prefix-length 30
    }
  }
  sap 1/1/c3/1:10 {
  }
  ipv6 {
    address 2001:db8:1b0c:2121::4 {
      prefix-length 127
    }
  }
}
static-routes {
  route 0.0.0.0/0 route-type unicast {
    next-hop "192.168.2.6" {
      admin-state enable
    }
  }
  route ::/0 route-type unicast {
    next-hop "2001:db8:1b0c:2121::5" {

```

```

        admin-state enable
    }
}

```

On-ramping the traffic back onto the network after cleansing the traffic is via IES 3, which is configured as follows. This way the cleansed traffic re-enters the network and is forwarded toward its destination using the GRT.

```

A:admin@PE-5# admin show configuration /configure service ies "3"
admin-state enable
description "FlowSpec-OnRamp-IES"
customer "1"
interface "OnRamp-from-Scrubbing-Center" {
    sap 1/1/c3/1:11 {
    }
    ipv4 {
        primary {
            address 192.168.2.6
            prefix-length 30
        }
    }
    ipv6 {
        address 2001:db8:1b0c:2121::5 {
            prefix-length 127
        }
    }
}

```

To validate the instantiation of the redirection filter, the same bidirectional traffic stream is started between T1 (172.16.15.148) in AS 64511 and T2 (172.31.100.232) in AS 64496. In the T1 to T2 direction, the destination port is TCP port 4191. When the IPv4 FlowSpec route is received at PE-2, the NLRI shows the same traffic match criteria previously used for the black-hole/drop scenario. The extended community has changed to **redirect-to-vrf** with a route-target value of **64496:2**, as shown in the following output.

```

<timestamp> MINOR: DEBUG #2001 Base Peer 1: 192.0.2.7
"Peer 1: 192.0.2.7: UPDATE
Peer 1: 192.0.2.7 - Received BGP UPDATE:
Withdrawn Length = 0
Total Path Attr Length = 77
Flag: 0x90 Type: 14 Len: 28 Multiprotocol Reachable NLRI:
Address Family FLOW_IPV4
NLRI len: 22
  dest_pref  172.31.100.232/32
  src_pref   172.16.15.148/32
  ip_proto   [ == 6 ]
  dest_port  [ >4190 ] and [ <4199 ]
Flag: 0x40 Type: 1 Len: 1 Origin: 0
Flag: 0x40 Type: 2 Len: 6 AS Path:
Type: 2 Len: 1 < 65530 >
Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
Flag: 0x80 Type: 9 Len: 4 Originator ID: 192.0.2.6
Flag: 0x80 Type: 10 Len: 4 Cluster ID:
192.0.2.7
Flag: 0xc0 Type: 16 Len: 8 Extended Community:
redirect-to-vrf:64496:2
"

```


The dynamically created FlowSpec IPv4 ingress filter is identified as *fSpec-0*, as follows. The filter match criteria for entry 256 indicate the primary action is *forward (VRF)*, and the forwarding router/service ID is service ID 2 (the off-ramp VPRN).

```
A:admin@PE-2# /show filter ip "fSpec-0" detail

=====
IP Filter
=====
Filter Id       : fSpec-0
Scope          : Embedded
Type           : Normal
Shared Policer  : Off
Entries        : 1 (insert By Bgp)
Sub-Entries     : 4 (insert By Bgp)
Description     : IPv4 BGP FlowSpec filter for the Base router
-----
Filter Match Criteria : IP
-----
Entry           : 256
Origin          : Inserted by BGP FlowSpec
Description     : (Not Specified)
Log Id          : n/a
Src. IP         : 172.16.15.148/32
Dest. IP        : 172.31.100.232/32
Port            : port-list "_tmnx_fSpec_ipv4_11_both"
Protocol        : 6
Dscp            : Undefined
ICMP Type       : Undefined          ICMP Code      : Undefined
Fragment        : Off                Src Route Opt  : Off
Sampling        : Off                Int. Sampling  : On
IP-Option       : 0/0                Multiple Option: Off
Tcp-flag        : (Not Specified)
Option-pres     : Off
Egress PBR      : Disabled
Primary Action  : Forward (VRF)
  Router        : 2
  Extended Action : None
PBR Down Action : Drop (entry-default)
Ing. Matches    : 4 pkts (328 bytes)
Egr. Matches    : 0 pkts
-----
Filter Match IP Prefix Lists
-----
No IP Prefix Lists
-----
Filter Match Port Lists
-----
Port list "_tmnx_fSpec_ipv4_11_both"
  0-4198      4191-65535
  NUM ports/ranges: 2

  References:
    IP-filter 104 entry 10256 (Both)
    IP-filter fSpec-0 entry 256 (Both)
    NUM references: 2

NUM Port Lists: 1
-----
Filter Match Protocol Lists
-----
No Protocol Lists
```

The configuration of filter 1 (embedding the *fSpec-0* filter) shows a count of ingress matches, and is as follows:

```

A:admin@PE-2# /show filter ip 104

=====
IP Filter
=====
Filter Id       : 104                               Applied       : Yes
Scope          : Template                           Def. Action   : Forward
Type           : Normal
Shared Policer : Off
System filter  : Unchained
Radius Ins Pt  : n/a
CrCtl. Ins Pt  : n/a
RadSh. Ins Pt  : n/a
PccRl. Ins Pt  : n/a
Entries        : 0/0/0/1 (Fixed/Radius/Cc/Embedded)
Sub-Entries    : 0/0/0/4
Description    : (Not Specified)
Filter Name    : 104
-----
Filter Match Criteria : IP
-----
Entry          : 10256
Origin         : Inserted by embedded filter fSpec-0 entry 256
Description    : (Not Specified)
Log Id        : n/a
Src. IP       : 172.16.15.148/32
Dest. IP      : 172.31.100.232/32
Port          : port-list "_tmnx_fSpec_ipv4_12_both"
Protocol      : 6
Dscp          : Undefined
ICMP Type     : Undefined                           ICMP Code    : Undefined
Fragment     : Off                                 Src Route Opt : Off
Sampling     : Off                                 Int. Sampling : On
IP-Option    : 0/0                                 Multiple Option: Off
Tcp-flag     : (Not Specified)
Option-pres  : Off
Egress PBR   : Disabled
Primary Action : Forward (VRF)
  Router      : 2
  Extended Action : None
PBR Down Action : Drop (entry-default)
Ing. Matches   : 4 pkts (328 bytes)
Egr. Matches   : 0 pkts
-----
Filter Match IP Prefix Lists
-----
No IP Prefix Lists
-----
Filter Match Port Lists
-----
Port list "_tmnx_fSpec_ipv4_13_both"
  0-4198      4191-65535
  NUM ports/ranges: 2

References:
  IP-filter 104 entry 10256 (Both)
  IP-filter fSpec-0 entry 256 (Both)

```

```

NUM references: 2
NUM Port Lists: 1
-----
Filter Match Protocol Lists
-----
No Protocol Lists
=====

```

Traffic is correctly received in the T1 to T2 direction, and also in the reverse direction. However, traffic in the T1 to T2 direction is redirected by PE-2 toward the scrubbing center attached to PE-5, before being forwarded to its destination at PE-4.

IPv6 FlowSpec

To validate the instantiation of ingress filters based on IPv6 FlowSpec routes, a bidirectional traffic stream is commenced between T1 (2001:db8:4511:188::177) in AS 64511 and T2 (2001:db8:4496:100::32) in AS 64496. In the T1 to T2 direction, the destination port is TCP port 4191.

An IPv6 FlowSpec route is generated to black-hole/drop traffic with a source address of 2001:db8:4511:188::177 (T1) and a destination address of 2001:db8:4496:100::32 (T2), for any destination ports in the range 4191-4198. The following output shows the route as received at PE-2.

```

<timestamp> MINOR: DEBUG #2001 Base Peer 1: 192.0.2.7
"Peer 1: 192.0.2.7: UPDATE
Peer 1: 192.0.2.7 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 103
  Flag: 0x90 Type: 14 Len: 54 Multiprotocol Reachable NLRI:
    Address Family FLOW_IPV6
    NLRI len: 48
      dest_pref  2001:db8:4496:100::32/128 offset 0
      src_pref   2001:db8:4511:188::177/128 offset 0
      ip_proto   [ == 6 ]
      dest_port  [ >4190 ] and [ <4199 ]
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 6 AS Path:
    Type: 2 Len: 1 < 65530 >
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
  Flag: 0x80 Type: 9 Len: 4 Originator ID: 192.0.2.6
  Flag: 0x80 Type: 10 Len: 4 Cluster ID:
    192.0.2.7
  Flag: 0xc0 Type: 16 Len: 8 Extended Community:
    rate-limit: 0 kbps
"

```

The route is shown as an MP_REACH_NLRI for address family Flow-IPv6 (AFI 2 SAFI 133). As with the FlowSpec IPv4 example, the NLRI uses the source and destination prefixes, the IP protocol, and the destination-port components to describe the flow and create the filter match criteria. The traffic rate extended community is then used to define a rate of 0, which is equivalent to a filter drop action.

The dynamically created FlowSpec IPv6 ingress filter is identified as *fSpec-0*, as follows. The description indicates entry 256 has been added through BGP Flowspec.

```
A:admin@PE-2# show filter ipv6 "fSpec-0" detail
```

```

=====
IPv6 Filter

```

```

=====
Filter Id       : fSpec-0
Scope          : Embedded
Type           : Normal
Shared Policer : Off
Entries        : 1 (insert By Bgp)
Sub-Entries    : 4 (insert By Bgp)
Description     : IPv6 BGP FlowSpec filter for the Base router
-----
Filter Match Criteria : IPv6
-----
Entry          : 256
Origin         : Inserted by BGP FlowSpec
Description    : (Not Specified)
Log Id        : n/a
Src. IP       : 2001:db8:4511:188::177/128
Dest. IP      : 2001:db8:4496:100::32/128
Port          : port-list "_tmnx_fSpec_ipv6_14_both"
Next Header   : 6
Dscp          : Undefined
ICMP Type     : Undefined          ICMP Code      : Undefined
Sampling      : Off                Int. Sampling : On
Tcp-flag      : (Not Specified)
Fragment      : Off
HopByHop Opt  : Off                Routing Type0 : Off
Auth Hdr      : Off                ESP header    : Off
Flow-label    : n/a                Flow-label Mask: n/a
Egress PBR    : Disabled
Primary Action : Drop
Ing. Matches  : 0 pkts
Egr. Matches  : 0 pkts
-----
Filter Match IPv6 Prefix Lists
-----
No IPv6 Prefix Lists
-----
Filter Match Port Lists
-----
Port list "_tmnx_fSpec_ipv6_14_both"
  0-4198      4191-65535
  NUM ports/ranges: 2

  References:
    IPv6-filter 106 entry 10256 (Both)
    IPv6-filter fSpec-0 entry 256 (Both)
    NUM references: 2

NUM Port Lists: 1
-----
Filter Match Protocol Lists
-----
No Protocol Lists
=====

```

The configuration of filter 106 (embedding the *fSpec-0* filter) is as follows, and shows a count of ingress matches, which are dropped (primary action is drop). This is observed with the loss of traffic in the direction from T1 to T2, but not in the reverse direction.

```
A:admin@PE-2# /show filter ipv6 106 detail
```

```
=====
IPv6 Filter
```

```

=====
Filter Id           : 106                               Applied           : Yes
Scope              : Template                          Def. Action       : Forward
Type               : Normal
Shared Policer     : Off
System filter      : Unchained
Radius Ins Pt      : n/a
CrCtl. Ins Pt      : n/a
RadSh. Ins Pt      : n/a
PccRl. Ins Pt      : n/a
Entries            : 0/0/0/1 (Fixed/Radius/Cc/Embedded)
Sub-Entries        : 0/0/0/4
Description        : (Not Specified)
Filter Name        : 106
-----
Filter Match Criteria : IPv6
-----
Entry              : 10256
Origin             : Inserted by embedded filter fSpec-0 entry 256
Description        : (Not Specified)
Log Id            : n/a
Src. IP           : 2001:db8:4511:188::177/128
Dest. IP          : 2001:db8:4496:100::32/128
Port              : port-list "_tmnx_fSpec_ipv6_14_both"
Next Header       : 6
Dscp              : Undefined
ICMP Type         : Undefined                          ICMP Code         : Undefined
Sampling          : Off                               Int. Sampling     : On
Tcp-flag          : (Not Specified)
Fragment          : Off
HopByHop Opt      : Off                               Routing Type0     : Off
Auth Hdr          : Off                               ESP header        : Off
Flow-label        : n/a                               Flow-label Mask   : n/a
Egress PBR        : Disabled
Primary Action     : Drop
Ing. Matches      : 0 pkts
Egr. Matches      : 0 pkts
-----
Filter Match IPv6 Prefix Lists
-----
No IPv6 Prefix Lists
-----
Filter Match Port Lists
-----
Port list "_tmnx_fSpec_ipv6_14_both"
  0-4198          4191-65535
  NUM ports/ranges: 2

  References:
    IPv6-filter 106 entry 10256 (Both)
    IPv6-filter fSpec-0 entry 256 (Both)
    NUM references: 2

NUM Port Lists: 1
-----
Filter Match Protocol Lists
-----
No Protocol Lists
=====

```

The FlowSpec IPv6 route with the drop action is subsequently withdrawn, restoring the traffic flow between T1 and T2.

To off-ramp IPv6 traffic toward the scrubbing center, the same redirect infrastructure is used as in the IPv4 example:

- PE-2 and PE-5 use the same off-ramp VPRN (VPRN 2), which transports both VPN-IPv4 and VPN-IPv6 traffic.
- PE-5 uses the same on-ramp (IES). When traffic is returned from the scrubbing center, PE-5 routes packets toward their destination using the GRT.

An IPv6 FlowSpec route with a **redirect-to-vrf** extended community is then sourced by the FlowSpec route generator. When the route is received at PE-2, the NLRI shows the same traffic match criteria previously used for the IPv6 black-hole/drop scenario. The extended community has changed to **redirect-to-vrf** with a route-target value of **64496:2**, as shown in the following output.

```
<timestamp> CEST MINOR: DEBUG #2001 Base Peer 1: 192.0.2.7
"Peer 1: 192.0.2.7: UPDATE
Peer 1: 192.0.2.7 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 103
  Flag: 0x90 Type: 14 Len: 54 Multiprotocol Reachable NLRI:
    Address Family FLOW_IPV6
    NLRI len: 48
      dest_pref  2001:db8:4496:100::32/128 offset 0
      src_pref   2001:db8:4511:188::177/128 offset 0
      ip_proto   [ == 6 ]
      dest_port  [ >4190 ] and [ <4199 ]
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 6 AS Path:
    Type: 2 Len: 1 < 65530 >
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
  Flag: 0x80 Type: 9 Len: 4 Originator ID: 192.0.2.6
  Flag: 0x80 Type: 10 Len: 4 Cluster ID:
    192.0.2.7
  Flag: 0xc0 Type: 16 Len: 8 Extended Community:
    redirect-to-vrf:64496:2
"
```

The dynamically created FlowSpec IPv4 ingress filter is identified as *fSpec-0*, as follows. The filter match criteria for entry 256 indicate the primary action is *forward (VRF)*, and the forwarding router/service ID is service ID 2 (the off-ramp VPRN).

```
A:admin@PE-2# /show filter ipv6 "fSpec-0" detail

=====
IPv6 Filter
=====
Filter Id       : fSpec-0
Scope          : Embedded
Type           : Normal
Shared Policer  : Off
Entries        : 1 (insert By Bgp)
Sub-Entries    : 4 (insert By Bgp)
Description     : IPv6 BGP FlowSpec filter for the Base router
-----
Filter Match Criteria : IPv6
-----
Entry          : 256
Origin         : Inserted by BGP FlowSpec
Description    : (Not Specified)
Log Id        : n/a
Src. IP       : 2001:db8:4511:188::177/128
Dest. IP      : 2001:db8:4496:100::32/128
```

```

Port          : port-list "_tmnx_fSpec_ipv6_15_both"
Next Header   : 6
Dscp          : Undefined
ICMP Type     : Undefined          ICMP Code      : Undefined
Sampling      : Off                Int. Sampling  : On
Tcp-flag      : (Not Specified)
Fragment      : Off
HopByHop Opt  : Off                Routing Type0  : Off
Auth Hdr      : Off                ESP header     : Off
Flow-label    : n/a                Flow-label Mask: n/a
Egress PBR    : Disabled
Primary Action : Forward (VRF)
  Router       : 2
  Extended Action : None
PBR Down Action : Drop (entry-default)
Ing. Matches   : 0 pkts
Egr. Matches   : 0 pkts

-----
Filter Match IPv6 Prefix Lists
-----
No IPv6 Prefix Lists
-----
Filter Match Port Lists
-----
Port list "_tmnx_fSpec_ipv6_15_both"
  0-4198      4191-65535
  NUM ports/ranges: 2

  References:
    IPv6-filter 106 entry 10256 (Both)
    IPv6-filter fSpec-0 entry 256 (Both)
    NUM references: 2

NUM Port Lists: 1
-----
Filter Match Protocol Lists
-----
No Protocol Lists
=====

```

The configuration of IPv6 filter 106 (embedding the *fSpec-0* filter) shows a count of ingress matches, and is as follows:

```

A:admin@PE-2# /show filter ipv6 106 detail

=====
IPv6 Filter
=====
Filter Id      : 106                Applied       : Yes
Scope         : Template           Def. Action   : Forward
Type          : Normal
Shared Policer : Off
System filter  : Unchained
Radius Ins Pt : n/a
CrCtl. Ins Pt : n/a
RadSh. Ins Pt : n/a
PccRl. Ins Pt : n/a
Entries       : 0/0/0/1 (Fixed/Radius/Cc/Embedded)
Sub-Entries   : 0/0/0/4
Description   : (Not Specified)
Filter Name   : 106
-----

```

```

Filter Match Criteria : IPv6
-----
Entry                : 10256
Origin               : Inserted by embedded filter fSpec-0 entry 256
Description          : (Not Specified)
Log Id               : n/a
Src. IP              : 2001:db8:4511:188::177/128
Dest. IP             : 2001:db8:4496:100::32/128
Port                 : port-list "_tmnx_fSpec_ipv6_15_both"
Next Header          : 6
Dscp                 : Undefined
ICMP Type            : Undefined                ICMP Code       : Undefined
Sampling             : Off                    Int. Sampling    : On
Tcp-flag             : (Not Specified)
Fragment             : Off
HopByHop Opt         : Off                    Routing Type0    : Off
Auth Hdr             : Off                    ESP header       : Off
Flow-label           : n/a                    Flow-label Mask  : n/a
Egress PBR           : Disabled
Primary Action       : Forwarded (VRF)
  Router              : 2
  Extended Action     : None
PBR Down Action      : Drop (entry-default)
Ing. Matches         : 799 pkts (102272 bytes)
Egr. Matches         : 0 pkts
-----
Filter Match IPv6 Prefix Lists
-----
No IPv6 Prefix Lists
-----
Filter Match Port Lists
-----
Port list "_tmnx_fSpec_ipv6_15_both"
  0-4198      4191-65535
  NUM ports/ranges: 2

References:
  IPv6-filter 106 entry 10256 (Both)
  IPv6-filter fSpec-0 entry 256 (Both)
  NUM references: 2

NUM Port Lists: 1
-----
Filter Match Protocol Lists
-----
No Protocol Lists
=====

```

Traffic is correctly received in the T1 to T2 direction, and also in the reverse direction. However, traffic in the T1 to T2 direction is redirected by PE-2 toward the scrubbing center attached to PE-5, before being forwarded to its destination at PE-4.

Resource consumption

Similar to static filters consuming hardware resources, dynamically instantiated FlowSpec filters consume hardware resources (TCAM entries) on the associated line cards. Therefore, resources must be checked and monitored to ensure that the system operates within its scaling boundaries.

Before the activation of any FlowSpec routes, there are two ingress ACL/QoS entries consumed for IPv4 and another two entries for IPv6, as shown in the following output.

```
A:admin@PE-2# /tools dump resource-usage system all | match 'Usage|Free|ACL Entries'
Resource Usage Information for System
                                Total  Allocated  Free
Resource Usage Information for Card Slot #1
                                Total  Allocated  Free
Resource Usage Information for Card Slot #1 FP #1
                                Total  Allocated  Free
      Ingress ACL Entries (IPv4/v6) | 98304      2  98302
      Egress ACL Entries (IPv4/v6) | 49152      2  49150
Resource Usage Information for Card Slot #1 MDA #1
                                Total  Allocated  Free
Resource Usage Information for Card Slot #1 MDA #2
                                Total  Allocated  Free
```

When a FlowSpec IPv4 rule matching on a source/destination IP address is dynamically instantiated, one additional ACL entry is consumed in hardware, as shown in the following output.

```
A:admin@PE-2# /tools dump resource-usage system all | match 'Usage|Free|ACL Entries'
Resource Usage Information for System
                                Total  Allocated  Free
Resource Usage Information for Card Slot #1
                                Total  Allocated  Free
Resource Usage Information for Card Slot #1 FP #1
                                Total  Allocated  Free
      Ingress ACL Entries (IPv4/v6) | 98304      3  98301
      Egress ACL Entries (IPv4/v6) | 49152      2  49150
Resource Usage Information for Card Slot #1 MDA #1
                                Total  Allocated  Free
Resource Usage Information for Card Slot #1 MDA #2
                                Total  Allocated  Free
```

TCAM entries are not consumed on a per-interface basis. When TCAM entries are consumed on a line card for a FlowSpec NLRI match criteria, the same criteria can be used for filtering across multiple IP interfaces on the same line card without consuming additional TCAM entries.

Conclusion

FlowSpec IPv4 and IPv6 provide a dynamic way to activate (and tear down) ingress filters to mitigate against DDoS attacks. SR OS supports a wide range of match criteria (FlowSpec NLRI) coupled with the ability to either drop or redirect mitigated traffic. This offers flexibility not only in what traffic is matched, but also in traffic treatment, depending on the availability of a traffic-cleansing infrastructure.

The ability of FlowSpec to dynamically create and remove filters has some immediate benefits:

- Reduces the likelihood of configuration errors on one or more devices
- Allows for temporary use of hardware resources, which are released when the threat has passed
- Allows for a push configuration from a single point to a potentially large number of network devices, without having to visit each one to configure filters manually.

BGP Multipath

This chapter provides information about BGP Multipath.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 14.0.R4, but the MD-CLI in the current edition is based on SR OS Release 21.10.R1. Configurable BGP multipath parameters per address family and selective BGP multipath is supported in SR OS Release 19.5.R1, and later.

Overview

When BGP multipath is enabled, traffic can be forwarded to an IP prefix destination over multiple BGP paths that are considered equal by the BGP decision process. BGP multipath is supported in base router and VPRNs, both for iBGP and eBGP. The **multipath** command specifies the maximum number of BGP paths that each BGP RIB can submit to the route table for an IP prefix. The equal cost multipath (ECMP) limit defines how many paths are selected for installation in the forwarding information base (FIB). Traffic in the data path that matches the IP prefix is load-balanced across the ECMP next hops on a per-packet hash calculation.



Note:

As described in chapter [Separate BGP RIBs for Labeled Routes](#), labeled routes and unlabeled routes do not mix.

BGP multipath can be configured as follows:

1. The **multipath** commands present in the base router and VPRN **bgp** contexts are configurable on a global level or more specific, within an address family context (**ipv4**, **ipv6**, **label-ipv4**, and **label-ipv6**). Following parameters are possible:

```
[ex:configure router "Base" bgp multipath]
A:admin@PE-5#          multipath ?

multipath

ebgp                    - Maximum multipaths per prefix for EBGp learned routes
family                 + Enter the family list instance
ibgp                   - Maximum multipaths per prefix for IBGP learned routes
max-paths              - Maximum multipaths per prefix
restrict               - AS path restriction for the non-best path
unequal-cost           - Ignore differences in the next-hop cost for multipath
```

```
[ex:/configure router "Base" bgp multipath]
```

```
A:admin@PE-5# max-paths ?

max-paths <number>
<number> - <1..64>
Default   - 1

Maximum multipaths per prefix

[ex:/configure router "Base" bgp multipath]
A:admin@PE-5# restrict ?

restrict <keyword>
<keyword> - (same-as-path-length|same-neighbor-as|exact-as-path)
Default   - same-as-path-length

AS path restriction for the non-best path
```

- multipath configuration per address family overrules the generic max-paths configuration.
 - *max-paths* is the default maximum number of paths. It is overruled by *ebgp-max-paths* and *ibgp-max-paths*. However, if there is no maximum set for the number of eBGP paths or iBGP paths, then the maximum number of paths is set by *max-paths*.
 - *ebgp-max-paths* specifies the maximum number of paths that can be used when the best path is eBGP. If configured, *ebgp-max-paths* overrides the configured *max-paths* for eBGP paths.
 - *ibgp-max-paths* specifies the maximum number of paths that can be used when the best path is iBGP. If configured, *ibgp-max-paths* overrides the configured *max-paths* for iBGP paths.
 - **restrict same-neighbor-as** forces multipaths to have the same (shortest) AS path length (unless **as-path-ignore** is configured) and, for the paths with that length, the same neighbor AS.
 - **restrict exact-as-path** forces multipaths to have the exact same AS paths.
 - **unequal-cost** allows to use routes with different next-hop costs in multipath ECMP sets.
2. The **ebgp-ibgp-equal** command is added to the **best-path-selection** contexts in base router and VPRN **bgp** contexts. When this command is configured, as follows, the BGP decision process skips the step that prefers eBGP over iBGP. This enables load-balancing between eBGP and iBGP paths.

```
[ex:/configure router "Base" bgp]
A:admin@PE-5# best-path-selection ?

best-path-selection

always-compare-med + Enter the always-compare-med context
as-path-ignore     + Enter the as-path-ignore context
compare-origin-validation-state - Compare RPKI origin validation state of usable routes
d-path-length-ignore - Enable D-PATH length ignore
deterministic-med - Group paths based on AS before MED attribute comparison
ebgp-ibgp-equal + Enter the ebgp-ibgp-equal context
ignore-nh-metric - Ignore next-hop distance in best path selection
ignore-router-id + Enable the ignore-router-id context
origin-invalid-unusable - Deem invalid routes unusable for best-path selection
```



Note:

The *ebgp-ibgp-equal* command is not to be confused with the *eibgp-loadbalance* command in a VPRN, that is used to provide ECMP over BGP-VPN (imported routes) and BGP routes. It is called *eibgp-loadbalance* because, in such scenarios, BGP-VPN is typically used between iBGP

peers and BGP is used between eBGP peers. However, this is not always the case, so the name can be misleading.

Entire BGP groups or a selection of BGP neighbors can be configured as multipath-eligible. If a route is learned for an IPv4, IPv6, label-IPv4, or label-IPv6 prefix, and the associated maximum number of paths is N (which can depend on the address family and whether the best path was received from an eBGP or iBGP peer), then one of the following three rules applies:

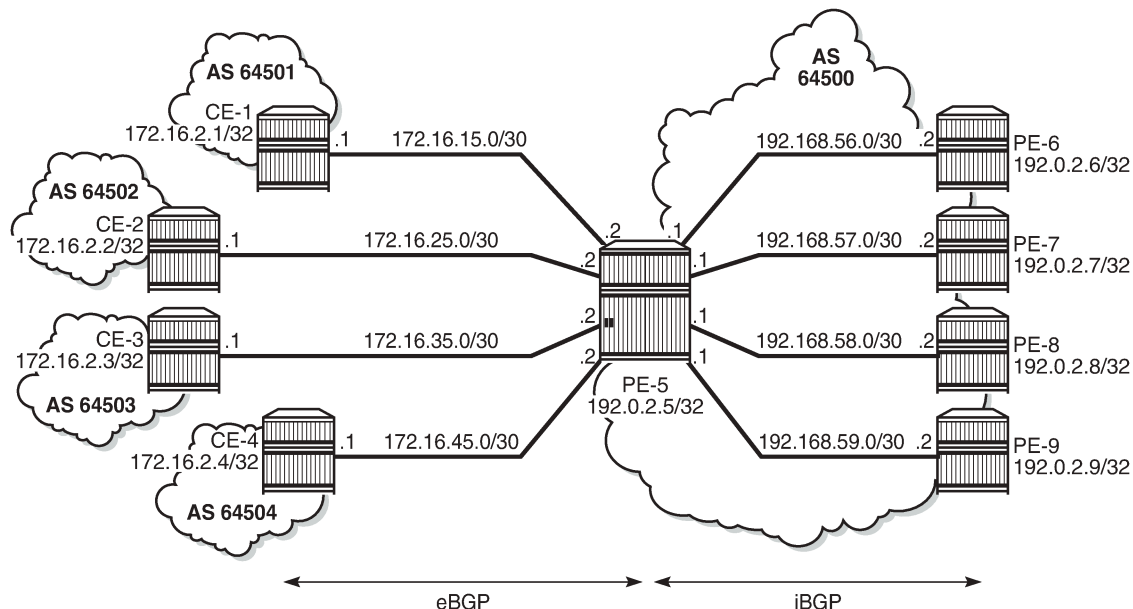
- If the best path came from a neighbor marked as **multipath-eligible**, then only paths marked as multipath-eligible are candidates for the BGP multipath and the best N are chosen for installation as ECMP next-hops.
- If none of the paths from the set of all possible multipaths came from a neighbor marked as **multipath-eligible**, the best N are chosen.
- If the best path did not come from a neighbor marked as **multipath-eligible** and at least one path from the set of all possible multipaths came from a multipath-eligible peer, then only the best path is chosen and all other paths are eliminated.

Configuration

The examples in this section show the multipath BGP configuration in the base router. For BGP multipath in a VPRN, the configuration is similar.

[Figure 78: Example topology](#) shows the example configuration with the used IP addresses. PE-5 has eBGP sessions with CEs in different autonomous systems (ASs) and iBGP sessions with PE-6, PE-7, PE-8, and PE-9.

Figure 78: Example topology



26053

The initial configuration includes:

- Cards, MDAs, ports
- Router interfaces
- IS-IS in AS 64500
- LDP in AS 64500
- BGP on all nodes (eBGP between CEs and PE-5; iBGP between PEs)
- Export policy "export-bgp" accepting routes from protocol direct on all nodes

The BGP configuration on CE-1 is as follows:

```
# on CE-1:
configure {
  router "Base" {
    autonomous-system 64501
    bgp {
      split-horizon true
      ebgp-default-reject-policy {
        import false
        export false
      }
      group "eBGP" {
        peer-as 64500
        export {
          policy ["export-bgp"]
        }
      }
      neighbor "172.16.15.2" {
        group "eBGP"
      }
    }
  }
}
```

The BGP configuration on the other nodes that advertise routes to PE-5 is similar.

The BGP configuration on PE-5 is as follows:

```
# on PE-5:
configure {
  router "Base" {
    autonomous-system 64500
    bgp {
      split-horizon true
      group "eBGP" {
        ebgp-default-reject-policy {
          import false
          export false
        }
      }
      group "iBGP" {
        peer-as 64500
      }
      neighbor "172.16.15.1" {
        group "eBGP"
        peer-as 64501
      }
      neighbor "172.16.25.1" {
        group "eBGP"
        peer-as 64502
      }
      neighbor "172.16.35.1" {
        group "eBGP"
        peer-as 64503
      }
    }
  }
}
```

```
    }
    neighbor "172.16.45.1" {
        group "eBGP"
        peer-as 64504
    }
    neighbor "192.0.2.6" {
        group "iBGP"
    }
    neighbor "192.0.2.7" {
        group "iBGP"
    }
    neighbor "192.0.2.8" {
        group "iBGP"
    }
    neighbor "192.0.2.9" {
        group "iBGP"
    }
}
```

The following will be configured and verified:

- BGP multipath with different eBGP and iBGP path limits
- BGP multipath with equal eBGP and iBGP path treatment
- BGP multipath restricted to the same neighbor AS
- BGP multipath restricted to the exact AS path
- BGP multipath per address family
- Selective BGP multipath

BGP multipath with different eBGP and iBGP path limits

On PE-5, BGP multipath is configured as follows:

```
# on PE-5:
configure {
    router "Base" {
        bgp {
            multipath {
                max-paths 8
                ebgp 2
                ibgp 3
            }
        }
    }
}
```

The **max-paths** limit is only used when no limits are configured for eBGP or iBGP. It is allowed to specify a lower value for **max-paths** than for either eBGP or iBGP because the configured number of paths for eBGP and iBGP overrule the max-paths limitation, as follows:

```
# on PE-5:
configure {
    router "Base" {
        bgp {
            multipath {
                max-paths 1
                ebgp 2
                ibgp 3
            }
        }
    }
}
```

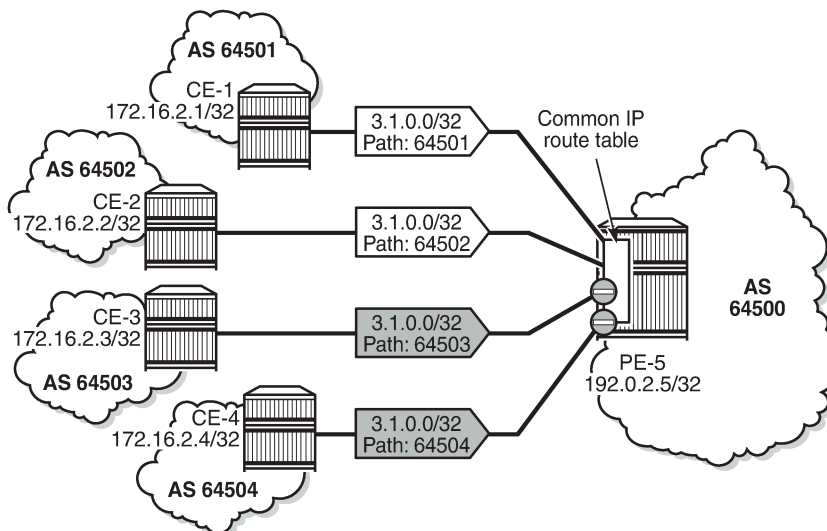
With this configuration, regardless of the value of max-paths, there can be two eBGP routes for the same prefix and three iBGP routes for the same prefix. If the best route is eBGP, the *ebgp-max-paths* value is 2; if the best route is iBGP, the *ibgp-max-paths* value is 3. The value for max-paths (1) is never used when limits for both eBGP and iBGP are configured.

```
# on PE-5:
configure {
  router "Base" {
    bgp {
      multipath {
        max-paths 3
        ebgp 2
      }
    }
  }
}
```

With this configuration, there can be two eBGP routes for the same prefix and three iBGP routes for the same prefix. If the best route is eBGP, the *ebgp-max-paths* value is 2, and if the best route is iBGP, the *max-paths* value is 3.

In the following example, all four eBGP neighbors advertise prefix 3.1.0.0/32 to PE-5 and all four iBGP neighbors advertise prefix 3.2.0.0/32 to PE-5. PE-5 receives four eBGP routes for prefix 3.1.0.0/32, but only two are added to the common IP route table, as shown in [Figure 79: BGP multipath with eBGP limit 2](#).

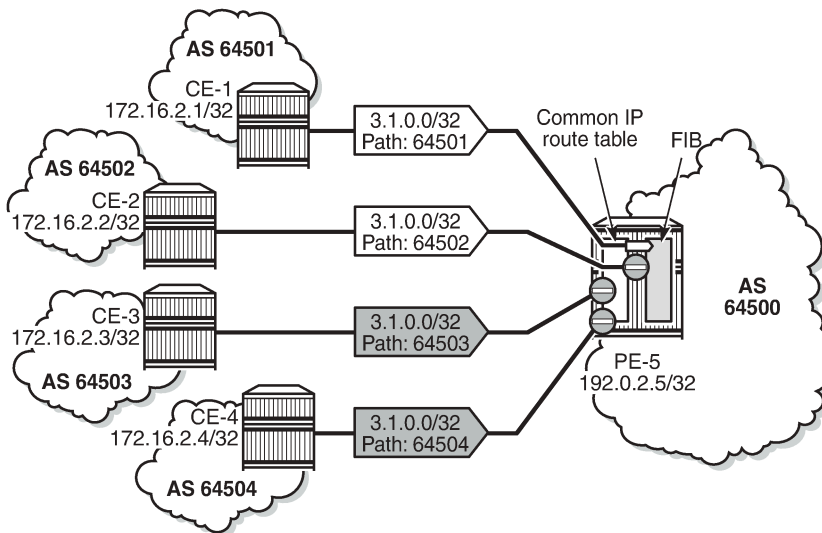
Figure 79: BGP multipath with eBGP limit 2



26054

These routes can only be added to the FIB if ECMP is configured to a value at least equal to the number of routes allowed in BGP multipath. By default, ECMP is disabled and only one route is added to the FIB, as shown in [Figure 80: eBGP multipath with limit 2 and ECMP disabled](#).

Figure 80: eBGP multipath with limit 2 and ECMP disabled



26055

With ECMP disabled, only one of the four paths is used for prefix 3.1.0.0/32, as follows:

```
[/]
A:admin@PE-5# show router bgp routes 3.1.0.0/32
=====
BGP Router ID:192.0.2.5      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  3.1.0.0/32                None       None
      172.16.15.1            None       0
      64501                  -
*i    3.1.0.0/32                None       None
      172.16.25.1           None       0
      64502                  -
*i    3.1.0.0/32                None       None
      172.16.35.1           None       0
      64503                  -
*i    3.1.0.0/32                None       None
      172.16.45.1           None       0
      64504                  -
-----
Routes : 4
=====
```


In the remainder of the chapter, ECMP is configured with a value of eight, implying that the routes added to the common IP route table will be added to the FIB as well. ECMP is configured on PE-5 as follows:

```
# on PE-5:
configure {
  router "Base" {
    ecmp 8
  }
}
```

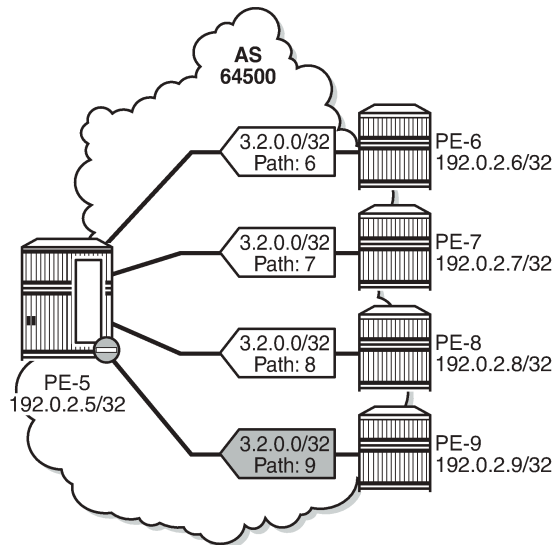
With ECMP configured with a limit of eight, two eBGP paths are used for prefix 3.1.0.0/32.

The first two of the following BGP routes for prefix 3.1.0.0/32 are used on PE-5:

```
[/]
A:admin@PE-5# show router bgp routes 3.1.0.0/32
=====
BGP Router ID:192.0.2.5      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  3.1.0.0/32                               None       None
      172.16.15.1                             None       0
      64501                                     -
u*>i  3.1.0.0/32                               None       None
      172.16.25.1                             None       0
      64502                                     -
*>i   3.1.0.0/32                               None       None
      172.16.35.1                             None       0
      64503                                     -
*>i   3.1.0.0/32                               None       None
      172.16.45.1                             None       0
      64504                                     -
-----
Routes : 4
=====
```

The four iBGP neighbors of PE-5 advertise prefix 3.2.0.0/32 to PE-5. BGP multipath has a limit of three for iBGP routes. Consequently, three BGP routes are added to the common IP route table and to the FIB, as shown in [Figure 81: BGP multipath with iBGP limit 3 and ECMP limit 8](#).

Figure 81: BGP multipath with iBGP limit 3 and ECMP limit 8



26056

Three iBGP paths are used for prefix 3.2.0.0/32, as follows:

```
[/]
A:admin@PE-5# show router bgp routes 3.2.0.0/32
=====
BGP Router ID:192.0.2.5      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  3.2.0.0/32                100        None
      192.0.2.6                None        10
      6
u*>i  3.2.0.0/32                100        None
      192.0.2.7                None        10
      7
u*>i  3.2.0.0/32                100        None
      192.0.2.8                None        10
      8
*>i   3.2.0.0/32                100        None
      192.0.2.9                None        10
      9
-----
Routes : 4
=====
```

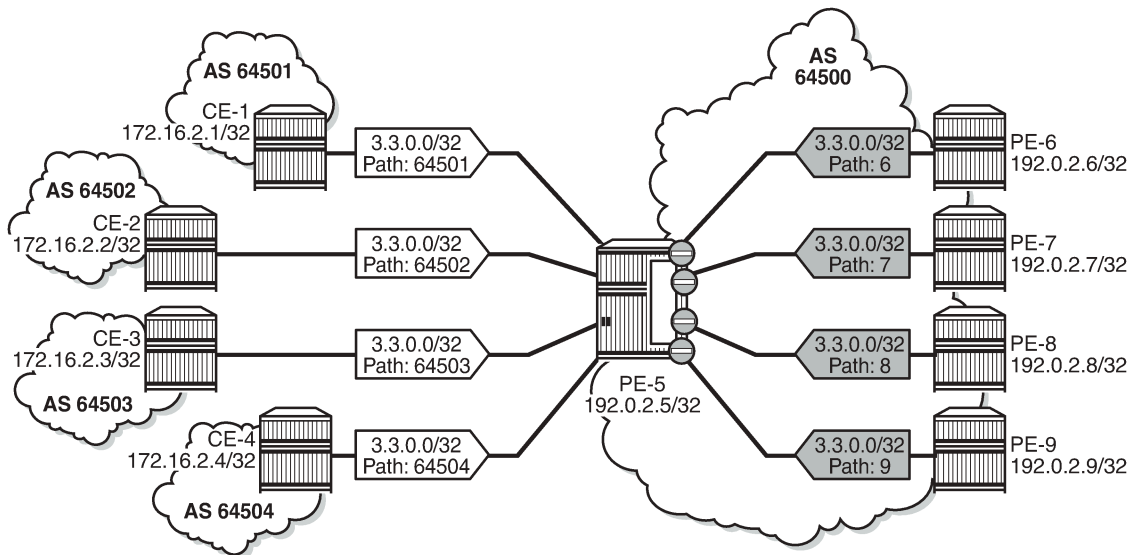
BGP multipath with equal eBGP and iBGP path treatment

It is optional to specify limits for eBGP and iBGP; an overall multipath limitation is sufficient, such as:

```
# on PE-5:
configure {
  router "Base" {
    bgp {
      multipath {
        max-paths 6
      }
    }
  }
}
```

With this configuration, there can be six routes for the same prefix. These routes can be eBGP or iBGP routes. By default, eBGP routes are preferred and, therefore, only the four eBGP routes are imported in the common IP route table, as shown in [Figure 82: BGP multipath with limit 6 and eBGP preferred](#).

Figure 82: BGP multipath with limit 6 and eBGP preferred



26057

Only the four eBGP paths are used, as follows:

```
[/]
A:admin@PE-5# show router bgp routes 3.3.0.0/32
=====
BGP Router ID:192.0.2.5      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)        Path-Id    IGP Cost
      As-Path                 Label
```

```

-----
u*>i 3.3.0.0/32          None      None
      172.16.15.1       None      0
      64501             -
u*>i 3.3.0.0/32          None      None
      172.16.25.1       None      0
      64502             -
u*>i 3.3.0.0/32          None      None
      172.16.35.1       None      0
      64503             -
u*>i 3.3.0.0/32          None      None
      172.16.45.1       None      0
      64504             -
*i   3.3.0.0/32          100      None
      192.0.2.6         None      10
      6                 -
*i   3.3.0.0/32          100      None
      192.0.2.7         None      10
      7                 -
*i   3.3.0.0/32          100      None
      192.0.2.8         None      10
      8                 -
*i   3.3.0.0/32          100      None
      192.0.2.9         None      10
      9                 -
-----
Routes : 8
=====

```

The BGP decision process prefers eBGP over iBGP, but this step can be skipped by configuring the following:

```

# on PE-5:
configure {
  router "Base" {
    bgp {
      best-path-selection {
        ebgp-ibgp-equal {
          ipv4 true
        }
      }
    }
  }
}

```

This configuration only skips one step in the BGP decision process. If the best route is still eBGP, the eBGP multipath limit applies; if the best route is iBGP, the iBGP multipath limit applies.

Optionally, other best path selection criteria can also be configured, such as ignore-nh-metric. However, the multipath configuration can also be configured with the unequal-cost option. This allows to ignore the next-hop cost in BGP multipath ECMP sets, while preserving the next-hop option in the BGP decision process.

```

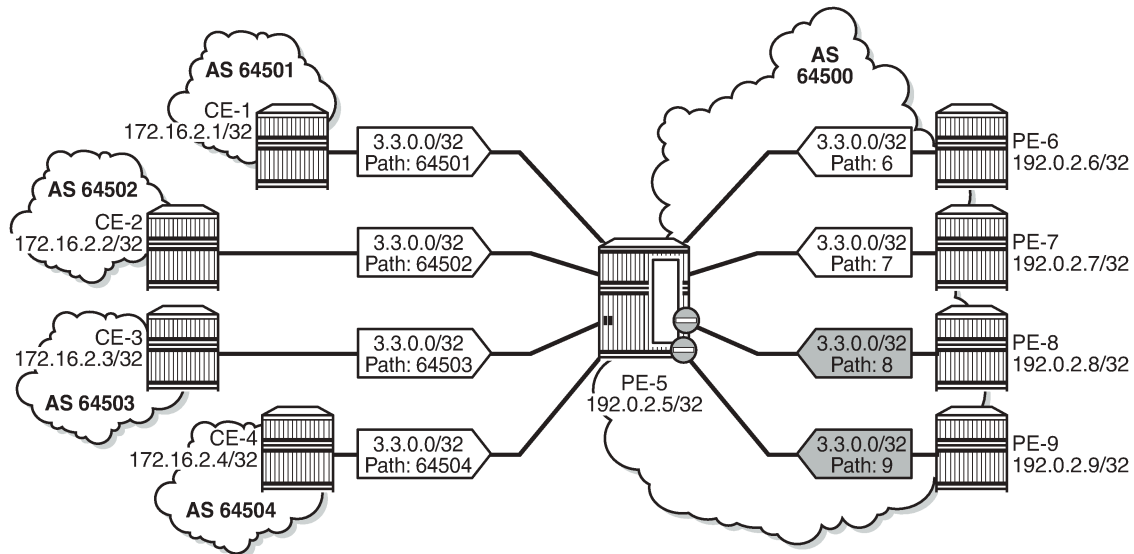
# on PE-5:
configure {
  router "Base" {
    bgp {
      multipath
      max-paths 6
      unequal-cost true
    }
  }
}

```

When all other path options are identical (such as local preference, MED, IGP cost, and other criteria from the BGP decision process), or when the best-path-selection is configured to ignore specific path options,

and the only differentiator is an originator ID, the remaining steps in the BGP decision process do not exclude any routes. In that case, six of the eight eligible BGP paths are included in the BGP multipath, as shown in [Figure 83: BGP multipath with limit 6, eBGP equal to iBGP, and other path options identical](#).

Figure 83: BGP multipath with limit 6, eBGP equal to iBGP, and other path options identical



26058

From the eight advertised BGP routes for prefix 3.3.0.0/32, six paths are used, as follows:

```
[/]
A:admin@PE-5# show router bgp routes 3.3.0.0/32
=====
BGP Router ID:192.0.2.5      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
  Nexthop (Router)                         Path-Id   IGP Cost
  As-Path                                    Label
-----
u*>i 3.3.0.0/32                             None     None
      172.16.15.1                            None     0
      64501
u*>i 3.3.0.0/32                             None     None
      172.16.25.1                            None     0
      64502
u*>i 3.3.0.0/32                             None     None
      172.16.35.1                            None     0
      64503
u*>i 3.3.0.0/32                             None     None
      172.16.45.1                            None     0
      64504
u*>i 3.3.0.0/32                             100     None
```

	192.0.2.6	None	10
	6		-
u*>i	3.3.0.0/32	100	None
	192.0.2.7	None	10
	7		-
*>i	3.3.0.0/32	100	None
	192.0.2.8	None	10
	8		-
*>i	3.3.0.0/32	100	None
	192.0.2.9	None	10
	9		-

Routes : 8			
=====			

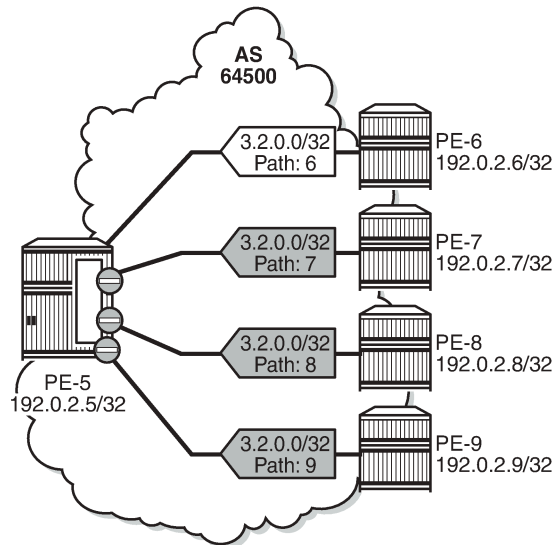
BGP multipath restricted to the same neighbor AS

BGP multipath can be configured with the restriction that the neighbor AS must be the same for all the used paths. When all routes have a different neighbor AS, only one path is used. This can be shown for prefix 3.2.0.0/32 that is advertised by the iBGP neighbors. The BGP multipath configuration on PE-5 is as follows:

```
# on PE-5:
configure {
  router "Base" {
    bgp {
      multipath {
        max-paths 8
        ebgp 2
        ibgp 3
        restrict same-neighbor-as
      }
    }
  }
}
```

Figure 84: BGP multipath configured with restriction to the same neighbor AS shows that with the restriction to the same neighbor AS, only one path is used because all BGP routes have a different neighbor AS.

Figure 84: BGP multipath configured with restriction to the same neighbor AS



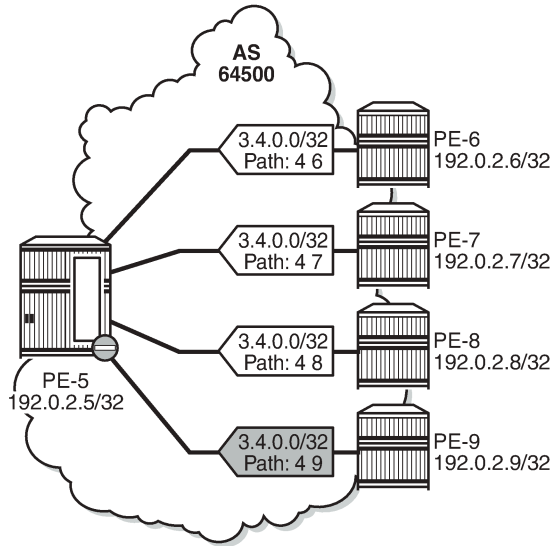
26059

Only one BGP path is used, because all the other routes have a different neighbor AS, as follows:

```
[/]
A:admin@PE-5# show router bgp routes 3.2.0.0/32
=====
BGP Router ID:192.0.2.5      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i 3.2.0.0/32                100        None
      192.0.2.6                None        10
      6
*>i  3.2.0.0/32                100        None
      192.0.2.7                None        10
      7
*>i  3.2.0.0/32                100        None
      192.0.2.8                None        10
      8
*>i  3.2.0.0/32                100        None
      192.0.2.9                None        10
      9
-----
Routes : 4
=====
```

Figure 85: BGP multipath restricted to the same neighbor AS: AS paths with same length shows that the iBGP neighbors also advertise prefix 3.4.0.0/32 with a different AS path, but the AS path is equally long and the neighbor AS is the same. Three of these BGP paths are used.

Figure 85: BGP multipath restricted to the same neighbor AS: AS paths with same length



26060

All iBGP neighbors have the same neighbor AS and an AS path of equal length. Three of the iBGP paths for prefix 3.4.0.0/32 are used, as follows:

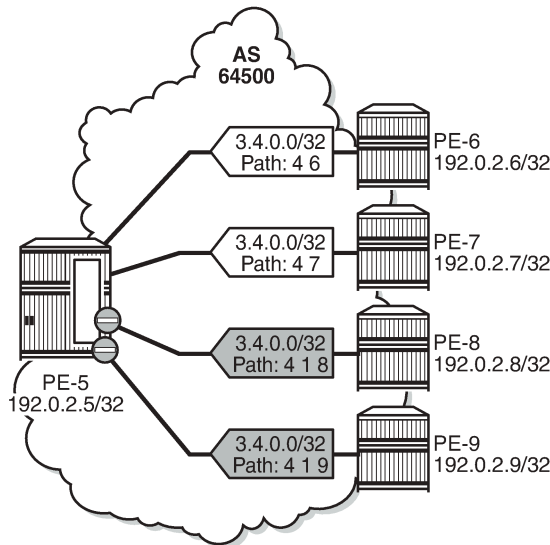
```
[/]
A:admin@PE-5# show router bgp routes 3.4.0.0/32
=====
BGP Router ID:192.0.2.5      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
```

Flag	Network Nexthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
u*>i	3.4.0.0/32 192.0.2.6 4 6	100 None	None 10 -
u*>i	3.4.0.0/32 192.0.2.7 4 7	100 None	None 10 -
u*>i	3.4.0.0/32 192.0.2.8 4 8	100 None	None 10 -
*>i	3.4.0.0/32 192.0.2.9 4 9	100 None	None 10 -


```
-----
Routes : 4
=====
```

The restriction that the neighbor AS must be the same does not overrule the BGP selection criterion that the shorter AS path is preferred. When the AS path is longer for the routes advertised by neighbors 192.0.2.8 and 192.0.2.9, only the BGP paths with the shorter AS path are used, as shown in [Figure 86: BGP multipath restricted to the same neighbor AS: AS paths of different lengths](#).

Figure 86: BGP multipath restricted to the same neighbor AS: AS paths of different lengths



26061

All BGP routes advertised by the iBGP neighbors have the same neighbor AS, but the AS path is longer for neighbors 192.0.2.8 and 192.0.2.9. The routes advertised by these neighbors will not be selected as best path and will not be added to the route table. Only the two BGP routes with the shorter AS path are used, as follows:

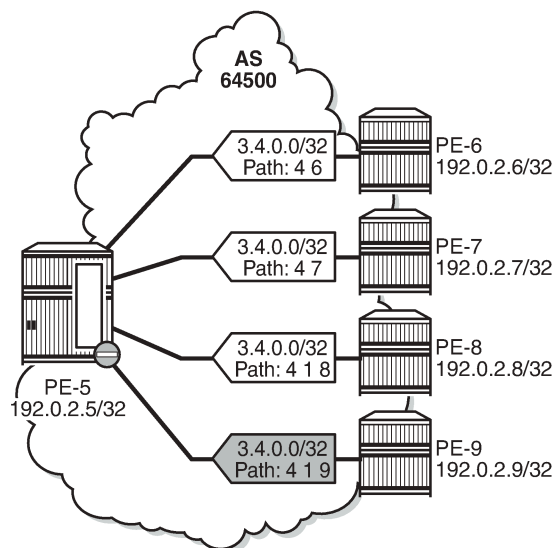
```
[/]
A:admin@PE-5# show router bgp routes 3.4.0.0/32
=====
BGP Router ID:192.0.2.5      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i 3.4.0.0/32                100        None
      192.0.2.6              None        10
      4 6
u*>i 3.4.0.0/32                100        None
```

	192.0.2.7	None	10
	4 7		-
*i	3.4.0.0/32	100	None
	192.0.2.8	None	10
	4 1 8		-
*i	3.4.0.0/32	100	None
	192.0.2.9	None	10
	4 1 9		-

Routes : 4			
=====			

When the best path selection is configured to ignore the AS path, three paths are used again, as shown in [Figure 87: BGP multipath restricted to the same neighbor AS: AS paths of different lengths, AS path ignored](#).

Figure 87: BGP multipath restricted to the same neighbor AS: AS paths of different lengths, AS path ignored



26062

The best path selection is reconfigured as follows:

```
# on PE-5:
configure {
  router "Base" {
    bgp {
      best-path-selection {
        as-path-ignore {
          ipv4 true
        }
      }
    }
  }
}
```

Three of the four eligible BGP routes are used, as follows:

```
[/]
A:admin@PE-5# show router bgp routes 3.4.0.0/32
=====
BGP Router ID:192.0.2.5      AS:64500      Local AS:64500
```

```

=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====

BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
  Nexthop (Router)                         Path-Id   IGP Cost
  As-Path                                   Label
-----
u*>i 3.4.0.0/32                               100      None
      192.0.2.6                               None     10
      4 6
u*>i 3.4.0.0/32                               100      None
      192.0.2.7                               None     10
      4 7
u*>i 3.4.0.0/32                               100      None
      192.0.2.8                               None     10
      4 1 8
*>i 3.4.0.0/32                               100      None
      192.0.2.9                               None     10
      4 1 9
-----
Routes : 4
=====

```

The best selection path settings are restored as follows:

```

# on PE-5:
configure {
  router "Base" {
    bgp {
      delete best-path-selection
    }
  }
}

```

BGP multipath restricted to the exact AS path

The BGP multipath configuration on PE-5 restricts BGP to only use identical AS paths, as follows:

```

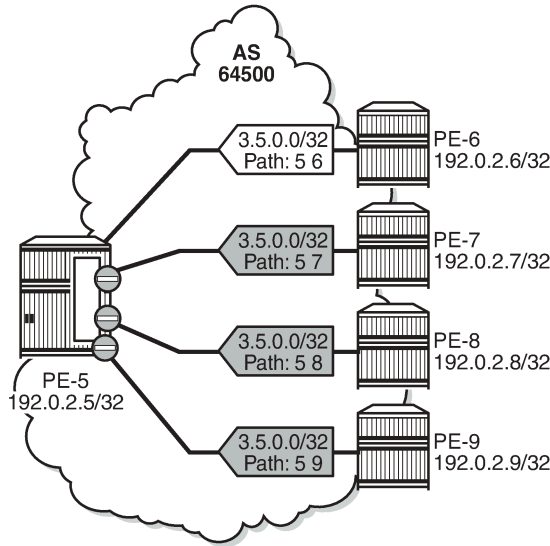
# on PE-5:
configure {
  router "Base" {
    bgp {
      multipath {
        max-paths 8
        ebgp 2
        ibgp 3
        restrict exact-as-path
      }
    }
  }
}

```

The four iBGP neighbors advertise prefixes 3.5.0.0/32 and 3.6.0.0/32 to PE-5, see [Figure 88: BGP multipath restricted to exact same AS. All AS paths are different.](#) and [Figure 89: BGP multipath restricted to exact same AS. All AS paths are identical.](#) The AS paths for prefix 3.5.0.0/32 are not identical, but the neighbor AS is the same, and the AS path is of equal length. The AS paths for prefix 3.6.0.0/32 are identical.

The BGP multipath configuration specifies that the AS paths must be identical, which is not the case for the received BGP routes for prefix 3.5.0.0/32. Only one BGP route is imported in the route table, as shown in [Figure 88: BGP multipath restricted to exact same AS. All AS paths are different.](#)

Figure 88: BGP multipath restricted to exact same AS. All AS paths are different.



26063

All the BGP routes for prefix 3.5.0.0/32 have a different AS path. Only the BGP route advertised by neighbor 192.0.2.6 is used, as follows:

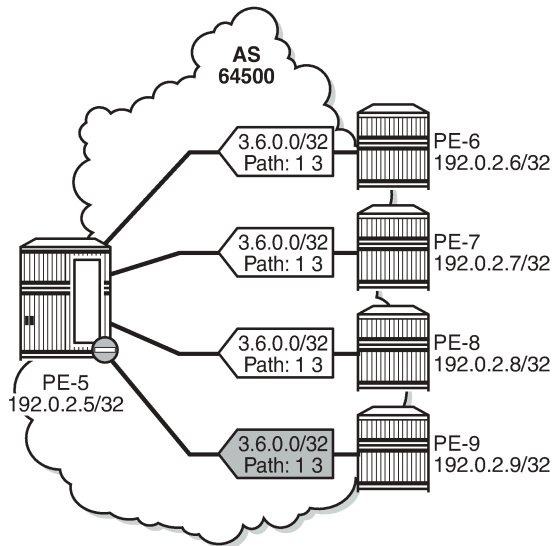
```
[/]
A:admin@PE-5# show router bgp routes 3.5.0.0/32
=====
BGP Router ID:192.0.2.5      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
```

Flag	Network	Nexthop (Router)	As-Path	LocalPref	MED	IGP Cost
u*>i	3.5.0.0/32	192.0.2.6	5 6	100	None	10
*>i	3.5.0.0/32	192.0.2.7	5 7	100	None	10
*>i	3.5.0.0/32	192.0.2.8	5 8	100	None	10
*>i	3.5.0.0/32	192.0.2.9	5 9	100	None	10

```
-----
Routes : 4
=====
```

However, all the received BGP routes for prefix 3.6.0.0/32 have the same AS path. Three of these BGP paths are used, as shown in [Figure 89: BGP multipath restricted to exact same AS. All AS paths are identical](#).

Figure 89: BGP multipath restricted to exact same AS. All AS paths are identical



26064

Three of the four received BGP routes for prefix 3.6.0.0/32 are used, as follows:

```
[/]
A:admin@PE-5# show router bgp routes 3.6.0.0/32
=====
BGP Router ID:192.0.2.5      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
```

Flag	Network Nexthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
u*>i	3.6.0.0/32 192.0.2.6 1 3	100 None	None 10 -
u*>i	3.6.0.0/32 192.0.2.7 1 3	100 None	None 10 -
u*>i	3.6.0.0/32 192.0.2.8 1 3	100 None	None 10 -

```
*>i 3.6.0.0/32          100      None
    192.0.2.9         None      10
    1 3                -
-----
Routes : 4
=====
```

BGP multipath per address family

On PE-6, PE-7, PE-8, and PE-9, the address families IPv4, label-IPv4, and label-IPv6 are configured in the context of iBGP neighbor 192.0.2.5. Prefix 3.7.0.0/32 is exported as IPv4 route, whereas prefix 3.8.0.0/32 is exported as label-IPv4 route, and prefix 2001:db8::3:8:0:0/32 as label-IPv6 route.

On PE-5, the address families IPv4, label-IPv4, and label-IPv6 are configured in the context of the "iBGP" group, each with a different *max-paths* setting: maximum two IPv4 paths, maximum three label-IPv4 paths, and maximum four label-IPv6 paths:

```
# on PE-5:
configure {
  router "Base" {
    bgp {
      multipath {
        family ipv4 {
          max-paths 2
          ibgp 2
        }
        family label-ipv4 {
          max-paths 3
          ibgp 3
        }
        family label-ipv6 {
          max-paths 4
          ibgp 4
        }
      }
    }
    group "iBGP" {
      peer-as 64500
      family {
        ipv4 true
        label-ipv4 true
        label-ipv6 true
      }
    }
  }
}
```

In this example, only iBGP routes are received. Two of the four received IPv4 routes for prefix 3.7.0.0/32 are used:

```
[/]
A:admin@PE-5# show router bgp routes 3.7.0.0/32
=====
BGP Router ID:192.0.2.5      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
```

```

=====
Flag   Network                               LocalPref  MED
      Nexthop (Router)                   Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  3.7.0.0/32                             100        None
      192.0.2.6                             None       10
      7
u*>i  3.7.0.0/32                             100        None
      192.0.2.7                             None       10
      7
*>i  3.7.0.0/32                             100        None
      192.0.2.8                             None       10
      7
*>i  3.7.0.0/32                             100        None
      192.0.2.9                             None       10
      7
-----
Routes : 4
=====

```

The last two IPv4 routes from PE-8 and PE-9 are not used because the maximum number of IPv4 iBGP paths (2) is exceeded:

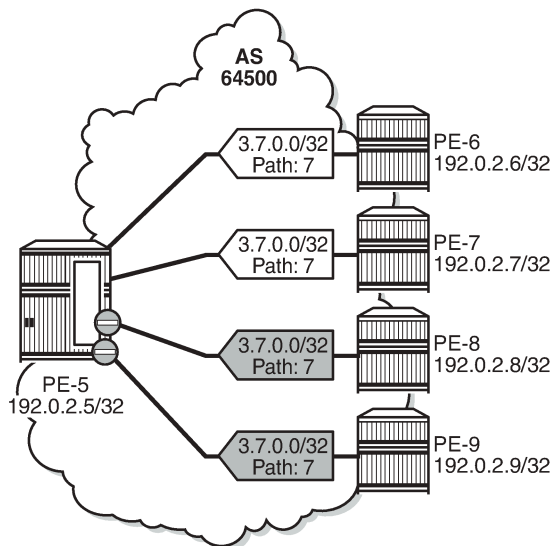
```

[/]A:admin@PE-5# show router bgp routes 3.7.0.0/32 hunt | match "MP Exc. Reason"
TieBreakReason : PeerRouterID           MP Exc. Reason : MaxPathsExceeded
TieBreakReason : PeerRouterID           MP Exc. Reason : MaxPathsExceeded

```

Figure 90: BGP multipath for the IPv4 address family shows that two of the four received IPv4 routes are used.

Figure 90: BGP multipath for the IPv4 address family



36400

Three of the four received label-IPv4 routes for prefix 3.8.0.0/32 are used:

```

[/]
A:admin@PE-5# show router bgp routes 3.8.0.0/32 label-ipv4

```

```

=====
BGP Router ID:192.0.2.5      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP LABEL-IPV4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  3.8.0.0/32                100        None
      192.0.2.6                None        10
      8                          524282
u*>i  3.8.0.0/32                100        None
      192.0.2.7                None        10
      8                          524282
u*>i  3.8.0.0/32                100        None
      192.0.2.8                None        10
      8                          524282
*>i   3.8.0.0/32                100        None
      192.0.2.9                None        10
      8                          524282
-----
Routes : 4
=====

```

The last label-IPv4 route from PE-9 is not used because the maximum number of label-IPv4 paths (3) is exceeded:

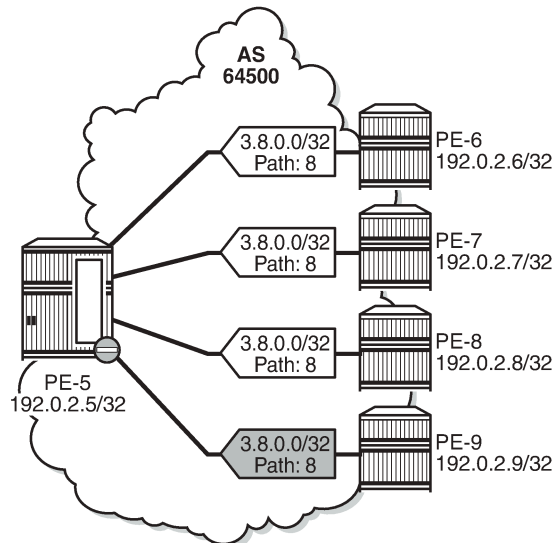
```

[/]
A:admin@PE-5# show router bgp routes 3.8.0.0/32 label-ipv4 hunt | match "MP Exc. Reason"
TieBreakReason : PeerRouterID      MP Exc. Reason : MaxPathsExceeded

```

Figure 91: BGP multipath for the label-IPv4 address family shows that three of the received label-IPv4 routes are used.

Figure 91: BGP multipath for the label-IPv4 address family



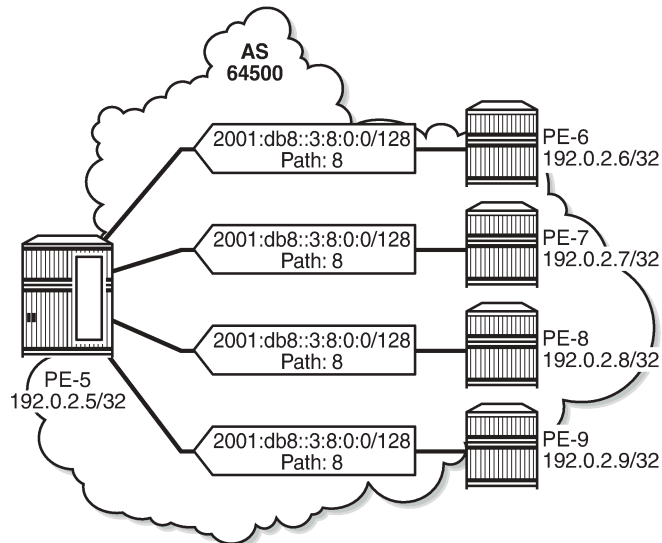
36401

All four received label-IPv6 routes for prefix 2001:db8::3:8:0:0/128 are used:

```
[/]
A:admin@PE-5# show router bgp routes 2001:db8::3:8:0:0/128 label-ipv6
=====
BGP Router ID:192.0.2.5      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP LABEL-IPV6 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  2001:db8::3:8:0:0/128                 100        None
      ::ffff:192.0.2.6                       None       10
      8                                       2
u*>i  2001:db8::3:8:0:0/128                 100        None
      ::ffff:192.0.2.7                       None       10
      8                                       2
u*>i  2001:db8::3:8:0:0/128                 100        None
      ::ffff:192.0.2.8                       None       10
      8                                       2
u*>i  2001:db8::3:8:0:0/128                 100        None
      ::ffff:192.0.2.9                       None       10
      8                                       2
-----
Routes : 4
=====
```

Figure 92: BGP multipath for the label-IPv6 address family shows that all four received label-IPv6 routes are used.

Figure 92: BGP multipath for the label-IPv6 address family



36402

Selective BGP multipath

Entire BGP groups or a selection of BGP neighbors can be configured as **multipath-eligible**. In all preceding examples, all BGP groups and BGP neighbors are—by default—marked as 'not multipath-eligible'. In a scenario where all paths originate from neighbors that are not marked as multipath-eligible, the N best routes are chosen.

For prefixes 3.7.0.0/32, 3.8.0.0/32, and 2001:db8::3:8:0:0/128, the best path is the path originating from neighbor 192.0.2.6, based on the (lowest) router ID.

In the following example, only neighbors 192.0.2.7 and 192.0.2.8 are configured as **multipath-eligible**:

```
# on PE-5:
configure {
  router "Base"
    bgp {
      multipath {
        family ipv4 {
          max-paths 2
          ibgp 2
        }
        family label-ipv4 {
          max-paths 3
          ibgp 3
        }
        family label-ipv6 {
          max-paths 4
          ibgp 4
        }
      }
    }
  neighbor "192.0.2.6" {
```

```

    group "iBGP"
  }
  neighbor "192.0.2.7" {
    multipath-eligible true
    group "iBGP"
  }
  neighbor "192.0.2.8" {
    multipath-eligible true
    group "iBGP"
  }
  neighbor "192.0.2.9" {
    group "iBGP"
  }
}

```

When the best path originates from a neighbor that is not multipath-eligible (default), while at least one path originates from a neighbor that is marked as **multipath-eligible**, only the best path is used (no multipath in this scenario):

```

[/]
A:admin@PE-5# show router bgp routes 3.7.0.0/32
=====
BGP Router ID:192.0.2.5      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path
-----
u*>i 3.7.0.0/32                               100        None
      192.0.2.6                               None       10
      7
*>i  3.7.0.0/32                               100        None
      192.0.2.7                               None       10
      7
*>i  3.7.0.0/32                               100        None
      192.0.2.8                               None       10
      7
*>i  3.7.0.0/32                               100        None
      192.0.2.9                               None       10
      7
-----
Routes : 4
=====

```

The routes originating from PE-7, PE-8, and PE-9 are not used because the best BGP path toward 3.7.0.0/32 originates from PE-6, which is not multipath-eligible:

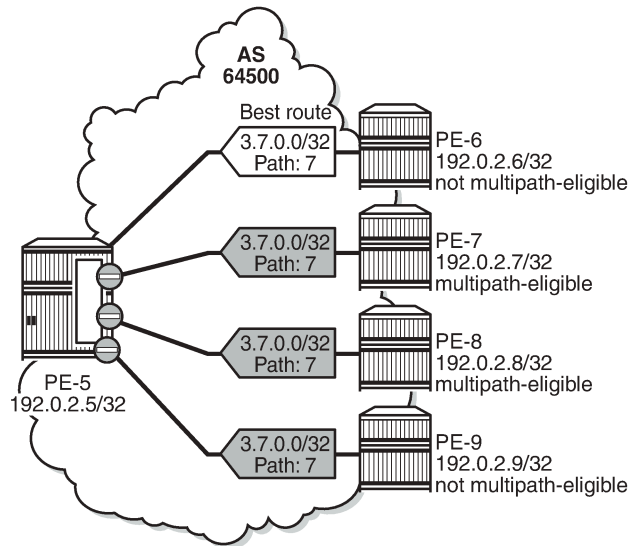
```

[/]
A:admin@PE-5# show router bgp routes 3.7.0.0/32 hunt | match "MP Exc. Reason"
TieBreakReason : PeerRouterID      MP Exc. Reason : NotMultipathEligible
TieBreakReason : PeerRouterID      MP Exc. Reason : NotMultipathEligible
TieBreakReason : PeerRouterID      MP Exc. Reason : NotMultipathEligible

```

Figure 93: Best IPv4 path originates from a non-multipath-eligible BGP neighbor shows that only the best IPv4 route is used when the best path originates from a non-multipath-eligible BGP neighbor.

Figure 93: Best IPv4 path originates from a non-multipath-eligible BGP neighbor



36403

Also, for the label-IPv4 and label-IPv6 routes, only the best path is used and the other routes are not included in the multipath because the best path is not multipath-eligible.

In the following example, BGP neighbors 192.0.2.6, 192.0.2.8, and 192.0.2.9 are configured as multipath-eligible:

```
# on PE-5:
configure {
  router "Base" {
    bgp {
      multipath {
        family ipv4 {
          max-paths 2
          ibgp 2
        }
        family label-ipv4 {
          max-paths 3
          ibgp 3
        }
        family label-ipv6 {
          max-paths 4
          ibgp 4
        }
      }
    }
    neighbor "192.0.2.6" {
      multipath-eligible true
      group "iBGP"
    }
    neighbor "192.0.2.7" {
      delete multipath-eligible # restore default
      group "iBGP"
    }
    neighbor "192.0.2.8" {
```

```

        multipath-eligible true
        group "iBGP"
    }
    neighbor "192.0.2.9" {
        multipath-eligible true
        group "iBGP"
    }

```

The best path originates from neighbor 192.0.2.6 that is marked as multipath-eligible. In this case, only paths marked as multipath-eligible are candidates for the BGP multipath algorithm and the best N multipath-eligible routes will be chosen (if available): two IPv4 paths, three label-IPv4 paths, and four label-IPv6 paths.

On PE-5, two IPv4 routes are used for prefix 3.7.0.0/32: the best path from neighbor 192.0.2.6 and a path from neighbor 192.0.2.8:

```

[/]
A:admin@PE-5# show router bgp routes 3.7.0.0/32
=====
BGP Router ID:192.0.2.5      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
u*>i  3.7.0.0/32                100        None
      192.0.2.6                None        10
      7
*>i   3.7.0.0/32                100        None
      192.0.2.7                None        10
      7
u*>i  3.7.0.0/32                100        None
      192.0.2.8                None        10
      7
*>i   3.7.0.0/32                100        None
      192.0.2.9                None        10
      7
-----
Routes : 4
=====

```

IPv4 route from neighbor 192.0.2.7 is not used because it is not multipath-eligible; IPv4 route from neighbor 192.0.2.9 is not used because the maximum number of IPv4 paths is exceeded, as follows:

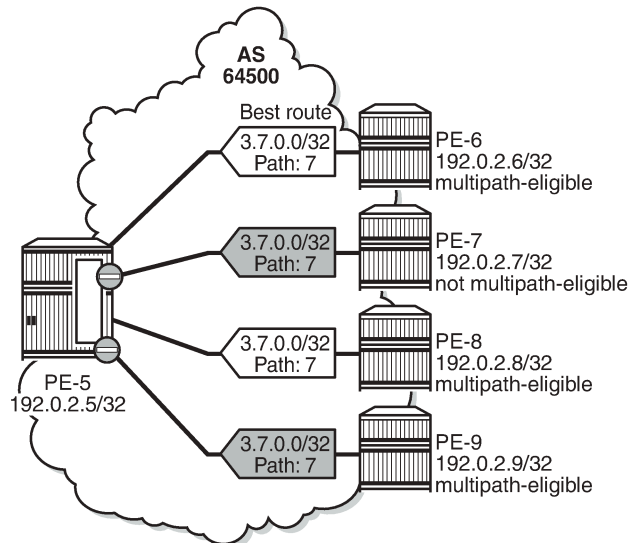
```

[/]
A:admin@PE-5# show router bgp routes 3.7.0.0/32 hunt | match "MP Exc. Reason"
TieBreakReason : PeerRouterID      MP Exc. Reason : NotMultipathEligible
TieBreakReason : PeerRouterID      MP Exc. Reason : MaxPathsExceeded

```

Figure 93: Best IPv4 path originates from a non-multipath-eligible BGP neighbor shows that two IPv4 routes from multipath-eligible peers are used: the best path originating from PE-6 and the second best path originating from PE-8.

Figure 94: Two IPv4 paths from multipath-eligible BGP peers are used



36404

Conclusion

BGP multipath allows the IP routing table to have multiple BGP paths to the same destination. Different path limits can be applied for eBGP and iBGP paths and per address family. It is possible to treat eBGP and iBGP routes as equal. Restrictions can be imposed related to AS path. Specific BGP neighbors or entire BGP groups can be marked as multipath-eligible, resulting in selective BGP multipath behavior.

BGP Optimal Route Reflection for Hierarchical Networks

This chapter provides information about BGP optimal route reflection for hierarchical networks.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

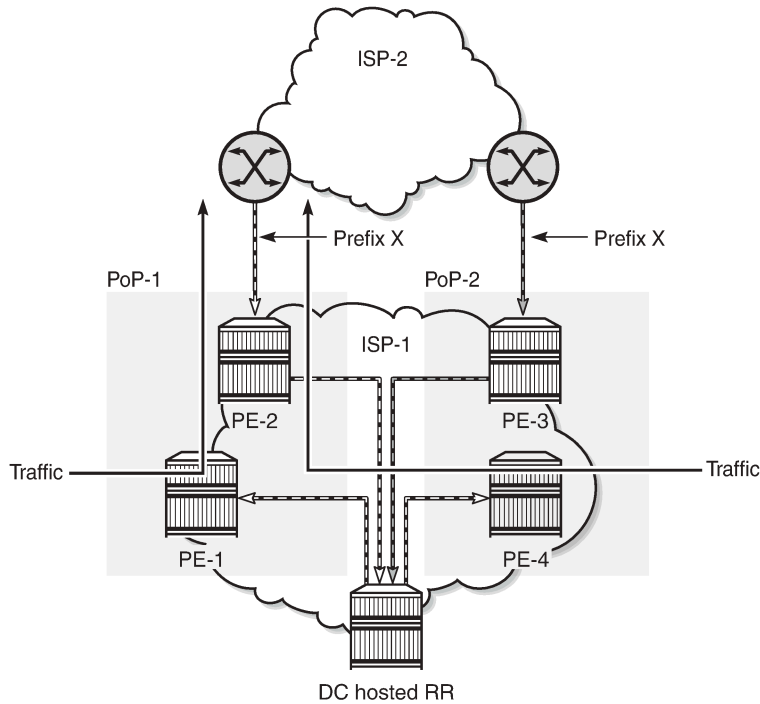
This chapter was initially written based on SR OS Release 15.0.R4, but the MD-CLI in the current edition corresponds to SR OS Release 23.7.R2.

Overview

BGP route reflectors are used in many networks. They improve network scalability by eliminating or reducing the need for a full-mesh of IBGP sessions.

When a BGP route reflector receives multiple paths for the same IP destination, it normally selects and reflects a single best path in its routing domain to all clients in that domain, based on its own location in the domain. In [Figure 95: Centralized route reflection](#), the centralized route reflector RR for ISP-1 is located in the datacenter (DC), and receives prefix X from ISP-2 through PE-2 in point of presence PoP-1 and also through PE-3 in PoP-2. RR selects and reflects PE-2 as the best path to the remaining route reflector clients because RR is closer to PoP-1 than it is to PoP-2, so the traffic to destination X flows as indicated. Therefore, sending traffic to another autonomous system (AS) through the closest possible exit point from the local AS, known as hot-potato routing, cannot be achieved.

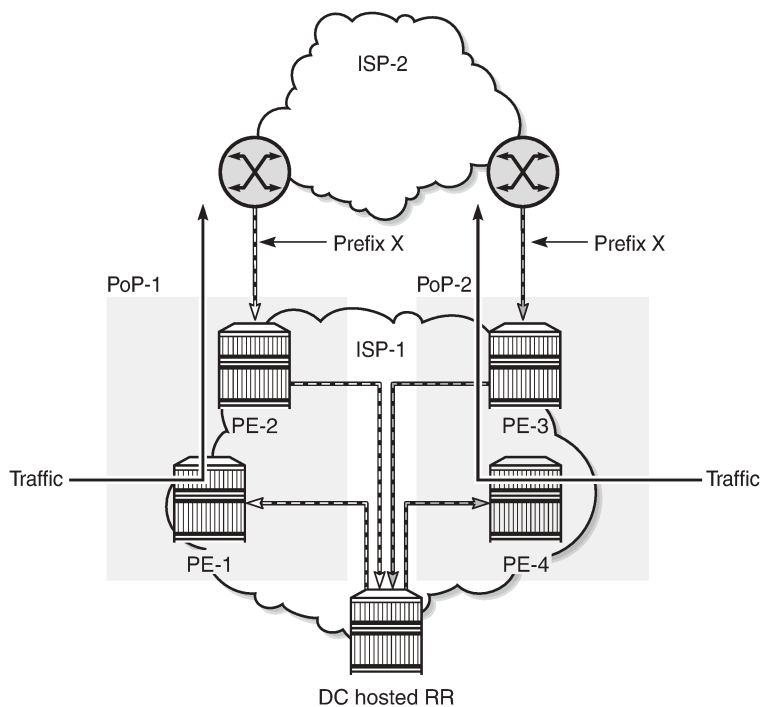
Figure 95: Centralized route reflection



26679

Hot-potato routing can be achieved using a route reflector selecting and reflecting multiple best paths, for different subdomains and from the point of view of a client in a subdomain, as outlined in RFC 9107 *BGP optimal route reflection* (ORR), and requires the route reflector to know the topology of each subdomain. In [Figure 96: Centralized route reflection with ORR](#), the route reflector calculates the best path for PoP-1 and reflects that to the clients in PoP-1 (PE-1), and it also calculates the best path for PoP-2 and reflects that to the clients in PoP-2 (PE-4).

Figure 96: Centralized route reflection with ORR



26680

If the routing domain is non-hierarchical, the route reflector is part of the routing domain and thus has a view on the entire topology through the interior gateway protocol (IGP). See the [BGP Optimal Route Reflection for Non-Hierarchical Networks](#) chapter if the network topology is non-hierarchical.

If the routing domain is hierarchical, the route reflector needs to extract the link state database (LSDB) from the subdomains it is not part of, which is achieved through BGP link state (BGP-LS). The use of BGP-LS allows the route reflector to learn the IGP topology information for OSPF areas and IS-IS levels in which the route reflector is not a direct participant.

ORR CLI commands

The BGP **optimal-route-reflection** context defines the shortest path first (SPF) parameters, and multiple locations.

```
*[ex:/configure router "Base" bgp]
A:admin@RR-5# optimal-route-reflection ?

optimal-route-reflection

location          + Enter the location list instance
spf-wait          + Enter the spf-wait context
```

The SPF calculation is configurable with the **spf-wait** command. **Initial-wait** and **second-wait** are optional arguments. These timers define when to initiate the first, second, and subsequent SPF runs after a topology change occurs.

```
*[ex:/configure router "Base" bgp optimal-route-reflection]
A:admin@RR-5# spf-wait ?

spf-wait

initial-wait      - Initial SPF calculation delay after a topology change
max-wait          - Maximum interval between consecutive SPF calculations
second-wait       - Delay between first and second SPF calculation
```

Multiple locations can be created in the **optimal-route-reflection** context, as follows. Each location is identified through a location ID [1..255], and contains a primary IP address and, optionally, a secondary IP address and a tertiary IP address, for redundancy reasons. These addresses must correspond to loopback or system IP addresses of routers participating in the IGP protocols, and are used as the starting point (or seed) for the SPF calculation. Because all clients in the same location receive the same optimal path for that location, these addresses must be close to the clients in that part of the network.

```
*[ex:/configure router "Base" bgp optimal-route-reflection location 1]
A:admin@RR-5# ?

apply-groups      - Apply a configuration group at this level
apply-groups-exclude - Exclude a configuration group at this level
primary-ip-address - Primary IPv4 address of the reference location for ORR
primary-ipv6-address - Primary IPv6 address of the reference location for ORR
secondary-ip-address - Secondary IPv4 address of reference location for ORR
secondary-ipv6-address - Secondary IPv6 address of reference location for ORR
tertiary-ip-address - Tertiary IPv4 address of the reference location for ORR
tertiary-ipv6-address - Tertiary IPv6 address of the reference location for ORR
```

The locations are then referred to with the **cluster** command (residing in the BGP group or neighbor context) through the **orr-location** argument, as follows.

```
*[ex:/configure router "Base" bgp group "IBGP-1"]
A:admin@RR-5# cluster ?

cluster

allow-local-fallback - Allow fallback to RR topology location
cluster-id           - Route reflector cluster ID
orr-location       - Optimal route reflection location for the cluster

*[ex:/configure router "Base" bgp neighbor "192.0.2.3"]
A:admin@RR-5# cluster ?

cluster

allow-local-fallback - Allow fallback to RR topology
cluster-id           - Route reflector cluster ID
orr-location       - Optimal route reflection location for the cluster
```

The location ID is referred to in the **orr-location** argument of the **cluster** command. Typically, the **cluster** command applies to a BGP peer group; all neighbors in that group share the same location ID, unless the **cluster** command applies at a neighbor level. The **allow-local-fallback** option allows the RR to advertise

the best reachable BGP path using its own location, but only when no BGP routes are reachable for some location. Otherwise, no path would be advertised to the clients in that location.

Properties

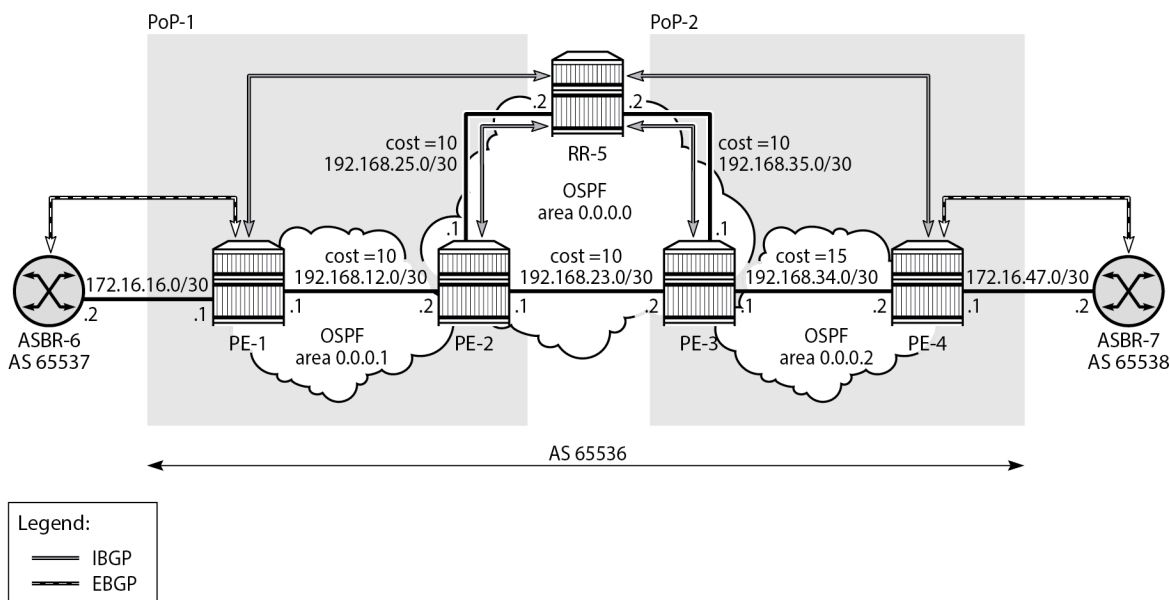
The following properties apply to ORR in SR OS:

- ORR is supported in the Base router BGP instance.
- ORR is supported for the IPv4, label-IPv4, label-IPv6, VPN-IPv4, and VPN-IPv6 address families.
- ORR is supported with add-paths, meaning that add-paths advertised to ORR clients are also ORR location-based.

Configuration

Figure 97: Example hierarchical networking using OSPF shows the example topology. OSPF is used as the IGP for AS 65536, with RR-5 taking the role of the route reflector for clients PE-1 to PE-4. The OSPF backbone area is area 0.0.0.0, connecting routers PE-2, PE-3, and RR-5. Area 0.0.0.1 is a stub area interconnecting PE-1 and PE-2; area 0.0.0.2 is a stub area interconnecting PE-3 and PE-4. Both PE-2 and PE-3 are area border routers (ABRs). Additionally, ASBR-6 in AS 65537 peers with PE-1, and ASBR-7 in AS 65538 peers with PE-4.

Figure 97: Example hierarchical networking using OSPF



26685

The initial configuration on all nodes includes:

- Cards, MDAs, and ports
- Router interfaces

- OSPF as IGP on all interfaces within AS 65536, with multiple non-backbone areas (alternatively, IS-IS can be used), and traffic engineering enabled

The following shows the OSPF configuration on ABR PE-3 with some interfaces in backbone area 0.0.0.0 and other interfaces in stub area 0.0.0.2. The metric on the interfaces is 10, except for the interface between PE-3 and PE-4 with metric 15 in stub area 0.0.0.2.

```
# on PE-3:
configure {
  router "Base" {
    ospf 0 {
      admin-state enable
      traffic-engineering true
      area 0.0.0.0 {
        interface "int-PE-3-PE-2" {
          interface-type point-to-point
          metric 10
        }
        interface "int-PE-3-RR-5" {
          interface-type point-to-point
          metric 10
        }
        interface "system" {
        }
      }
      area 0.0.0.2 {
        stub {
        }
        interface "int-LB-BGP" {
        }
        interface "int-PE-3-PE-4" {
          interface-type point-to-point
          metric 15
        }
      }
    }
  }
}
```

Route reflection without ORR

RR-5 peers with clients PE-1 to PE-4, and because RR-5 is the route reflector, the **cluster** command is added, defining the cluster ID attribute value to use. The configuration for RR-5 is as follows:

```
# on RR-5:
configure {
  router "Base" {
    autonomous-system 65536
    bgp {
      loop-detect discard-route
      split-horizon true
      group "IBGP" {
        peer-as 65536
        cluster {
          cluster-id 192.0.2.5
        }
      }
      neighbor "192.0.2.1" {
        group "IBGP"
      }
      neighbor "192.0.2.2" {
        group "IBGP"
      }
    }
  }
}
```

```

    }
    neighbor "192.0.2.3" {
        group "IBGP"
    }
    neighbor "192.0.2.4" {
        group "IBGP"
    }
}

```

PE-1 belongs to the cluster defined in the route reflector, so it does not need to be fully meshed with the other routers in the area; peering with the route reflectors in the area is sufficient for PE-1 to receive updates. Typically, two route reflectors are provisioned for redundancy, but that does not apply in this example. PE-1 also peers with ASBR-6 in AS 65537 through EBGP, so the PE-1 configuration is as follows:

```

# on PE-1:
configure {
    router "Base" {
        autonomous-system 65536
        bgp {
            loop-detect discard-route
            split-horizon true
            group "EBGP" {
            }
            group "IBGP" {
                next-hop-self true
                peer-as 65536
            }
        }
        neighbor "172.16.16.2" {
            group "EBGP"
            peer-as 65537
            ebgp-default-reject-policy {
                import false
            }
        }
        neighbor "192.0.2.5" {
            group "IBGP"
        }
    }
}

```

PE-2 and PE-3 only peer with the route reflector. Their configuration is the same:

```

# on PE-2, PE-3:
configure {
    router "Base" {
        autonomous-system 65536
        bgp {
            loop-detect discard-route
            split-horizon true
            group "IBGP" {
                peer-as 65536
            }
        }
        neighbor "192.0.2.5" {
            group "IBGP"
        }
    }
}

```

PE-4 also belongs to the IBGP cluster defined in the route reflector and PE-4 peers with ASBR-7 in AS 65538. The PE-4 configuration is similar to the configuration of PE-1.

Loopback address 10.1.11.1/24 is configured on ASBR-8 in AS 65540 (not shown in the example topology). ASBR-8 exports prefix 10.1.11.0/24 to its EBGP peers ASBR-6 in AS 65537 and ASBR-7 in AS 65538. ASBR-6 advertises prefix 10.1.11.0/24 to router PE-1; ASBR-7 advertises the same prefix to router PE-4.

RR-5 receives IBGP updates from PE-1 and PE-4, and selects the best path based on its own position in the topology. The IGP cost from RR-5 to PE-1 is 20, and the cost from RR-5 to PE-4 is 25, so RR-5 selects the BGP path with next hop 192.0.2.1.

```
[/]
A:admin@RR-5# show router bgp routes
=====
BGP Router ID:192.0.2.5      AS:65536      Local AS:65536
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
      Nexthop (Router)                     Path-Id   IGP Cost
      As-Path                               Label
-----
u*>i 10.1.11.0/24                           100      None
      192.0.2.1                             None     20
      65537 65540
*i    10.1.11.0/24                           100      None
      192.0.2.4                             None     25
      65538 65540
-----
Routes : 2
=====
```

RR-5 reflects the path with next hop 192.0.2.1 to all clients except PE-1, because PE-1 is the client where the path was learned from).

For prefix 10.1.11.0/24, PE-1 received an EBGP route from ASBR-6 in AS 65537 with next hop 172.16.16.2 and no IBGP route from RR-5:

```
[/]
A:admin@PE-1# show router bgp routes
=====
BGP Router ID:192.0.2.1      AS:65536      Local AS:65536
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
      Nexthop (Router)                     Path-Id   IGP Cost
      As-Path                               Label
-----
u*>i 10.1.11.0/24                           None     None
      172.16.16.2                          None     0
      65537 65540
-----
```

```
-----
Routes : 1
=====
```

As a result, traffic offered to PE-1 for destination 10.1.11.0/24 is routed to ASBR-6, as follows:

```
[/]
A:admin@PE-1# show router route-table protocol bgp
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]          Metric
-----
10.1.11.0/24                       Remote BGP      00h04m15s  170
  172.16.16.2                       0
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```

PE-2 received an IBGP route for prefix 10.1.11.0/24 with next hop 192.0.2.1 from RR-5:

```
[/]
A:admin@PE-2# show router bgp routes
=====
BGP Router ID:192.0.2.2      AS:65536      Local AS:65536
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  10.1.11.0/24             100        None
      192.0.2.1             None        10
      65537 65540           -
-----
Routes : 1
=====
```

Traffic offered to PE-2 for destination 10.1.11.0/24 is routed to PE-1, as follows:

```
[/]
A:admin@PE-2# show router route-table protocol bgp
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]          Metric
-----
```

```

10.1.11.0/24                               Remote BGP           00h03m22s 170
      192.168.12.1                          10
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

Likewise, PE-3 received an IBGP route for prefix 10.1.11.0/24 with next hop 192.0.2.1 from RR-5:

```

[/]
A:admin@PE-3# show router bgp routes
=====
BGP Router ID:192.0.2.3      AS:65536      Local AS:65536
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                  l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i 10.1.11.0/24                           100        None
      192.0.2.1                             None       20
      65537 65540                             -
-----
Routes : 1
=====

```

Traffic offered to PE-3 for destination 10.1.11.0/24 is routed via the interface address 192.168.23.1 on PE-2, as follows:

```

[/]
A:admin@PE-3# show router route-table protocol bgp
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
      Next Hop[Interface Name]                Metric
-----
10.1.11.0/24                       Remote BGP      00h03m20s 170
      192.168.23.1                          20
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```


For prefix 10.1.11.0/24, PE-4 received an EBGP route from ASBR-7 with next hop 172.16.47.2 and an IBGP route from RR-5 with next hop 192.0.2.1, as follows. EBGP routes are preferred over IBGP routes.

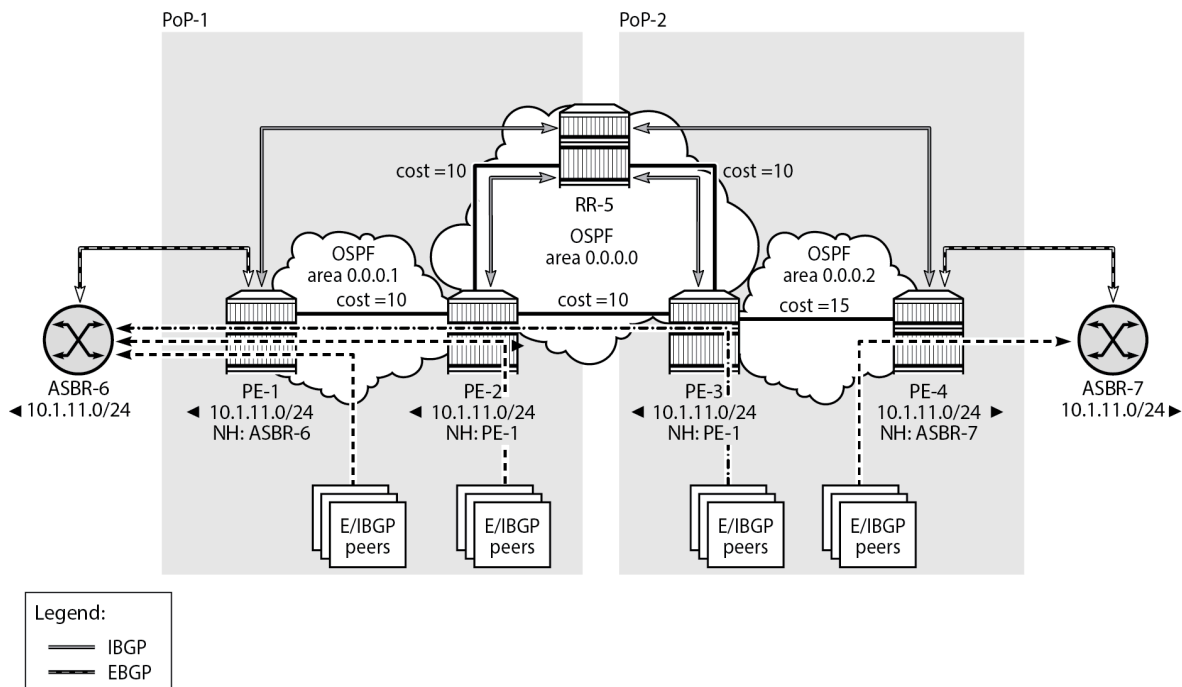
```
[/]
A:admin@PE-4# show router bgp routes
=====
BGP Router ID:192.0.2.4      AS:65536      Local AS:65536
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
u*>i  10.1.11.0/24              None       None
      172.16.47.2           None       0
      65538 65540
*i    10.1.11.0/24              100       None
      192.0.2.1             None       35
      65537 65540
-----
Routes : 2
=====
```

The used route is the EBGP route from ASBR-7, so the traffic offered to PE-4 for destination 10.1.11.0/24 is routed to ASBR-7, as follows:

```
[/]
A:admin@PE-4# show router route-table protocol bgp
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age           Pref
Next Hop[Interface Name]   Path-Id  Metric
-----
10.1.11.0/24                Remote  BGP    00h03m59s    170
      172.16.47.2           0
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

This is summarized in [Figure 98: Suboptimal route reflection](#). Ultimately, PE-1 only has one path, and so do PE-2 and PE-3. PE-4 has two paths, but by default prefers the EBGP learned path over the IBGP learned path. The routing is suboptimal on PE-3, where the IGP cost to PE-1 is 20 and the IGP cost to PE-4 is 15.

Figure 98: Suboptimal route reflection



26686

Route reflection with ORR

Implementing ORR using the hierarchical topology from [Figure 98: Suboptimal route reflection](#) requires changes in the non-backbone OSPF areas as well as changes to the route reflector.

Because the route reflector is part of the backbone area, and ABRs do not pass the link state advertisements (LSAs) describing the topology and the traffic engineering data for the non-backbone areas, that data must be extracted from the non-backbone areas and copied to the route reflector. This is achieved using BGP-LS, with additional support from OSPF.

In this example, BGP-LS is activated in PE-1, PE-4, and RR-5. PE-1 in area 0.0.0.1 has the BGP-LS address family configured. The BGP option **link-state-import-enable** is needed for PE-1 to advertise the LSDB and traffic engineering database (TED) to the route reflector. On the same router PE-1, OSPF is instructed to provide the **bgp-ls-identifier 1** using the **database-export** command. The configuration for PE-1 is as follows:

```
# on PE-1:
configure {
  router "Base" {
    ospf 0 {
      admin-state enable
      traffic-engineering true
      database-export {
        igp-identifier 1
        bgp-ls-identifier {
          value 1
        }
      }
    }
  }
}
```

```
    }
    area 0.0.0.1 {
        stub {
        }
        interface "int-PE-1-PE-2" {
            interface-type point-to-point
            metric 10
        }
        interface "system" {
        }
    }
}
bgp {
    loop-detect discard-route
    split-horizon true
    link-state-route-import true
    group "EBGP" {
        peer-as 65537
    }
    group "IBGP" {
        next-hop-self true
        peer-as 65536
        family {
            ipv4 true
            bgp-ls true
        }
    }
    neighbor "172.16.16.2" {
        group "EBGP"
        peer-as 65537
        ebgp-default-reject-policy {
            import false
        }
    }
    neighbor "192.0.2.5" {
        group "IBGP"
    }
}
```

The configuration on PE-4 is similar, and there the **bgp-ls-identifier** is set to 2. Routers PE-2 and PE-3 do not need to be reconfigured.

RR-5 in the backbone area also has BGP-LS activated with the **family** command, and **link-state-export-enable true** is required for accepting and storing the LSDB and TED. No reconfiguration of OSPF is required in RR-5.

For implementing ORR using the hierarchical topology shown in [Figure 103: Suboptimal route reflection](#), the route reflector RR-5 defines two locations in the **optimal-route-reflection** context. The primary IP address for location 1 is the PE-1 system IP address 192.0.2.1; the primary IP address for location 2 is loopback address 192.0.2.44 on PE-4 and the secondary IP address is loopback address 192.0.2.33 on PE-3. These addresses are used as the starting point for the SPF run. The ORR locations 1 and 2 are then referred to from within the group definitions through the **cluster** command. Because RR-5 is not on the data path, there is no need for implementing the routes into the FIB, which is achieved through the **route-table-install false** command. The overall BGP configuration of RR-5 is as follows:

```
# on RR-5:
configure {
    router "Base" {
        autonomous-system 65536
        bgp {
            loop-detect discard-route
```

```

route-table-install false
split-horizon true
link-state-route-export true
family {
    bgp-ls true
}
optimal-route-reflection {
    spf-wait {
        max-wait 1
        initial-wait 1
        second-wait 1
    }
    location 1 {
        primary-ip-address 192.0.2.1
    }
    location 2 {
        primary-ip-address 192.0.2.44 # loopback address on PE-4
        secondary-ip-address 192.0.2.33 # loopback address on PE-3
    }
}
group "IBGP-1" {
    peer-as 65536
    cluster {
        cluster-id 192.0.2.5
        orr-location 1
        allow-local-fallback true
    }
}
group "IBGP-2" {
    peer-as 65536
    cluster {
        cluster-id 192.0.2.5
        orr-location 2
        allow-local-fallback true
    }
}
neighbor "192.0.2.1" {
    group "IBGP-1"
}
neighbor "192.0.2.2" {
    group "IBGP-1"
}
neighbor "192.0.2.3" {
    group "IBGP-2"
}
neighbor "192.0.2.4" {
    group "IBGP-2"
}
}

```

With these changes applied, the following command can be used for verification of the BGP sessions:

```

[/]
A:admin@RR-5# show router bgp summary all

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId          AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)

```

```

PktSent OutQ
-----
192.0.2.1
Def. Inst      65536      20      0 00h03m56s 1/0/0 (IPv4)
                19      0                18/0/18 (LinkState)
192.0.2.2
Def. Inst      65536      11      0 00h03m56s 0/0/1 (IPv4)
                12      0
192.0.2.3
Def. Inst      65536      11      0 00h03m56s 0/0/1 (IPv4)
                12      0
192.0.2.4
Def. Inst      65536      20      0 00h03m56s 1/0/0 (IPv4)
                19      0                18/0/18 (LinkState)
-----

```

ASBR-6 advertises prefix 10.1.11.0/24 to router PE-1; ASBR-7 advertises the same prefix to router PE-4. RR-5 receives the updates from PE-1 and PE-4, and now performs two SPF runs because two locations are used. The first SPF run uses the 192.0.2.1 address of PE-1 as the starting point for the first location, selects the path via PE-1 as the best path, and reflects that path to the remaining peers in the first location. The second SPF run uses the 192.0.2.44 loopback address of PE-4 as the starting point for the second location, selects the path via PE-4 as the best path, and reflects that path to the remaining peers in the second location.

In comparison with the previous scenario, there only is a change in the routing for this prefix on PE-3. RR-5 reflects the route with next hop 192.0.2.4 to PE-3.

```

[/]
A:admin@PE-3# show router bgp routes
=====
BGP Router ID:192.0.2.3      AS:65536      Local AS:65536
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
      Nexthop (Router)                     Path-Id   IGP Cost
      As-Path                               Path-Id   Label
-----
u*>i 10.1.11.0/24                            100      None
      192.0.2.4                              None     15
      65538 65540                             -
-----
Routes : 1
=====

```

Traffic offered to PE-3 for destination 10.1.11.0/24 has next hop PE-4 and is routed via the interface address 192.168.34.2 on PE-4, as follows:

```

[/]
A:admin@PE-3# show router route-table protocol bgp
=====
Route Table (Router: Base)
=====

```

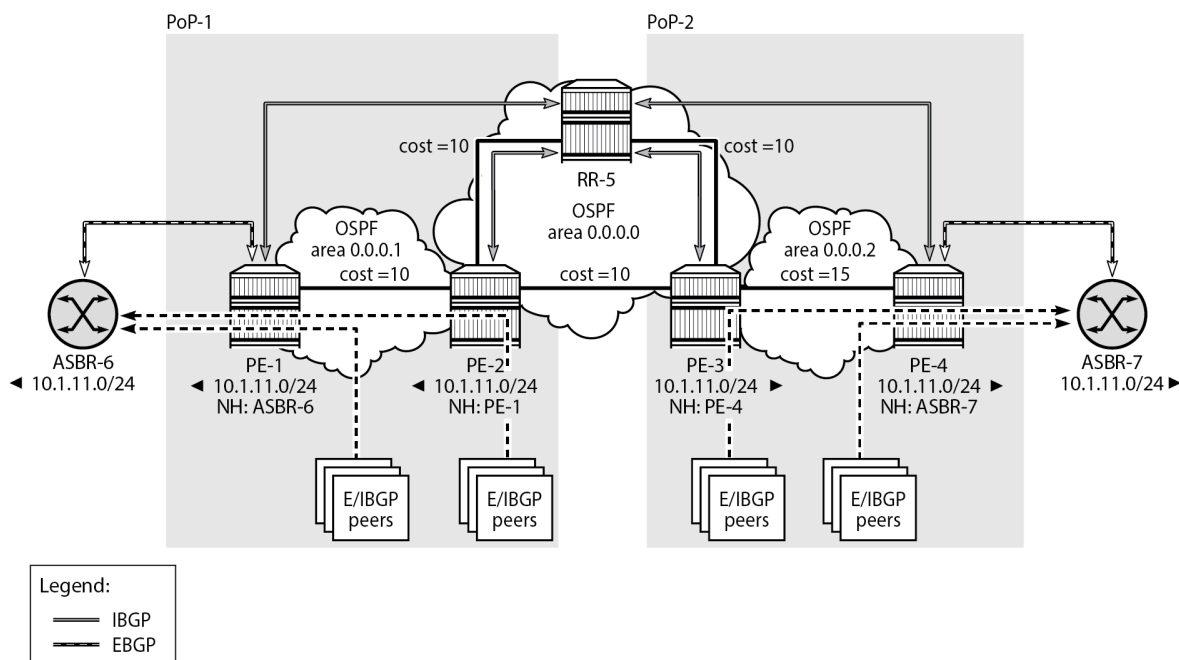
```

Dest Prefix[Flags]
Next Hop[Interface Name]
-----
10.1.11.0/24
  192.168.34.2
-----
Type      Proto  Age      Pref
-----
Remote   BGP    00h04m56s 170
          15
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

This is summarized in [Figure 99: Optimal route reflection](#).

Figure 99: Optimal route reflection



26687

The following command provides the IGP distances for the configured reference points to all available BGP peers and all detected BGP next hops on the route reflector.

```

[/]
A:admin@RR-5# show router bgp optimal-route-reflection bgp-nh-info

=====
ORR BGP-NH Table (Router: Base)
=====
Location 1:
  Primary      : 192.0.2.1 [active]
  Secondary    : -
  Tertiary     : -
  Primary-ipv6 : -
  Secondary-ipv6 : -
  Tertiary-ipv6 : -
Location 2:

```

```

Primary      : 192.0.2.44 [active]
Secondary    : 192.0.2.33
Tertiary     : -
Primary-ipv6 : -
Secondary-ipv6 : -
Tertiary-ipv6 : -

Age          : 00h05m43s
Spf wait     : 1
Initial wait : 1
Second wait  : 1

-----
Next Hop
Loc  Dest-Prefix          DB-Source  Type      Proto    Metric  Pref
-----
192.0.2.1
  1  192.0.2.1/32          BGP-LS     Local     Local     0        0
  2  192.0.2.1/32          BGP-LS     Remote    OSPFv2    35       10

192.0.2.4
  1  192.0.2.4/32          BGP-LS     Remote    OSPFv2    35       10
  2  192.0.2.4/32          BGP-LS     Local     Local     0        0

-----
No. of BGP-NHs: 2
=====

```

Conclusion

BGP optimal route reflection allows operators to optimize traffic streams through their network, even when the route reflector is placed out-of-path, for example in datacenters, thereby reducing the OPEX and CAPEX of route reflector deployment.

BGP Optimal Route Reflection for Non-Hierarchical Networks

This chapter provides information about BGP optimal route reflection for non-hierarchical networks.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

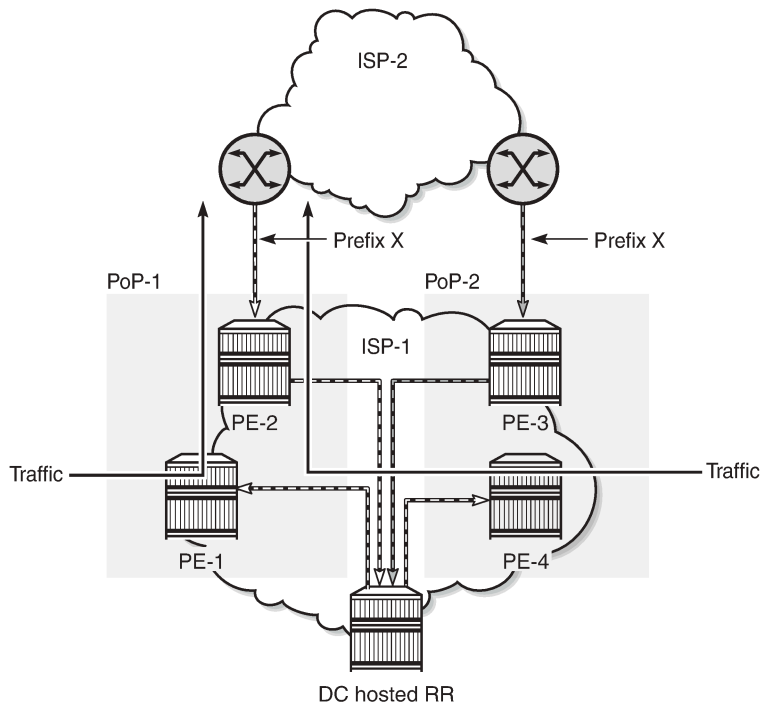
This chapter was initially written based on SR OS Release 15.0.R4, but the MD-CLI in the current edition corresponds to SR OS Release 23.7.R2.

Overview

BGP route reflectors are used in many networks. They improve network scalability by eliminating or reducing the need for a full-mesh of IBGP sessions.

When a BGP route reflector receives multiple paths for the same IP destination, it normally selects and reflects a single best path in its routing domain to all clients in that domain, based on its own location in the domain. In [Figure 100: Centralized route reflection](#), the centralized route reflector RR for ISP-1 is located in the datacenter (DC), and receives prefix X from ISP-2 through PE-2 in point of presence PoP-1 and also through PE-3 in PoP-2. RR selects and reflects PE-2 as the best path to the remaining route reflector clients because RR is closer to PoP-1 than it is to PoP-2, so the traffic to destination X flows as indicated. Therefore, sending traffic to another autonomous system (AS) through the closest possible exit point from the local AS, known as hot-potato routing, cannot be achieved.

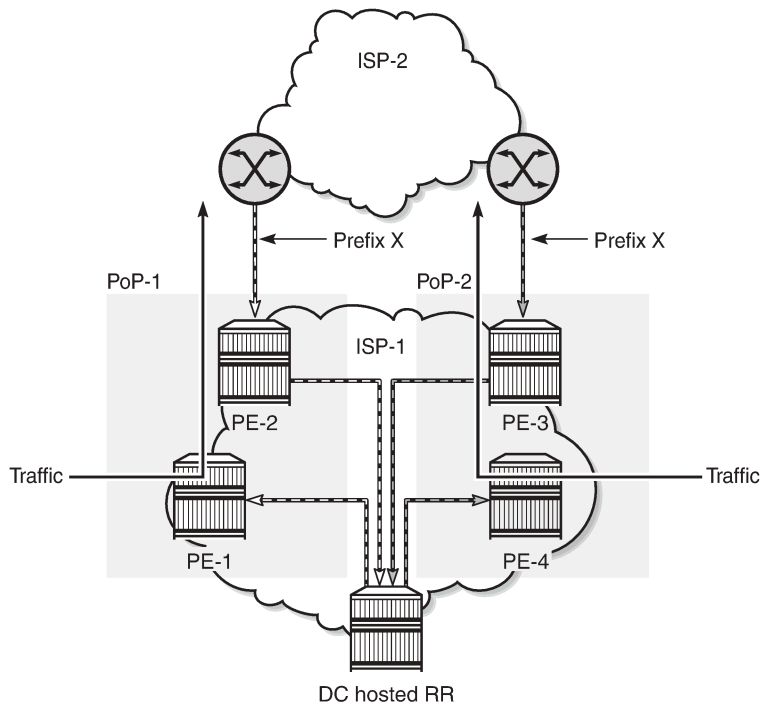
Figure 100: Centralized route reflection



26679

Hot-potato routing can be achieved using a route reflector selecting and reflecting multiple best paths, for different subdomains and from the point of view of a client in a subdomain, as outlined in RFC 9107 *BGP optimal route reflection* (ORR), and requires the route reflector to know the topology of each subdomain. In [Figure 101: Centralized route reflection with ORR](#), the route reflector calculates the best path for PoP-1 and reflects that to the clients in PoP-1 (PE-1), and it also calculates the best path for PoP-2 and reflects that to the clients in PoP-2 (PE-4).

Figure 101: Centralized route reflection with ORR



26680

If the routing domain is non-hierarchical, the route reflector is part of the routing domain and thus has a view on the entire topology through the interior gateway protocol (IGP).

If the routing domain is hierarchical, the route reflector needs to extract the link state database (LSDB) from the subdomain it is not part of, which is achieved through BGP link state (BGP-LS). The use of BGP-LS allows the route reflector to learn the IGP topology information for OSPF areas and IS-IS levels in which the route reflector is not a direct participant. See the [BGP Optimal Route Reflection for Hierarchical Networks](#) chapter if the network topology is hierarchical.

ORR CLI commands

The BGP **optimal-route-reflection** context defines the shortest path first (SPF) parameters, and multiple locations.

```
*[ex:/configure router "Base" bgp]
A:admin@RR-5# optimal-route-reflection ?

optimal-route-reflection

location          + Enter the location list instance
spf-wait          + Enter the spf-wait context
```

The SPF calculation is configurable with the **spf-wait** command. **Initial-wait** and **second-wait** are optional arguments. These timers define when to initiate the first, second, and subsequent SPF runs after a topology change occurs.

```
*[ex:/configure router "Base" bgp optimal-route-reflection]
A:admin@RR-5# spf-wait ?

spf-wait

initial-wait      - Initial SPF calculation delay after a topology change
max-wait          - Maximum interval between consecutive SPF calculations
second-wait       - Delay between first and second SPF calculation
```

Multiple locations can be created in the **optimal-route-reflection** context, as follows. Each location is identified through a location ID [1..255], and contains a primary IP address and, optionally, a secondary IP address and a tertiary IP address, for redundancy reasons. These addresses must correspond to loopback or system IP addresses of routers participating in the IGP protocols, and are used as the starting point (or seed) for the SPF calculation. Because all clients in the same location receive the same optimal path for that location, these addresses must be close to the clients in that part of the network.

```
*[ex:/configure router "Base" bgp optimal-route-reflection location 1]
A:admin@RR-5# ?

apply-groups      - Apply a configuration group at this level
apply-groups-exclude - Exclude a configuration group at this level
primary-ip-address - Primary IPv4 address of the reference location for ORR
primary-ipv6-address - Primary IPv6 address of the reference location for ORR
secondary-ip-address - Secondary IPv4 address of reference location for ORR
secondary-ipv6-address - Secondary IPv6 address of reference location for ORR
tertiary-ip-address - Tertiary IPv4 address of the reference location for ORR
tertiary-ipv6-address - Tertiary IPv6 address of the reference location for ORR
```

The locations are then referred to with the **cluster** command (residing in the BGP group or neighbor context) through the **orr-location** argument, as follows.

```
*[ex:/configure router "Base" bgp group "IBGP-1"]
A:admin@RR-5# cluster ?

cluster

allow-local-fallback - Allow fallback to RR topology location
cluster-id           - Route reflector cluster ID
orr-location       - Optimal route reflection location for the cluster

*[ex:/configure router "Base" bgp neighbor "192.0.2.3"]
A:admin@RR-5# cluster ?

cluster

allow-local-fallback - Allow fallback to RR topology
cluster-id           - Route reflector cluster ID
orr-location       - Optimal route reflection location for the cluster
```

The location ID is referred to in the **orr-location** argument of the **cluster** command. Typically, the **cluster** command applies to a BGP peer group; all neighbors in that group share the same location ID, unless the **cluster** command applies at a neighbor level. The **allow-local-fallback** option allows the RR to advertise

the best reachable BGP path using its own location, but only when no BGP routes are reachable for some location. Otherwise, no path would be advertised to the clients in that location.

Properties

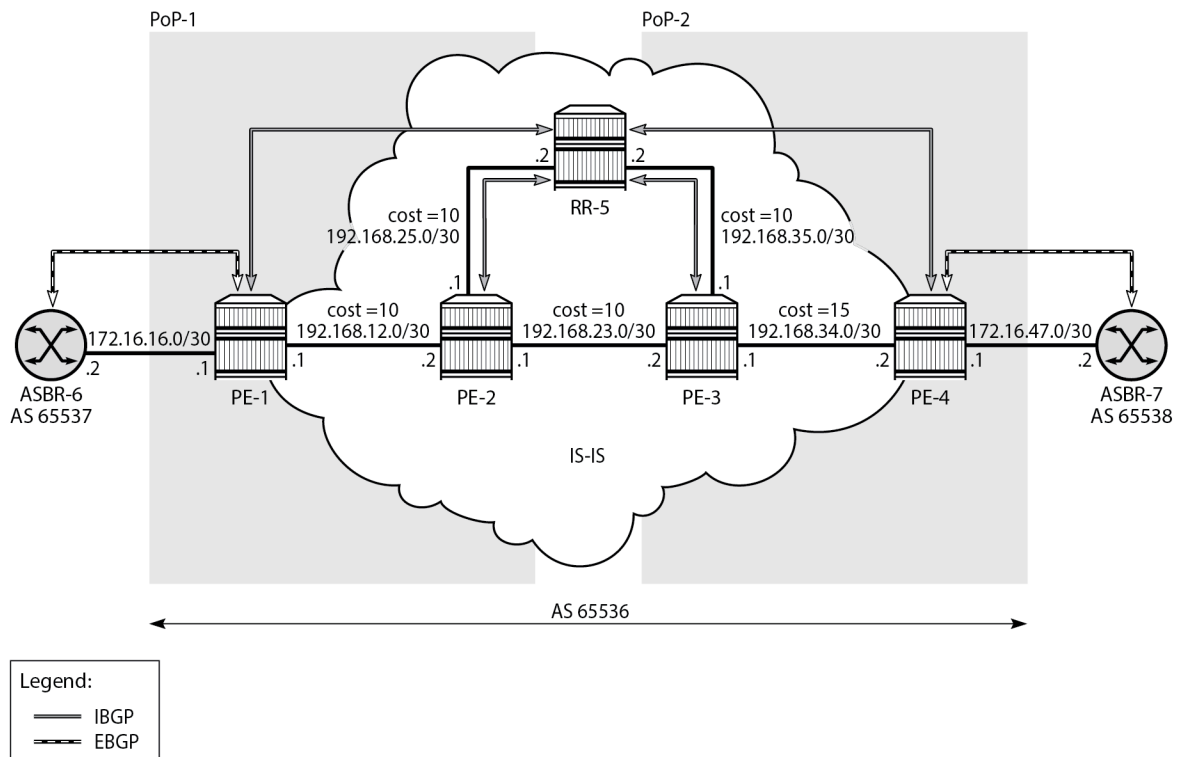
The following properties apply to ORR in SR OS:

- ORR is supported in the Base router BGP instance.
- ORR is supported for the IPv4, label-IPv4, label-IPv6, VPN-IPv4, and VPN-IPv6 address families.
- ORR is supported with add-paths, meaning that add-paths advertised to ORR clients are also ORR location-based.

Configuration

Figure 102: Example non-hierarchical networking using IS-IS shows the example topology. IS-IS is used as the IGP for AS 65536, with RR-5 taking the role of the route reflector for clients PE-1 to PE-4. Additionally, ASBR-6 in AS 65537 peers with PE-1, and ASBR-7 in AS 65538 peers with PE-4.

Figure 102: Example non-hierarchical networking using IS-IS



26682

The initial configuration on all nodes includes:

- Cards, MDAs, and ports

- Router interfaces
- IS-IS as IGP on all interfaces within AS 65536, in a non-hierarchical way (alternatively, OSPF can be used), and traffic engineering enabled

The basic IS-IS configuration is very similar for all routers, including the route reflector. The RR-5 configuration is as follows:

```
# on RR-5:
configure {
  router "Base" {
    isis 0 {
      admin-state enable
      traffic-engineering true
      area-address [49.0001]
      interface "int-RR-5-PE-2" {
        interface-type point-to-point
      }
      interface "int-RR-5-PE-3" {
        interface-type point-to-point
      }
      interface "system" {
      }
    }
  }
}
```

Route reflection without ORR

RR-5 peers with clients PE-1 to PE-4, and because RR-5 is the route reflector, the **cluster** command is added, defining the cluster ID attribute value to use. The configuration for RR-5 is as follows:

```
# on RR-5:
configure {
  router "Base" {
    autonomous-system 65536
    bgp {
      loop-detect discard-route
      split-horizon true
      group "IBGP" {
        peer-as 65536
        cluster {
          cluster-id 192.0.2.5
        }
      }
      neighbor "192.0.2.1" {
        group "IBGP"
      }
      neighbor "192.0.2.2" {
        group "IBGP"
      }
      neighbor "192.0.2.3" {
        group "IBGP"
      }
      neighbor "192.0.2.4" {
        group "IBGP"
      }
    }
  }
}
```

PE-1 belongs to the cluster defined in the route reflector, so it does not need to be fully meshed with the other routers in the area; peering with the route reflectors in the area is sufficient for PE-1 to receive

updates. Typically, two route reflectors are provisioned for redundancy, but that does not apply in this example. PE-1 also peers with ASBR-6 in AS 65537 through EBGP, so the PE-1 configuration is as follows:

```
# on PE-1:
configure {
  router "Base" {
    autonomous-system 65536
    bgp {
      loop-detect discard-route
      split-horizon true
      group "EBGP" {
      }
      group "IBGP" {
        next-hop-self true
        peer-as 65536
      }
      neighbor "172.16.16.2" {
        group "EBGP"
        peer-as 65537
        ebgp-default-reject-policy {
          import false
        }
      }
      neighbor "192.0.2.5" {
        group "IBGP"
      }
    }
  }
}
```

PE-2 and PE-3 only peer with the route reflector. Their configuration is the same:

```
# on PE-2, PE-3:
configure {
  router "Base" {
    autonomous-system 65536
    bgp {
      loop-detect discard-route
      split-horizon true
      group "IBGP" {
        peer-as 65536
      }
      neighbor "192.0.2.5" {
        group "IBGP"
      }
    }
  }
}
```

PE-4 also belongs to the IBGP cluster defined in the route reflector and PE-4 peers with ASBR-7 in AS 65538. The PE-4 configuration is similar to the configuration of PE-1.

Loopback address 10.1.11.1/24 is configured on ASBR-8 in AS 65540 (not shown in the example topology). ASBR-8 exports prefix 10.1.11.0/24 to its EBGP peers ASBR-6 in AS 65537 and ASBR-7 in AS 65538. ASBR-6 advertises prefix 10.1.11.0/24 to router PE-1; ASBR-7 advertises the same prefix to router PE-4.

RR-5 receives IBGP updates from PE-1 and PE-4, and selects the best path based on its own position in the topology. The IGP cost from RR-5 to PE-1 is 20, and the cost from RR-5 to PE-4 is 25, so RR-5 selects the BGP path with next hop 192.0.2.1.

```
[/]
A:admin@RR-5# show router bgp routes
=====
```

```

BGP Router ID:192.0.2.5      AS:65536      Local AS:65536
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  10.1.11.0/24              100        None
      192.0.2.1              None        20
      65537 65540             -
*i    10.1.11.0/24              100        None
      192.0.2.4              None        25
      65538 65540             -
-----
Routes : 2
=====

```

RR-5 reflects the path with next hop 192.0.2.1 to all clients except PE-1, because PE-1 is the client where the path was learned from).

For prefix 10.1.11.0/24, PE-1 received an EBGP route from ASBR-6 in AS 65537 with next hop 172.16.16.2 and no IBGP route from RR-5:

```

[/]
A:admin@PE-1# show router bgp routes
=====
BGP Router ID:192.0.2.1      AS:65536      Local AS:65536
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  10.1.11.0/24              None        None
      172.16.16.2           None        0
      65537 65540             -
-----
Routes : 1
=====

```

As a result, traffic offered to PE-1 for destination 10.1.11.0/24 is routed to ASBR-6, as follows:

```

[/]
A:admin@PE-1# show router route-table protocol bgp
=====
Route Table (Router: Base)
=====

```

```

Dest Prefix[Flags]                Type  Proto  Age      Pref
Next Hop[Interface Name]          Metric
-----
10.1.11.0/24                      Remote BGP    00h04m15s 170
    172.16.16.2                    0
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

PE-2 received an IBGP route for prefix 10.1.11.0/24 with next hop 192.0.2.1 from RR-5:

```

[/]
A:admin@PE-2# show router bgp routes
=====
BGP Router ID:192.0.2.2      AS:65536      Local AS:65536
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)    Path-Id    IGP Cost
      As-Path              Label
-----
u*>i  10.1.11.0/24            100        None
      192.0.2.1          None        10
      65537 65540                -
-----
Routes : 1
=====

```

Traffic offered to PE-2 for destination 10.1.11.0/24 is routed to PE-1, as follows:

```

[/]
A:admin@PE-2# show router route-table protocol bgp
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
Next Hop[Interface Name]          Metric
-----
10.1.11.0/24                      Remote BGP    00h17m22s 170
    192.168.12.1                    10
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```


Likewise, PE-3 received an IBGP route for prefix 10.1.11.0/24 with next hop 192.0.2.1 from RR-5:

```
[/]
A:admin@PE-3# show router bgp routes
=====
BGP Router ID:192.0.2.3      AS:65536      Local AS:65536
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  10.1.11.0/24              100        None
      192.0.2.1              None        20
      65537 65540              -          -
-----
Routes : 1
=====
```

Traffic offered to PE-3 for destination 10.1.11.0/24 is routed via the interface address 192.168.23.1 on PE-2, as follows:

```
[/]
A:admin@PE-3# show router route-table protocol bgp
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age           Pref
  Next Hop[Interface Name]           Metric
-----
10.1.11.0/24                Remote BGP    00h10m26s  170
  192.168.23.1                20
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

For prefix 10.1.11.0/24, PE-4 received an EBGP route from ASBR-7 with next hop 172.16.47.2 and an IBGP route from RR-5 with next hop 192.0.2.1, as follows. EBGP routes are preferred over IBGP routes.

```
[/]
A:admin@PE-4# show router bgp routes
=====
BGP Router ID:192.0.2.4      AS:65536      Local AS:65536
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
```

```

BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
      Nexthop (Router)                     Path-Id   IGP Cost
      As-Path                               Label
-----
u*>i 10.1.11.0/24                             None     None
      172.16.47.2                             None     0
      65538 65540                               -
*i   10.1.11.0/24                             100     None
      192.0.2.1                               None     35
      65537 65540                               -
-----
Routes : 2
=====

```

The used route is the EBGP route from ASBR-7, so the traffic offered to PE-4 for destination 10.1.11.0/24 is routed to ASBR-7, as follows:

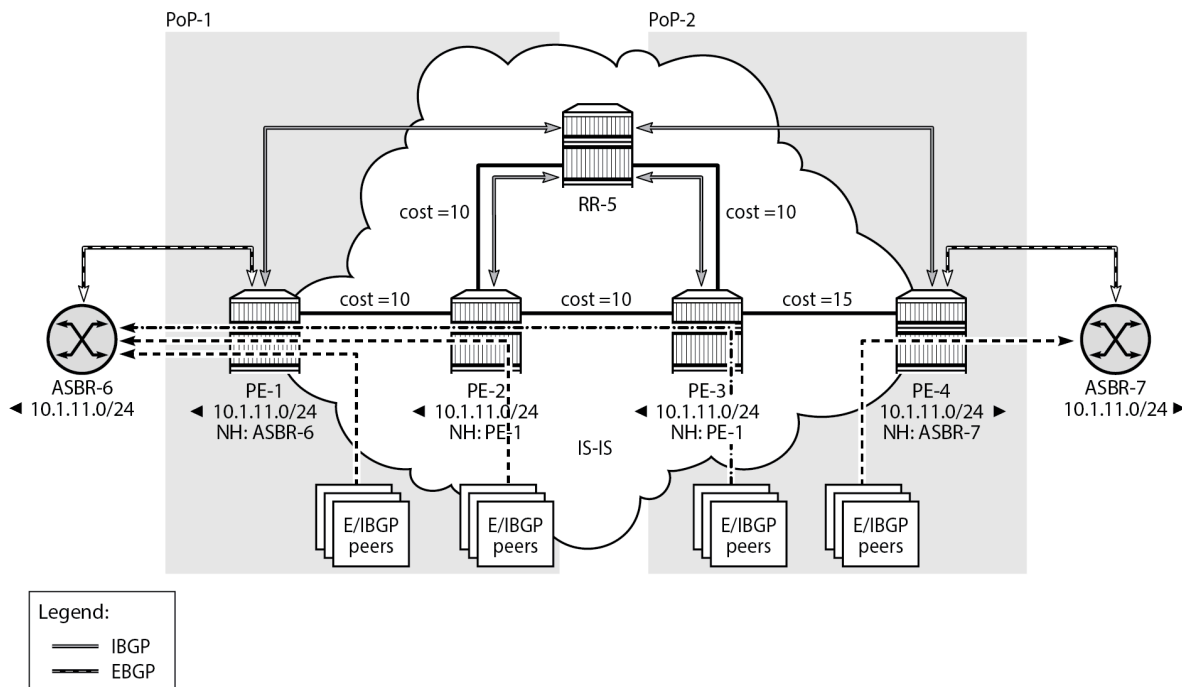
```

[/]
A:admin@PE-4# show router route-table protocol bgp
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]           Type  Proto  Age      Pref
      Next Hop[Interface Name]           Metric
-----
10.1.11.0/24                 Remote BGP    00h18m08s 170
      172.16.47.2                 0
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

This is summarized in [Figure 103: Suboptimal route reflection](#). Ultimately, PE-1 only has one path, and so do PE-2 and PE-3. PE-4 has two paths, but by default prefers the EBGP learned path over the IBGP learned path. The routing is suboptimal on PE-3, where the IGP cost to PE-1 is 20 and the IGP cost to PE-4 is 15.

Figure 103: Suboptimal route reflection



26683

Route reflection with ORR

For implementing ORR using the non-hierarchical topology from [Figure 103: Suboptimal route reflection](#) the route reflector RR-5 defines two locations in the **optimal-route-reflection** context. The primary IP address for location 1 is the PE-1 system IP address 192.0.2.1; the primary IP address for location 2 is loopback address 192.0.2.44 on PE-4 and the secondary IP address is loopback address 192.0.2.33 on PE-3. These addresses are used as the starting point for the SPF run. The ORR locations 1 and 2 are then referred to from within the group definitions through the **cluster** command. The overall BGP configuration of RR-5 is as follows:

```
# on RR-5
configure {
  router "Base" {
    autonomous-system 65536
    bgp {
      loop-detect discard-route
      split-horizon true
      optimal-route-reflection {
        spf-wait {
          max-wait 1
          initial-wait 1
          second-wait 1
        }
        location 1 {
          primary-ip-address 192.0.2.1
        }
        location 2 {
```

```

        primary-ip-address 192.0.2.44      # loopback address on PE-4
        secondary-ip-address 192.0.2.33    # loopback address on PE-3
    }
}
group "IBGP-1" {
    peer-as 65536
    cluster {
        cluster-id 192.0.2.5
        orr-location 1
        allow-local-fallback true
    }
}
group "IBGP-2" {
    peer-as 65536
    cluster {
        cluster-id 192.0.2.5
        orr-location 2
        allow-local-fallback true
    }
}
neighbor "192.0.2.1" {
    group "IBGP-1"
}
neighbor "192.0.2.2" {
    group "IBGP-1"
}
neighbor "192.0.2.3" {
    group "IBGP-2"
}
neighbor "192.0.2.4" {
    group "IBGP-2"
}
}

```

No changes are required in the BGP clients.

ASBR-6 advertises prefix 10.1.11.0/24 to router PE-1; ASBR-7 advertises the same prefix to router PE-4. RR-5 receives the updates from PE-1 and PE-4, and now performs two SPF runs because two locations are used. The first SPF run uses the 192.0.2.1 address of PE-1 as the starting point for the first location, selects the path via PE-1 as the best path, and reflects that path to the remaining peers in the first location. The second SPF run uses the 192.0.2.44 loopback address of PE-4 as the starting point for the second location, selects the path via PE-4 as the best path, and reflects that path to the remaining peers in the second location.

In comparison with the previous scenario, there only is a change in the routing for this prefix on PE-3. RR-5 reflects the route with next hop 192.0.2.4 to PE-3.

```

[/]
A:admin@PE-3# show router bgp routes
=====
BGP Router ID:192.0.2.3      AS:65536      Local AS:65536
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                       Path-Id    IGP Cost

```

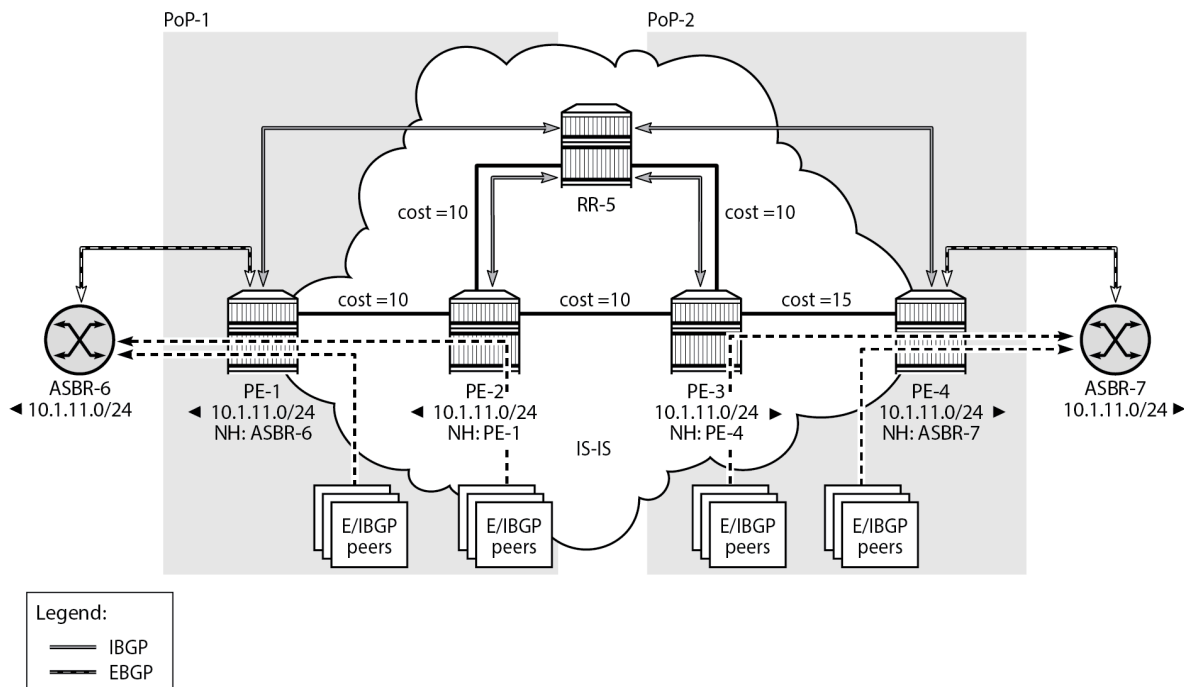
As-Path	Label
-----	-----
u*>i 10.1.11.0/24	100 None
192.0.2.4	None 15
65538 65540	-
-----	-----
Routes : 1	
=====	=====

Traffic offered to PE-3 for destination 10.1.11.0/24 has next hop PE-4 and is routed via the interface address 192.168.34.2 on PE-4, as follows:

```
[/]
A:admin@PE-3# show router route-table protocol bgp
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]                Metric
-----
10.1.11.0/24                      Remote BGP      00h02m06s  170
  192.168.34.2                      15
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```

This is summarized in [Figure 104: Optimal route reflection](#).

Figure 104: Optimal route reflection



26684

The following command provides the IGP distances for the configured reference points to all available BGP peers and all detected BGP next hops on the route reflector.

```
[/]
A:admin@RR-5# show router bgp optimal-route-reflection bgp-nh-info

=====
ORR BGP-NH Table (Router: Base)
=====

Location 1:
  Primary       : 192.0.2.1 [active]
  Secondary     : -
  Tertiary      : -
  Primary-ipv6  : -
  Secondary-ipv6 : -
  Tertiary-ipv6 : -

Location 2:
  Primary       : 192.0.2.44 [active]
  Secondary     : 192.0.2.33
  Tertiary      : -
  Primary-ipv6  : -
  Secondary-ipv6 : -
  Tertiary-ipv6 : -

Age           : 00h02m55s
Spf wait      : 1
Initial wait  : 1
Second wait   : 1

-----
Next Hop
```

Loc	Dest-Prefix	DB-Source	Type	Proto	Metric	Pref

192.0.2.1						
1	192.0.2.1/32	IGP	Local	Local	0	0
2	192.0.2.1/32	IGP	Remote	ISIS	35	18
192.0.2.4						
1	192.0.2.4/32	IGP	Remote	ISIS	35	18
2	192.0.2.4/32	IGP	Local	Local	0	0

No. of BGP-NHs: 2						
=====						

Conclusion

BGP optimal route reflection allows operators to optimize traffic streams through their network, even when the route reflector is placed out-of-path, for example in datacenters, thereby reducing the OPEX and CAPEX of route reflector deployment.

BGP Prefix Limit per Address Family

This chapter provides information about BGP prefix limit per address family.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written based on SR OS Release 15.0.R1, but the MD-CLI in the current edition is based on SR OS Release 22.10.R1.

Overview

A BGP per address family prefix limit can be defined to control the number of prefixes learned per neighbor or per group of neighbors in the base router or in a VPRN. This feature allows ISPs to secure their network from misbehaving or misconfigured peers. This feature can also be used to enforce the terms of a service contract.

[Table 2: Supported address families for BGP prefix limit](#) lists the address families for which a prefix limit can be defined in the base router and in VPRNs.

Table 2: Supported address families for BGP prefix limit

Address family	Base router	VPRN
ipv4	X	X
ipv6	X	X
mcast-ipv4	X	X
mcast-ipv6	X	X
flow-ipv4	X	X
flow-ipv6	X	X
label-ipv4	X	X
label-ipv6	X	–
vpn-ipv4	X	–
vpn-ipv6	X	–
mvpn-ipv4	X	–

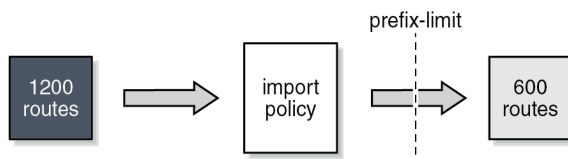
Address family	Base router	VPRN
mvpn-ipv6	X	–
mcast-vpn-ipv4	X	–
mcast-vpn-ipv6	X	–
flow-vpn-ipv4	X	–
flow-vpn-ipv6	X	–
sr-policy-ipv4	X	–
sr-policy-ipv6	X	–
l2-vpn	X	–
mdt-safi	X	–
ms-pw	X	–
route-target	X	–
evpn	X	–
bgp-ls	X	–

If the number of received routes from a peer exceeds a defined per address family limit, the BGP session is torn down, the state is changed to disabled, the routes learned from that peer are deleted, and the RIB and FIB are recalculated. With the **log-only** option enabled, the BGP session is not torn down and no routes are deleted. An SNMP trap message is issued when exceeding the per address family threshold (default: 90%), and the per address family prefix limit.

Re-establishing the BGP session with the peer requires a manual intervention, or use of the **idle-timeout** option. The idle-timeout option defines the time in minutes after which the system attempts to re-establish the BGP session.

The **post-import** option indicates that the limit should be applied only to the routes accepted by import policies, as shown in [Figure 105: Post-import option](#). A route rejected by an import policy will not be counted when checking against the prefix limit. Not specifying the post-import option results in routes being counted and verified against the prefix limit when they are received, before the import policy is executed, and might lead to BGP sessions being torn down unexpectedly.

Figure 105: Post-import option



26848

BGP sessions will be torn down as soon as one of the address family prefix limits is exceeded, even when the limit for the other address family is not yet exceeded. In cases where this is important, consider defining two BGP sessions between two peers; the first using IPv4 for its transport, and the second using IPv6. In this way, an IPv4 limit being exceeded will not lead to IPv6 prefixes being affected.



Note: A VPN route carrying a route-target (for example, VPN-IPv4, VPN-IPv6, L2-VPN, MVPN-IPV4, MVPN-IPv6) might not be retained in the RIB-IN if it is not imported by any service. If a VPN route is not stored in the RIB-IN, it is not counted and not checked against the prefix limit for its associated address family. If **mp-bgp-keep true** is configured, or the router is a route reflector (using the **cluster** command) or an ASBR in an inter-AS VPRN model B, then the VPN-IP route is always stored.

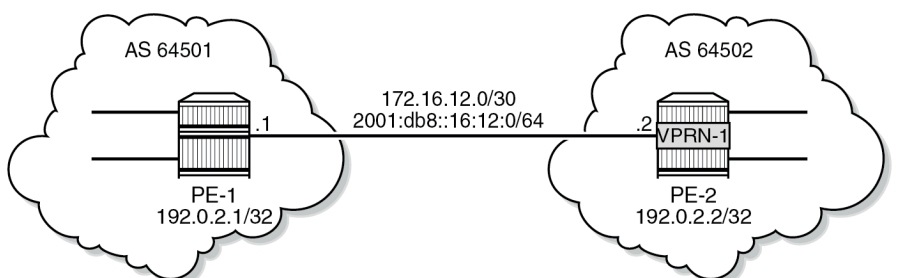
Configuration

Figure 106: Example topology shows the example topology. PE-1 in AS 64501 peers with VPRN-1 hosted by PE-2 in AS 64502.

Two scenarios are considered:

- Prefix limit without post-import option
- Prefix limit with post-import option

Figure 106: Example topology



26849

Prefix limit without post-import option

PE-1 peers with VPRN-1 on PE-2, where IP prefix limit is configured in the BGP group toward PE-1: the IPv4 prefix limit is 10, the threshold is 50%, and the idle-timeout is 1 minute; the IPv6 prefix limit is 10, the threshold 80%, and the idle-timeout is 4 minutes, as follows:

```
# on PE-2:
configure {
  service {
    vprn "VPRN-1" {
      admin-state enable
      description "VPRN with BGP prefix limit"
      service-id 1
      customer "1"
      autonomous-system 64502
      bgp-ipvprn {
        mpls {
          admin-state enable
          route-distinguisher "64502:1"
        }
      }
    }
  }
  bgp {
    loop-detect discard-route
  }
}
```

```

split-horizon true
group "EBGP-IPv4" {
  peer-as 64501
  family {
    ipv4 true
  }
  import {
    policy ["import-10.1-ranges"]
  }
  prefix-limit ipv4 {
    maximum 10
    threshold 50
    idle-timeout 1
  }
}
group "EBGP-IPv6" {
  peer-as 64501
  family {
    ipv6 true
  }
  import {
    policy ["import-ipv6-88-ranges"]
  }
  prefix-limit ipv6 {
    maximum 10
    threshold 80
    idle-timeout 4
  }
}
neighbor "172.16.12.1" {
  group "EBGP-IPv4"
}
neighbor "2001:db8::16:12:1" {
  group "EBGP-IPv6"
}
}
interface "int-VPRN-1onPE-2-PE-1" {
  ipv4 {
    primary {
      address 172.16.12.2
      prefix-length 30
    }
  }
  sap 1/1/c2/1:1 {
  }
  ipv6 {
    address 2001:db8::16:12:2 {
      prefix-length 126
    }
  }
}
}
}

```

The debug configuration (in classic CLI) is as follows:

```

# on PE-2:
debug
  router service-name "VPRN-1"
    bgp
      packets neighbor 172.16.12.1
      events neighbor 172.16.12.1
    exit
  exit

```

The debug output is sent to the log with log-id "log-1", as follows:

```
# on PE-2:
configure {
  log {
    log-id "log-1" {
      source {
        debug true
      }
      destination {
        memory {
        }
      }
    }
  }
}
```

Initially, the number of IPv4 routes received from PE-1 is below the threshold, and PE-1 gradually injects more IPv4 routes into VPRN-1 on PE-2. The following is a snapshot where three IPv4 routes and four IPv6 routes are received and active in PE-2:

```
[/]
A:admin@PE-2# show router 1 bgp summary
=====
BGP Router ID:192.0.2.2      AS:64502      Local AS:64502
=====
BGP Admin State      : Up          BGP Oper State      : Up
Total Peer Groups    : 2            Total Peers          : 2
Current Internal Groups : 2          Max Internal Groups : 2
Total BGP Paths       : 7            Total Path Memory    : 2480

Total IPv4 Remote Rts : 3          Total IPv4 Rem. Active Rts : 3
Total IPv6 Remote Rts : 4          Total IPv6 Rem. Active Rts : 4
Total IPv4 Backup Rts : 0            Total IPv6 Backup Rts : 0
Total LblIpv4 Rem Rts : 0            Total LblIpv4 Rem. Act Rts : 0
Total LblIpv6 Rem Rts : 0            Total LblIpv6 Rem. Act Rts : 0
Total LblIpv4 Bkp Rts : 0            Total LblIpv6 Bkp Rts : 0
Total Suppressed Rts : 0            Total Hist. Rts      : 0
Total Decay Rts      : 0

Total McIPv4 Remote Rts : 0          Total McIPv4 Rem. Active Rts: 0
Total McIPv6 Remote Rts : 0          Total McIPv6 Rem. Active Rts: 0

Total FlowIpv4 Rem Rts : 0          Total FlowIpv4 Rem Act Rts : 0
Total FlowIpv6 Rem Rts : 0          Total FlowIpv6 Rem Act Rts : 0
Total FlowVpvn4 Rem Rts : 0         Total FlowVpvn4 Rem Act Rts : 0
Total FlowVpvn6 Rem Rts : 0         Total FlowVpvn6 Rem Act Rts : 0
Total Link State Rem Rts: 0          Total Link State Rem Act Rts: 0
Total SrPlcyIpv4 Rem Rts: 0          Total SrPlcyIpv4 Rem Act Rts: 0
Total SrPlcyIpv6 Rem Rts: 0          Total SrPlcyIpv6 Rem Act Rts: 0

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
          AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
          PktSent OutQ
-----
172.16.12.1
          64501      8    0 00h01m54s 3/3/0 (IPv4)
          7      0
2001:db8::16:12:1
```

```
64501      8      0 00h01m45s 4/4/0 (IPv6)
          7      0
```

The following three BGP IPv4 routes are received by VPRN-1 on PE-2 and they are all active:

```
[/]
A:admin@PE-2# show router 1 bgp routes
=====
BGP Router ID:192.0.2.2      AS:64502      Local AS:64502
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
     Nexthop (Router)                       Path-Id    IGP Cost
     As-Path                                  Label
-----
u*>i  10.1.0.0/24                               None       None
     172.16.12.1                               None       0
     64501                                       -
u*>i  10.1.1.0/24                               None       None
     172.16.12.1                               None       0
     64501                                       -
u*>i  10.1.2.0/24                               None       None
     172.16.12.1                               None       0
     64501                                       -
-----
Routes : 3
=====
```

When the sixth BGP IPv4 route is received, the threshold value (50% of 10 is 5) is exceeded, and a message is generated and sent to log "99", as follows:

```
[/]
A:admin@PE-2# show log log-id "99"
=====
Event Log 99 log-name 99
=====
Description : Default System Log
Memory Log contents [size=500 next event=110 (not wrapped)]

109 2022/11/25 15:49:27.411 CET MINOR: BGP #2035 vprn1 Peer 2: 172.16.12.1
"(ASN 64501) VR 2: Group EBGp-IPv4: Peer 172.16.12.1: number of routes learned has exceeded 50
percentage of the configured maximum (10) for ipv4 family"
```

Likewise, when the ninth IPv6 route is received, the threshold value (80% of 10 is 8) is exceeded, the following message is added to log 99:

```
[/]
A:admin@PE-2# show log log-id "99"
---snip---

110 2022/11/25 15:50:04.412 CET MINOR: BGP #2035 vprn1 Peer 2: 2001:db8::16:12:1
```

```
"(ASN 64501) VR 2: Group EBGp-IPv6: Peer 2001:db8::16:12:1: number of routes learned has exceeded 80 percentage of the configured maximum (10) for ipv6 family"
```

When the eleventh BGP IPv4 route is received, the configured maximum number of BGP routes for IPv4 is exceeded. The BGP session state changes from *established* to *idle* and the peer is notified, as indicated in the following debug log:

```
[/]
A:admin@PE-2# show log log-id "log-1"

=====
Event Log 1 log-name log-1
=====
Description : (Not Specified)
Memory Log contents [size=100 next event=65 (not wrapped)]

64 2022/11/25 15:53:59.417 CET MINOR: DEBUG #2001 vprn1 Peer 2: 172.16.12.1
"Peer 2: 172.16.12.1: NOTIFICATION
Peer 2: 172.16.12.1 - Send BGP NOTIFICATION: Code = 6 (CEASE) Subcode = 1 (Maximum prefixed
reached)
  Data Length = 7  Data: 0x0 0x1 0x1 0x0 0x0 0x0 0xa
"

63 2022/11/25 15:53:59.417 CET MINOR: DEBUG #2001 vprn1 BGP
"BGP: STATE
Peer 2: 172.16.12.1 - Change State from ESTABLISHED to IDLE due to MAXPREFIX_EXCEEDED
"

62 2022/11/25 15:53:59.417 CET MINOR: DEBUG #2001 vprn1 Peer 2: 172.16.12.1
"Peer 2: 172.16.12.1: UPDATE
Peer 2: 172.16.12.1 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 20
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 6 AS Path:
    Type: 2 Len: 1 < 64501 >
  Flag: 0x40 Type: 3 Len: 4 Nexthop: 172.16.12.1
  NLRI: Length = 44
    10.1.0.0/24
    10.1.1.0/24
    10.1.10.0/24
    10.1.2.0/24
    10.1.3.0/24
    10.1.4.0/24
    10.1.5.0/24
    10.1.6.0/24
    10.1.7.0/24
    10.1.8.0/24
    10.1.9.0/24
"
```

The BGP session is torn down and the corresponding state is disabled, as follows:

```
[/]
A:admin@PE-2# show router 1 bgp summary

=====
BGP Router ID:192.0.2.2      AS:64502      Local AS:64502
=====
BGP Admin State      : Up      BGP Oper State      : Up
Total Peer Groups    : 2      Total Peers         : 2
Current Internal Groups : 1      Max Internal Groups : 2
Total BGP Paths      : 6      Total Path Memory   : 2120
```

```

Total IPv4 Remote Rts : 0
Total IPv6 Remote Rts : 10
Total IPv4 Backup Rts : 0
Total LblIpv4 Rem Rts : 0
Total LblIpv6 Rem Rts : 0
Total LblIpv4 Bkp Rts : 0
Total Supressed Rts : 0
Total Decay Rts : 0

Total McIPv4 Remote Rts : 0
Total McIPv6 Remote Rts : 0

Total FlowIpv4 Rem Rts : 0
Total FlowIpv6 Rem Rts : 0
Total FlowVpvn4 Rem Rts : 0
Total FlowVpvn6 Rem Rts : 0
Total Link State Rem Rts: 0
Total SrPlcyIpv4 Rem Rts: 0
Total SrPlcyIpv6 Rem Rts: 0

Total IPv4 Rem. Active Rts : 0
Total IPv6 Rem. Active Rts : 10
Total IPv6 Backup Rts : 0
Total LblIpv4 Rem. Act Rts : 0
Total LblIpv6 Rem. Act Rts : 0
Total LblIpv6 Bkp Rts : 0
Total Hist. Rts : 0

Total McIPv4 Rem. Active Rts: 0
Total McIPv6 Rem. Active Rts: 0

Total FlowIpv4 Rem Act Rts : 0
Total FlowIpv6 Rem Act Rts : 0
Total FlowVpvn4 Rem Act Rts : 0
Total FlowVpvn6 Rem Act Rts : 0
Total Link State Rem Act Rts: 0
Total SrPlcyIpv4 Rem Act Rts: 0
Total SrPlcyIpv6 Rem Act Rts: 0
    
```

=====
BGP Summary
=====

Legend : D - Dynamic Neighbor
=====

Neighbor
Description

	AS	PktRcvd PktSent	InQ OutQ	Up/Down	State	Rcv/Act/Sent (Addr Family)
172.16.12.1	64501	0	0	00h00m10s	Disabled	
2001:db8::16:12:1	64501	25	0	00h09m11s	10/10/0	(IPv6)
		22	0			

Also, this event is recorded in the system logs, as follows:

```

[/]
A:admin@PE-2# show log log-id "99"
---snip---

137 2022/11/25 15:55:32.424 CET WARNING: BGP #2012 vprn1 Peer 2: 172.16.12.1
"(ASN 64501) Peer 2: 172.16.12.1: Closing connection: VR 2: Group EBGp-IPv4: Peer 172.16.12.1
not enabled or not in configuration"

136 2022/11/25 15:55:32.418 CET WARNING: BGP #2005 vprn1 Peer 2: 172.16.12.1
"(ASN 64501) VR 2: Group EBGp-IPv4: Peer 172.16.12.1: sending notification: code CEASE subcode
MAX_PFX_RCHD"

135 2022/11/25 15:55:32.418 CET WARNING: BGP #2039 vprn1 Peer 2: 172.16.12.1
"(ASN 64501) VR 2: Group EBGp-IPv4: Peer 172.16.12.1: moved from higher state ESTABLISHED to
lower state IDLE due to event MAXPREFIX_EXCEEDED"
    
```

When the idle-timeout expires, in this case, after one minute, the system tries to re-establish the session. With the BGP session re-established, the peer starts re-advertising its routes. As long as the number of received routes in VPRN-1 on PE-2 is lower than or equal to the limit, the session is maintained. In this example, the maximum number of received IPv4 routes is 10 and the maximum number of received IPv6 routes is 10.

Prefix limit with post-import option

Use caution when using the prefix limit in combination with import policies. By default, the routes are counted when receiving them, that is, before the import policy is enforced. To postpone the prefix limit check, the **post-import** option must be used.

The BGP configuration for VPRN-1 on PE-2 has **post-import** enabled, as follows:

```
# on PE-2:
configure {
  service {
    vprn "VPRN-1" {
      admin-state enable
      description "VPRN with BGP prefix limit"
      service-id 1
      customer "1"
      autonomous-system 64502
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "64502:1"
        }
      }
    }
  }
  bgp {
    loop-detect discard-route
    split-horizon true
    group "EBGP-IPv4" {
      peer-as 64501
      import {
        policy ["import-10.1-ranges"]
      }
      prefix-limit ipv4 {
        maximum 10
        threshold 50
        idle-timeout 1
        post-import true
      }
    }
    group "EBGP-IPv6" {
      peer-as 64501
      family {
        ipv6 true
      }
      import {
        policy ["import-ipv6-88-ranges"]
      }
      prefix-limit ipv6 {
        maximum 10
        threshold 80
        idle-timeout 4
      }
    }
    neighbor "172.16.12.1" {
      group "EBGP-IPv4"
    }
    neighbor "2001:db8::16:12:1" {
      group "EBGP-IPv6"
    }
  }
  interface "int-VPRN-1onPE-2-PE-1" {
    ipv4 {
      primary {
```



```

        address 172.16.12.2
        prefix-length 30
    }
}
sap 1/1/c2/1:1 {
}
ipv6 {
    address 2001:db8::16:12:2 {
        prefix-length 126
    }
}
}
}
}

```

The *import-10.1-ranges* policy is defined as follows:

```

# on PE-2:
configure {
    policy-options {
        prefix-list "pfx-10.1-ranges" {
            prefix 10.1.0.0/16 type longer {
            }
        }
    }
    policy-statement "import-10.1-ranges" {
        entry 10 {
            from {
                prefix-list ["pfx-10.1-ranges"]
            }
            action {
                action-type accept
            }
        }
        default-action {
            action-type reject
        }
    }
}
}

```

When twelve IPv4 routes are received over this BGP session, six in the 10.1.0.0/16 range and six in the 10.2.0.0/16 range, then only the six routes in the 10.1.0.0/16 range are accepted and active in the routing table, as follows:

```

[/]
A:admin@PE-2# show router 1 route-table protocol bgp
=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]          Type   Proto   Age           Pref
-----
10.1.0.0/24
172.16.12.1                        Remote BGP       00h02m07s    170
0
10.1.1.0/24
172.16.12.1                        Remote BGP       00h02m07s    170
0
10.1.2.0/24
172.16.12.1                        Remote BGP       00h02m07s    170
0
10.1.3.0/24
172.16.12.1                        Remote BGP       00h02m07s    170
0
10.1.4.0/24
172.16.12.1                        Remote BGP       00h02m07s    170
0
10.1.5.0/24
172.16.12.1                        Remote BGP       00h02m07s    170
0

```

```

-----
No. of Routes: 6
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

The BGP session remains established with twelve received routes and six of these being active, as follows:

```

[/]
A:admin@PE-2# show router 1 bgp summary
=====
BGP Router ID:192.0.2.2      AS:64502      Local AS:64502
=====
BGP Admin State      : Up          BGP Oper State      : Up
Total Peer Groups    : 2            Total Peers          : 2
Current Internal Groups : 2          Max Internal Groups  : 2
Total BGP Paths       : 7            Total Path Memory    : 2480

Total IPv4 Remote Rts : 12          Total IPv4 Rem. Active Rts : 6
Total IPv6 Remote Rts : 10          Total IPv6 Rem. Active Rts : 10
Total IPv4 Backup Rts : 0            Total IPv6 Backup Rts   : 0
Total LblIPv4 Rem Rts : 0            Total LblIPv4 Rem. Act Rts : 0
Total LblIPv6 Rem Rts : 0            Total LblIPv6 Rem. Act Rts : 0
Total LblIPv4 Bkp Rts : 0            Total LblIPv6 Bkp Rts   : 0
Total Suppressed Rts  : 0            Total Hist. Rts       : 0
Total Decay Rts       : 0

Total McIPv4 Remote Rts : 0          Total McIPv4 Rem. Active Rts: 0
Total McIPv6 Remote Rts : 0          Total McIPv6 Rem. Active Rts: 0

Total FlowIPv4 Rem Rts : 0          Total FlowIPv4 Rem Act Rts : 0
Total FlowIPv6 Rem Rts : 0          Total FlowIPv6 Rem Act Rts : 0
Total FlowVpnv4 Rem Rts : 0         Total FlowVpnv4 Rem Act Rts : 0
Total FlowVpnv6 Rem Rts : 0         Total FlowVpnv6 Rem Act Rts : 0
Total Link State Rem Rts: 0          Total Link State Rem Act Rts: 0
Total SrPlcyIPv4 Rem Rts: 0         Total SrPlcyIPv4 Rem Act Rts: 0
Total SrPlcyIPv6 Rem Rts: 0         Total SrPlcyIPv6 Rem Act Rts: 0

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
          AS PktRcvd InQ Up/Down State|Rcv/Act/Sent (Addr Family)
          PktSent OutQ
-----
172.16.12.1
          64501      18   0 00h06m14s 12/6/0 (IPv4)
          17       0
2001:db8::16:12:1
          64501      39   0 00h16m14s 10/10/0 (IPv6)
          36       0
-----

```

Without the **post-import** option, the session is torn down as soon as the number of received routes exceeds the configured prefix limit.

Conclusion

The BGP prefix limit per address family feature allows ISPs to protect their network from misbehaving or misconfigured peers, and can also be used to enforce the terms of a service contract.

BGP Remove-Private ASN

This chapter describes BGP Remove-Private ASN.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information and configuration in this chapter are based on SR OS Release 22.10.R2.

Overview

In some networks, the network operator may need to assign a private Autonomous System Number (ASN) to the BGP speakers in a region or domain. These private ASNs are taken from the following ranges defined by IANA:

- 64512 to 65534 inclusive, for 2-octet ASNs
- 4200000000 to 4294967294 inclusive, for 4-octet ASNs

In SR OS, the ASN numbers 65535 and 4294967295, which are reserved values, are also treated as private ASNs.

The **remove-private** command is required when routes originated by a BGP speaker with a private ASN need to be advertised into a public domain, such as the Internet, where private ASNs may not be unique. The functionality of the **remove-private** command in SR OS is as follows:

- When the **remove-private** command is configured for neighbor X, the stripping of private ASNs applies only to outbound routes advertised to neighbor X.
- The **remove-private** command supports the following three options, which can be configured standalone or combined:
 - The **limited true** option causes BGP to remove only the private ASNs until the first public ASN.
 - The **skip-peer-as true** option causes BGP to not remove a private ASN from the AS path attribute if that ASN is the same as the BGP peer ASN.
 - The **replace true** option replaces the private ASN with the ASN of the router, as configured in:
 - **local-as** if the router advertises routes to a peer covered by such a command, and not configured as **private**
 - **configure router autonomous-system** if there is no applicable **local-as** configuration in BGP and the router is not part of a confederation
 - **configure router bgp confederation** if the router advertises routes to an eBGP peer outside the confederation



Note:

The use of the **remove-private** command without the **replace true** option can make the AS path attribute shorter. This makes the route more preferable for the BGP decision process, which may not be the wanted outcome.



Note:

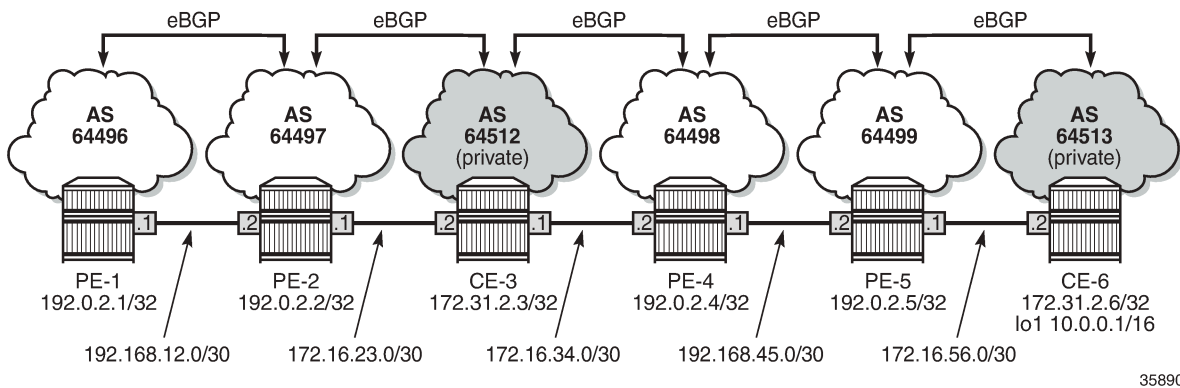
When **as-override** is enabled in the same session as **remove-private**, processing related to **remove-private** occurs first, followed by the processing related to **as-override**.

Configuration

Use case 1: Initial example topology

Figure 107: Use case 1 topology shows the initial example topology with six nodes in different ASs: CE-3 and CE-6 have a private ASN, whereas PE-1, PE-2, PE-4, and PE-5 have a public ASN.

Figure 107: Use case 1 topology



The initial configuration on the nodes includes:

- Cards, MDAs, ports
- Router interfaces
- eBGP between adjacent nodes for the IPv4 address family

The initial BGP configuration on PE-2 is as follows:

```
# on PE-2:
configure {
  router "Base" {
    bgp {
      split-horizon true
      group "eBGP" {
        family {
          ipv4 true
        }
      }
    }
    neighbor "172.16.23.2" {
      group "eBGP"
      peer-as 64512
    }
  }
}
```

```

    }
    neighbor "192.168.12.1" {
        group "eBGP"
        peer-as 64496
    }
}

```

CE-6 exports prefix 10.0.0.0/16. The configuration is as follows:

```

# on CE-6:
configure {
    policy-options {
        prefix-list "10.0.0.0/16" {
            prefix 10.0.0.0/16 type longer {
            }
        }
        policy-statement "export-prefix" {
            entry 10 {
                from {
                    prefix-list ["10.0.0.0/16"]
                }
                action {
                    action-type accept
                }
            }
        }
    }
}
router "Base" {
    autonomous-system 64513
    interface "int-CE-6-PE-5" {
        port 1/1/c1/2:100
        ipv4 {
            primary {
                address 172.16.56.2
                prefix-length 30
            }
        }
    }
    interface "lol" {
        loopback
        ipv4 {
            primary {
                address 10.0.0.1
                prefix-length 16
            }
        }
    }
    interface "system" {
        ipv4 {
            primary {
                address 172.31.2.6
                prefix-length 32
            }
        }
    }
}
bgp {
    split-horizon true
    group "eBGP" {
        family {
            ipv4 true
        }
    }
}
neighbor "172.16.56.1" {

```

```

        group "eBGP"
        peer-as 64499
        export {
            policy ["export-prefix"]
        }
    }
}

```

PE-2 receives the following BGP route for prefix 10.0.0.0/16 with public and private ASNs in the AS path: 64512 (private ASN of CE-3) – 64498 (public ASN of PE-4) – 64499 (public ASN of PE-5) – 64513 (private ASN of CE-6).

```

[/]
A:admin@PE-2# show router bgp routes 10.0.0.0/16
=====
BGP Router ID:192.0.2.2      AS:64497      Local AS:64497
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  10.0.0.0/16                None       None
      172.16.23.2             None       0
      64512 64498 64499 64513
-----
Routes : 1
=====

```

PE-2 adds its own public ASN (64497) to the AS path when it sends the BGP route to its neighbor PE-1. The following BGP route is received by PE-1:

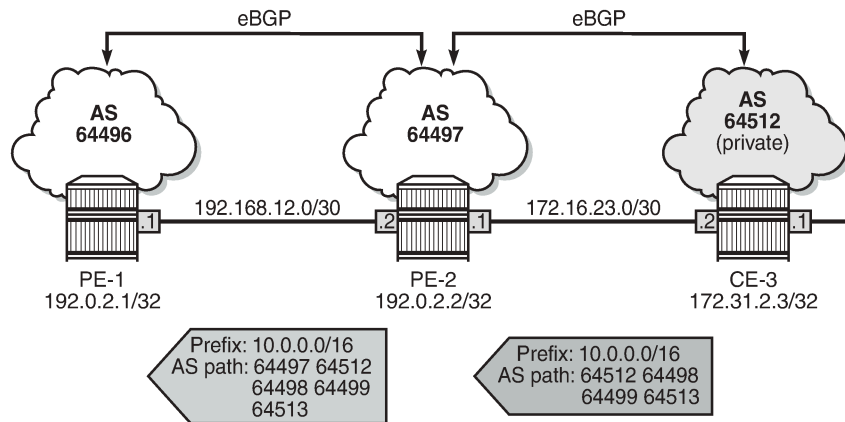
```

[/]
A:admin@PE-1# show router bgp routes 10.0.0.0/16
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  10.0.0.0/16                None       None
      192.168.12.2           None       0
      64497 64512 64498 64499 64513
-----
Routes : 1
=====

```

Figure 108: PE-2 adds its ASN and keeps all ASNs in the AS path (default) shows the BGP routes for prefix 10.0.0.0/16 received by PE-2 and PE-1:

Figure 108: PE-2 adds its ASN and keeps all ASNs in the AS path (default)



35891

In the following examples, different **remove-private** ASN configurations are demonstrated: first without **replace true** and afterward with **replace true**.

- **remove-private** ASN without any extra option (= default setting)
- **remove-private** ASN with **limited true** option
- **remove-private** ASN with **skip-peer-as true** option

Remove all private ASNs

On PE-2, the **remove-private** command is configured for neighbor 192.168.12.1, as follows:

```
# on PE-2:
configure {
  router "Base" {
    bgp {
      split-horizon true
      group "eBGP" {
        family {
          ipv4 true
        }
      }
    }
    neighbor "172.16.23.2" {
      group "eBGP"
      peer-as 64512
    }
    neighbor "192.168.12.1" {
      group "eBGP"
      peer-as 64496
      remove-private {
        skip-peer-as false
      }
    }
  }
}
```



```
}

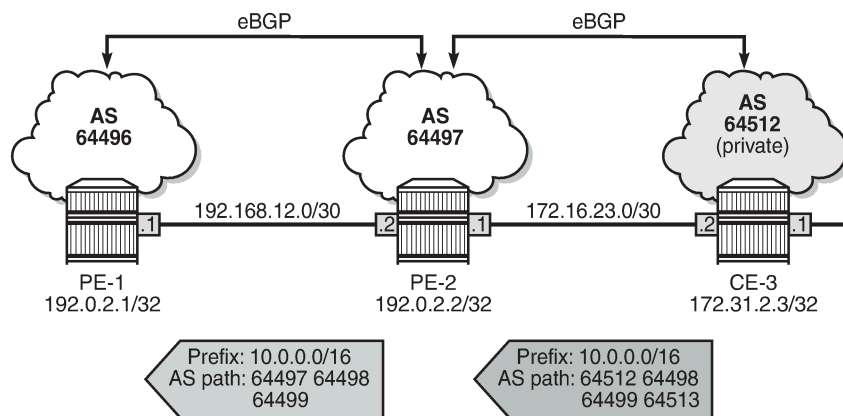
```

PE-2 removes all private ASNs (64512 from CE-3 and 64513 from CE-6) from the AS path, which makes the AS path shorter. PE-1 receives the following BGP route for prefix 10.0.0.0/16:

```
[/]
A:admin@PE-1# show router bgp routes 10.0.0.0/16
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
  Nexthop (Router)                         Path-Id   IGP Cost
  As-Path                                   Label
-----
u*>i 10.0.0.0/16                             None     None
      192.168.12.2                             None      0
      64497 64498 64499
-----
Routes : 1
=====
```

Figure 109: PE-2 adds its own ASN and removes all private ASNs shows the AS path of the BGP routes for prefix 10.0.0.0/16 received by PE-2 and PE-1:

Figure 109: PE-2 adds its own ASN and removes all private ASNs



35892

Replace all private ASNs

On PE-2, the **remove-private** command is configured with the **replace true** option for neighbor 192.168.12.1, as follows:

```
# on PE-2:
```

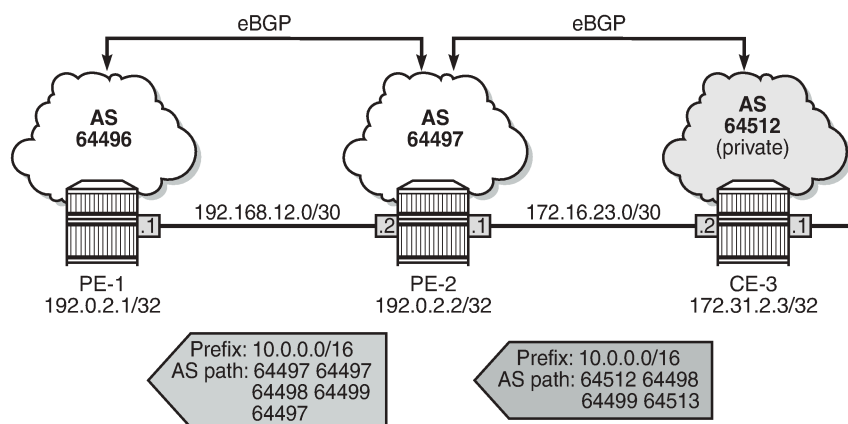
```
configure {
  router "Base" {
    bgp {
      split-horizon true
      group "eBGP" {
        family {
          ipv4 true
        }
      }
      neighbor "172.16.23.2" {
        group "eBGP"
        peer-as 64512
      }
      neighbor "192.168.12.1" {
        group "eBGP"
        peer-as 64496
        remove-private {
          skip-peer-as false
          replace true
        }
      }
    }
  }
}
```

PE-2 adds its ASN 64497 and replaces the private ASNs 64512 and 64513 with its own public ASN 64497 (in bold), so ASN 64497 occurs three times in the AS path, as follows:

```
[/]
A:admin@PE-1# show router bgp routes 10.0.0.0/16
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  10.0.0.0/16                None       None
      192.168.12.2           None       0
      64497 64497 64498 64499 64497
      -
-----
Routes : 1
=====
```

Figure 110: PE-2 adds its own ASN and replaces all private ASNs with its own ASN shows the BGP routes for prefix 10.0.0.0/16 received by PE-2 and PE-1.

Figure 110: PE-2 adds its own ASN and replaces all private ASNs with its own ASN

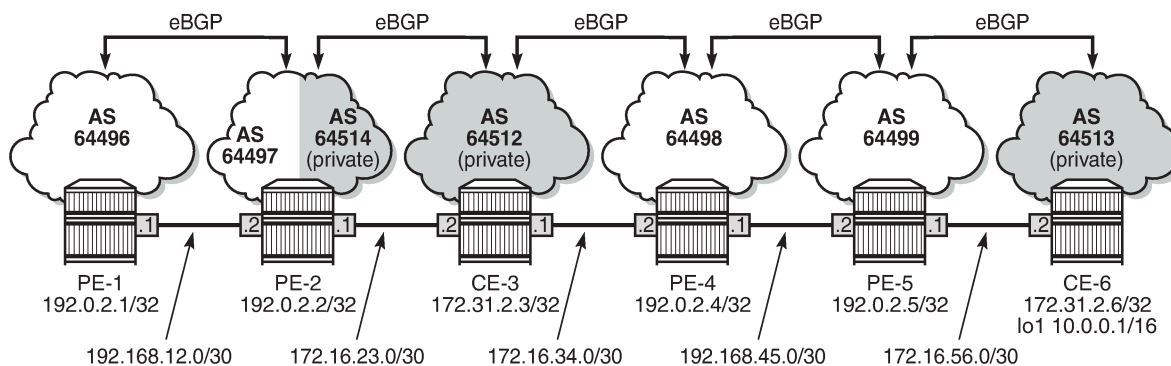


35893

Use case 2: Local private ASN in PE-2

Figure 111: Use case 2 topology shows the example topology that is modified with local private ASN 64514 configured on PE-2 for the neighbor 172.16.23.2. On CE-3, the peering with neighbor 172.16.23.1 is configured with private ASN 64514.

Figure 111: Use case 2 topology



35894

Initially (without **remove-private** command), the private ASN is kept. The BGP configuration on PE-2 is as follows:

```
# on PE-2:
configure {
  router "Base" {
    bgp {
      split-horizon true
      group "eBGP" {
        family {
          ipv4 true
        }
      }
    }
  }
}
```

```

neighbor "172.16.23.2" {
  group "eBGP"
  peer-as 64512
  local-as {
    as-number 64514
  }
}
neighbor "192.168.12.1" {
  group "eBGP"
  peer-as 64496
  delete remove-private
}
}

```

The BGP configuration on CE-3 is modified as follows:

```

# on CE-3:
configure {
  router "Base" {
    bgp {
      neighbor "172.16.23.1" {
        group "eBGP"
        peer-as 64514
      }
    }
  }
}

```

On PE-2, the received BGP route for prefix 10.0.0.0/16 is the same as before. With the preceding BGP configuration, PE-2 adds two ASNs: private ASN 64514 and public ASN 64497. PE-1 receives the following BGP route for prefix 10.0.0.0/16:

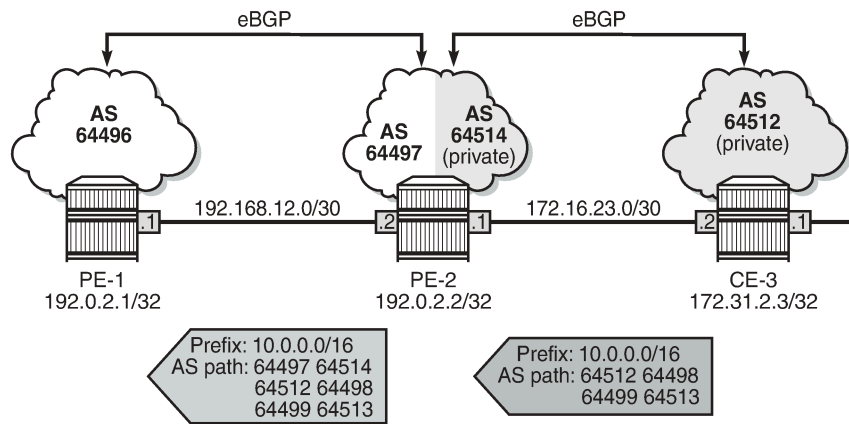
```

[/]
A:admin@PE-1# show router bgp routes 10.0.0.0/16
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)        Path-Id    IGP Cost
      As-Path                  Label
-----
u*>i  10.0.0.0/16                None       None
      192.168.12.2           None       0
      64497 64514 64512 64498 64499 64513
      -
-----
Routes : 1
=====

```

Figure 112: PE-2 adds its own private ASN and its public ASN (default) shows the AS path of the BGP routes received by PE-2 and PE-1.

Figure 112: PE-2 adds its own private ASN and its public ASN (default)



35895

When the local ASN is explicitly configured as private, the local ASN is not added to the AS path attribute. The local address configuration on PE-2 is modified with the **private** option, as follows:

```
# on PE-2:
configure {
  router "Base" {
    bgp {
      neighbor "172.16.23.2" {
        group "eBGP"
        peer-as 64512
        local-as {
          as-number 64514
          private true
        }
      }
    }
  }
}
```

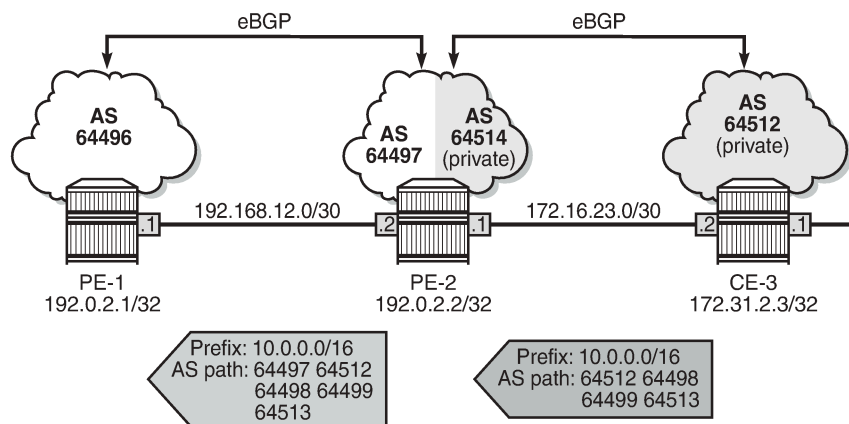
PE-1 receives the BGP route for prefix 10.0.0.0/16 with an AS path that does not include the private ASN 64514 anymore, as follows:

```
[/]
A:admin@PE-1# show router bgp routes 10.0.0.0/16
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  10.0.0.0/16                None       None
      192.168.12.2          None       0
      64497 64512 64498 64499 64513
-----
```

```
Routes : 1
=====
```

Figure 113: PE-2 adds only its own public ASN when local ASN is configured as private shows the AS paths in the BGP routes received by PE-2 and PE-1.

Figure 113: PE-2 adds only its own public ASN when local ASN is configured as private



35896

Remove private ASNs until the first public ASN

On PE-2, the **remove-private** command is configured with the **limited true** option, as follows:

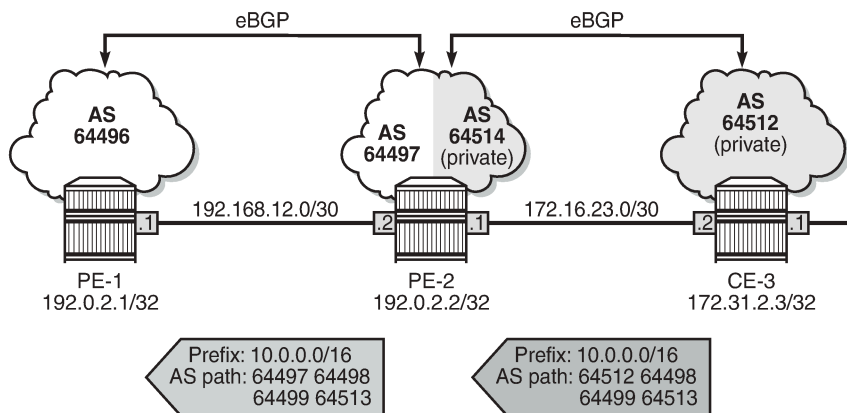
```
# on PE-2:
configure {
  router "Base" {
    bgp {
      split-horizon true
      group "eBGP" {
        family {
          ipv4 true
        }
      }
    }
    neighbor "172.16.23.2" {
      group "eBGP"
      peer-as 64512
      local-as {
        as-number 64514
        private true
      }
    }
    neighbor "192.168.12.1" {
      group "eBGP"
      peer-as 64496
      remove-private {
        limited true
        skip-peer-as false
      }
    }
  }
}
```

The first ASN in the AS path is private (64512) and is removed by PE-2. The next ASN in the AS path is public (64498), so the rest of the AS path is preserved. PE-1 receives the following BGP route for prefix 10.0.0.0/16:

```
[/]
A:admin@PE-1# show router bgp routes 10.0.0.0/16
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  10.0.0.0/16              None       None
      192.168.12.2          None       0
      64497 64498 64499 64513
-----
Routes : 1
=====
```

Figure 114: PE-2 removes the private ASNs until the first public ASN shows the BGP routes received by PE-2 and PE-1.

Figure 114: PE-2 removes the private ASNs until the first public ASN



35897

Replace private ASNs until the first public ASN

On PE-2, the **replace true** option is added to the **remove-private** settings:

```
# on PE-2:
configure {
  router "Base" {
```

```

    bgp {
      split-horizon true
      group "eBGP" {
        family {
          ipv4 true
        }
      }
      neighbor "172.16.23.2" {
        group "eBGP"
        peer-as 64512
        local-as {
          as-number 64514
          private true
        }
      }
      neighbor "192.168.12.1" {
        group "eBGP"
        peer-as 64496
        remove-private {
          limited true
          skip-peer-as false
          replace true
        }
      }
    }
  
```

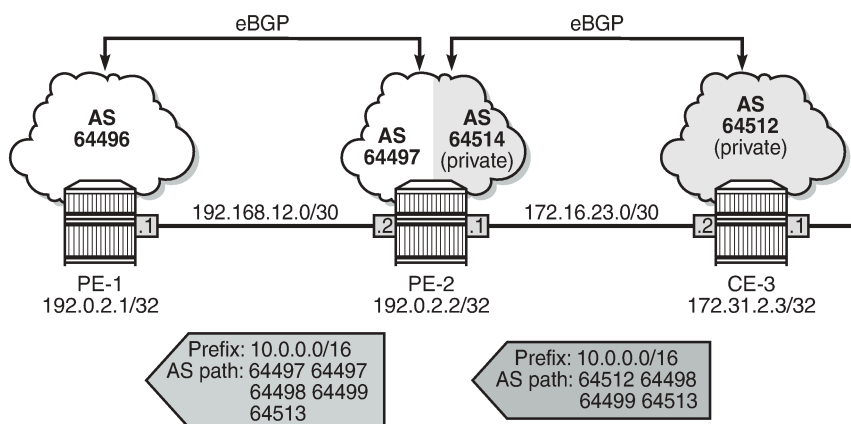
Instead of removing the private ASN 64512, PE-2 replaces it with its own public ASN 64497, so PE-1 receives the following BGP route for prefix 10.0.0.0/16:

```

[/]
A:admin@PE-1# show router bgp routes 10.0.0.0/16
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
u*>i  10.0.0.0/16                None       None
      192.168.12.2           None       0
      64497 64497 64498 64499 64513
-----
Routes : 1
=====
  
```

This route is shown in [Figure 115: PE-2 replaces the private ASNs until the first public ASN.](#)

Figure 115: PE-2 replaces the private ASNs until the first public ASN

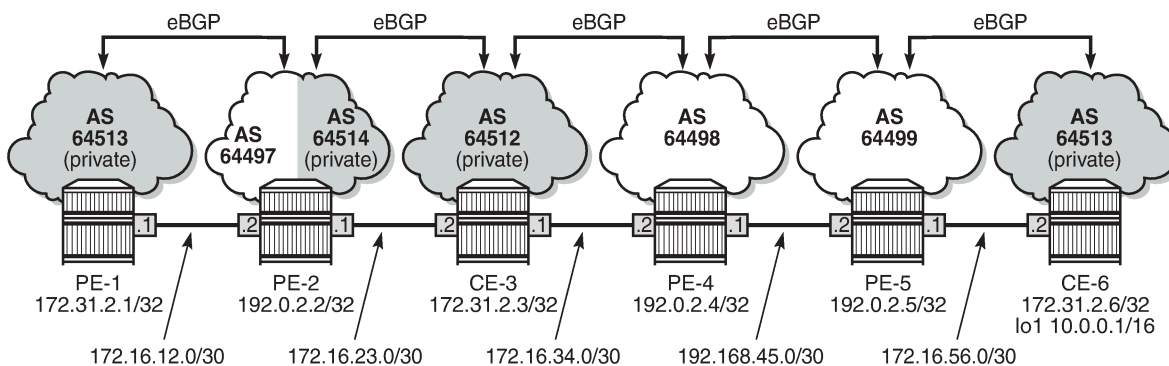


35898

Use case 3: CE-1 and CE-6 in the same private AS

Figure 116: Use case 3 topology with private ASN 64513 on CE-1 and CE-6 shows the Use case 3 topology where PE-1 is replaced by CE-1 with a private ASN 64513, equal to the private ASN of CE-6.

Figure 116: Use case 3 topology with private ASN 64513 on CE-1 and CE-6



35899

On PE-2, the peer ASN for neighbor 172.16.12.1 is 64513. Initially, no private ASNs are removed. The BGP configuration is as follows:

```
# on PE-2:
configure {
  router "Base" {
    bgp {
      split-horizon true
      group "eBGP" {
        family {
          ipv4 true
        }
      }
    }
    neighbor "172.16.23.2" {
```

```

group "eBGP"
peer-as 64512
local-as {
    as-number 64514
    private true
}
}
neighbor "172.16.12.1" {
    group "eBGP"
    peer-as 64513
}

```

On CE-1, the received route for prefix 10.0.0.0/16 is invalid, because CE-1 detects its own ASN in the AS path attribute, which is considered an AS loop:

```

[/]
A:admin@CE-1# show router bgp routes 10.0.0.0/16
=====
BGP Router ID:172.31.2.1      AS:64513      Local AS:64513
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
i     10.0.0.0/16            None       None
      172.16.12.2          None       0
      64497 64512 64498 64499 64513
      -
-----
Routes : 1
=====

```

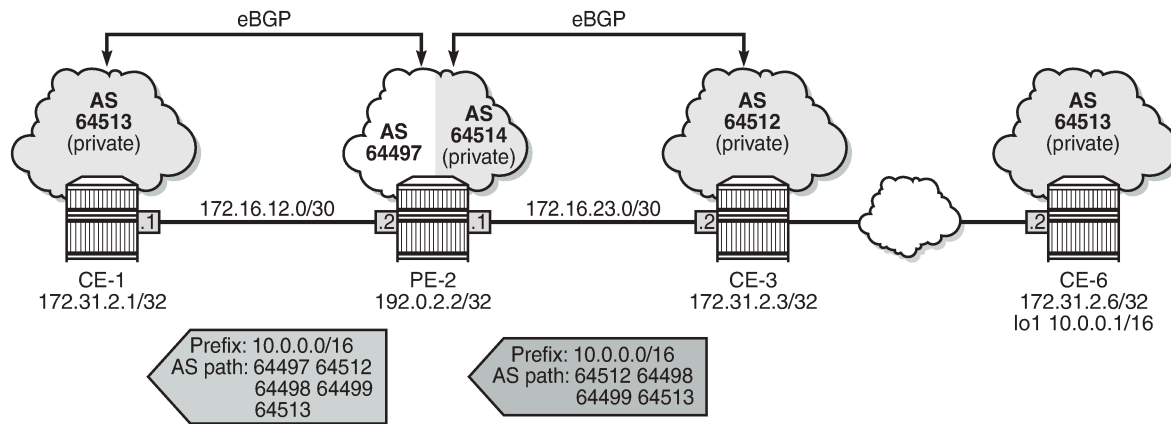
```

[/]
A:admin@CE-1# show router bgp routes 10.0.0.0/16 detail | match Flags
Flags      : Invalid IGP AS-Loop      # Original Attributes
Flags      : Invalid IGP AS-Loop      # Modified Attributes

```

Figure 117: PE-2 adds its public ASN to the AS path shows the BGP routes received by PE-2 and CE-1.

Figure 117: PE-2 adds its public ASN to the AS path



35900

Remove private ASNs except peer AS 64513

On PE-2, the **remove-private** command is configured with the **skip-peer-as true** option, as follows:

```
# on PE-2:
configure {
  router "Base" {
    bgp {
      split-horizon true
      group "eBGP" {
        family {
          ipv4 true
        }
      }
      neighbor "172.16.23.2" {
        group "eBGP"
        peer-as 64512
        local-as {
          as-number 64514
          private true
        }
      }
      neighbor "172.16.12.1" {
        group "eBGP"
        peer-as 64513
        remove-private {
          skip-peer-as true
        }
      }
    }
  }
}
```

On PE-2, for neighbor 172.16.12.1, the peer ASN is 64513, so this private ASN is not removed; only private ASN 64512 (from CE-3) is removed. As a result, CE-1 receives the following BGP route:

```
[/]
A:admin@CE-1# show router bgp routes 10.0.0.0/16
=====
BGP Router ID:172.31.2.1      AS:64513      Local AS:64513
```

```

=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
  Nexthop (Router)                         Path-Id   IGP Cost
  As-Path                                    Label
-----
i   10.0.0.0/16                             None      None
    172.16.12.2                             None      0
    64497 64498 64499 64513                  -
-----
Routes : 1
=====

```

Again, this route is invalid because of the AS loop, as indicated by the flags:

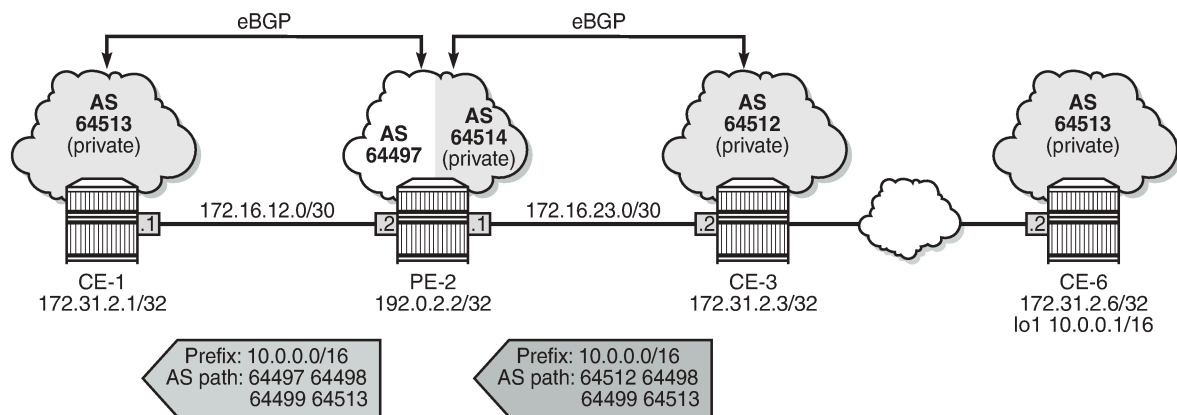
```

[/]
A:admin@CE-1# show router bgp routes 10.0.0.0/16 detail | match Flags
Flags          : Invalid IGP AS-Loop      # Original Attributes
Flags          : Invalid IGP AS-Loop      # Modified Attributes

```

Figure 118: PE-2 removes the private ASNs except peer ASN 64513 shows the BGP routes received by PE-2 and CE-1.

Figure 118: PE-2 removes the private ASNs except peer ASN 64513



35901

Replace private ASNs except peer AS 64513

On PE-2, the **remove-private** command is modified with the **replace true** option, as follows:

```

# on PE-2:
configure {
  router "Base" {
    bgp {

```

```

split-horizon true
group "eBGP" {
    family {
        ipv4 true
    }
}
neighbor "172.16.23.2" {
    group "eBGP"
    peer-as 64512
    local-as {
        as-number 64514
        private true
    }
}
neighbor "172.16.12.1" {
    group "eBGP"
    peer-as 64513
    remove-private {
        skip-peer-as true
        replace true
    }
}
}

```

The following BGP route for prefix 10.0.0.0/16 is received on CE-1. PE-2 has replaced the private ASN 64512 in the AS path with its own public ASN 64497, while the private ASN 64513 is preserved.

```

[/]
A:admin@CE-1# show router bgp routes 10.0.0.0/16
=====
BGP Router ID:172.31.2.1      AS:64513      Local AS:64513
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
i     10.0.0.0/16                           None       None
      172.16.12.2                           None       0
      64497 64497 64498 64499 64513         -
-----
Routes : 1
=====

```

Again, the route is invalid because of the AS loop, as indicated by the flags:

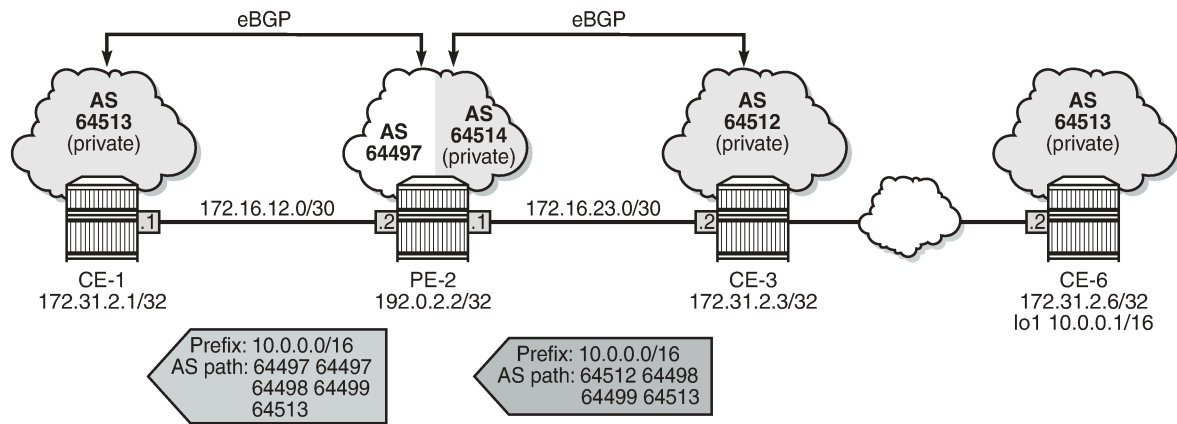
```

[/]
A:admin@CE-1# show router bgp routes 10.0.0.0/16 detail | match Flags
Flags      : Invalid IGP AS-Loop      # Original Attributes
Flags      : Invalid IGP AS-Loop      # Modified Attributes

```

Figure 119: PE-2 replaces the private ASNs except peer ASN 64513 shows the received BGP routes on PE-2 and CE-1.

Figure 119: PE-2 replaces the private ASNs except peer ASN 64513



35902

Loop-detect threshold N

If the received AS path has a local AS number of the router, the route is considered a loop if the number of occurrences is greater than the configured value N. By default, the loop-detect threshold in BGP is zero, meaning that any route with at least one occurrence of the local ASN is considered a loop and therefore invalid. The loop-detect threshold can be configured in the general **bgp** context, the **bgp group** context, or the **bgp neighbor** context.

On CE-1 and CE-6, the loop-detect threshold is configured with the value of 1 for group "eBGP", as follows:

```
# on CE-1 and CE-6:
configure {
  router "Base" {
    bgp {
      group "eBGP" {
        loop-detect-threshold 1
      }
    }
  }
}
```



Note:

Loop-detect thresholds are only applicable for newly learned prefixes. Existing loop states remain unchanged.

After the BGP session with peer PE-2 has been bounced (disabled and re-enabled), the prefix is learned again. The route is valid, because the local ASN only occurs once in the AS path attribute, so the loop-detect threshold is not violated on CE-1.

```
# Bounce BGP group "eBGP" on CE-1 and CE-6:
configure {
  router "Base" {
    bgp {
      group "eBGP" {
        admin-state disable
        commit
        admin-state enable
        commit
      }
    }
  }
}
```

```

}

[/]
A:admin@CE-1# show router bgp routes 10.0.0.0/16
=====
BGP Router ID:172.31.2.1      AS:64513      Local AS:64513
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                  l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  10.0.0.0/16              None       None
      172.16.12.2           None       0
      64497 64497 64498 64499 64513
-----
Routes : 1
=====

```



Note:

The loop-detect threshold is not reflected in the **show** commands.

Conclusion

Network operators may assign private ASNs to the BGP speakers in a region or domain. These private ASNs may not be unique when advertised into a public domain. In such cases, the **remove-private** command can either remove one or more private ASNs or replace the private ASNs with its public ASN.

BGP Route Leaking

This chapter provides information about BGP route leaking.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written based on SR OS Release 14.0.R4. The MD-CLI in the current edition corresponds to SR OS Release 22.2.R2.

Overview

Route leaking refers to the process of copying a route from one router context to another.

Network administrators may need to leak routes between routing instances in the same SR OS router. BGP route leaking is an alternative to using import and export policies based on communities to exchange routes between Virtual Router and Forwarders (VRFs).

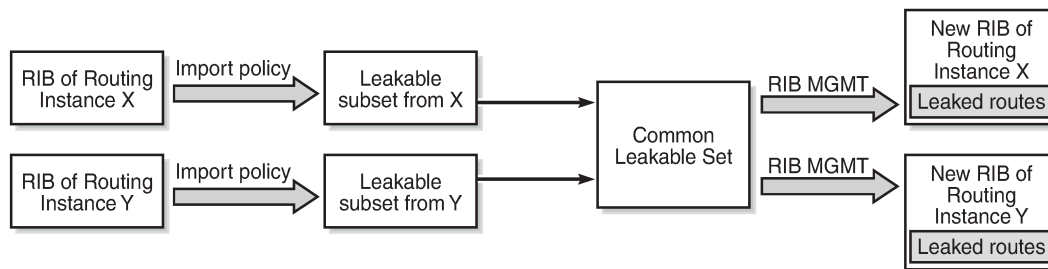
It is possible to leak a copy of a BGP route (including all its path attributes) from one routing instance to another in the same SR OS router. This BGP route leaking capability applies to IPv4, IPv6, and label-IPv4 routes. Leaking is supported from the GRT to a VPRN, from one VPRN to another VPRN, and from a VPRN to the GRT.

Any BGP route for an IPv4 or IPv6 prefix can be leaked. A BGP route does not have to be the best path or used for forwarding in the source instance in order to be leaked. In SR OS releases earlier than 19.10.R1, the BGP route had to be valid (that is, the next-hop must be resolved; the AS PATH must not exhibit a loop, for example). In SR OS Release 19.10.R1, and later, BGP in the base router can be configured to allow unresolved route leaking, as described in the *Unresolved Route Leaking from Base Router to VPRN* chapter in the "Unicast Routing Protocols" volume of *7450 ESS, 7750 SR, and 7950 XRS Advanced Configuration Guide — Part I*.

An IPv4 or IPv6 BGP route becomes a candidate for leaking to another instance when it is specially marked by a BGP import policy. This marking is achieved by accepting the route with a **bgp-leak** action in the route policy. Routes that are candidates for leaking to other instances show a **leakable** flag in the output of various **show router bgp** commands.

To copy a leakable BGP route from a source instance into the BGP RIB of a target instance, the target instance must be configured with a leak-import policy that matches and accepts the leakable route. There are separate leak-import policies for IPv4 and IPv6 routes. Up to 15 leak-import policies can be chained together for more complex examples. In the target instance, the **show router bgp routes** command displays leaked BGP RIB-IN routes in addition to direct RIB-IN routes learned from neighbors of the routing instance. A **leaked** flag is added to the leaked RIB-IN entries. [Figure 120: BGP route leaking process](#) shows the process of BGP route leaking.

Figure 120: BGP route leaking process



25963

Leaked BGP routes can be advertised to BGP neighbors (peers) of the target routing instance. The BGP next hop of a leaked route is automatically reset to self whenever it is advertised to a peer of the target instance. Normal route advertisement rules apply: by default, the leaked route is advertised if it is the overall best path that is used as the active route to the destination and it is not blocked by the IBGP-to-IBGP split-horizon rule.

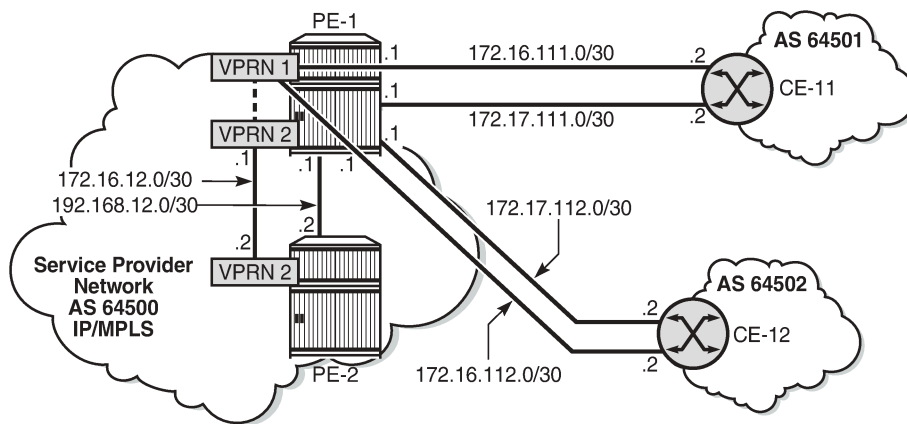
A BGP route leaked into a VPRN can be exported from the VPRN as a VPN-IPv4/v6 route if it matches the VRF export policy. Normal VPN export rules apply: by default, the leaked route is exported if it is the overall best path and it is used as the active route to the destination.

This chapter describes BGP route leaking only. For other routes, such as IS-IS, OSPF, RIP, and static routes, VPRN route leaking mechanisms apply that are protocol independent, see chapter *Traffic Leaking from VPRN to GRT* in the "Unicast Routing Protocols" volume of *7450 ESS, 7750 SR, and 7950 XRS Advanced Configuration Guide — Part I*.

Configuration

Figure 121: [Example topology](#) shows the example topology used in this chapter, including the IPv4 addresses. For each of the examples, a dedicated figure will show the specific topology, which is a subset of the topology in [Figure 121: Example topology](#). The interfaces also have IPv6 addresses, which will be shown in [Figure 125: BGP IPv6 route leaking between VPRNs](#) and [Figure 126: BGP IPv6 route leaking from GRT and VPRN to VPRN](#). VPRN 2 also has CEs attached, but for simplicity, these are not shown on the figures and no CLI will be shown for any CE.

Figure 121: Example topology



25964

The following examples will be explained:

- [Example 1 - BGP IPv4 route leaking between VPRNs. Global BGP policy](#)
- [Example 2 - BGP IPv4 route leaking between VPRNs per neighbor](#)
- [Example 3 - BGP IPv4 route leaking from VPRN to GRT per BGP group](#)
- [Example 4 - BGP IPv4 route leaking from GRT to VPRN per neighbor](#)
- [Example 5 - BGP IPv6 route leaking between VPRNs. Global VPRN BGP configuration](#)
- [Example 6 - BGP IPv6 route leaking from GRT to VPRN and from VPRN to VPRN](#)

Initial configuration

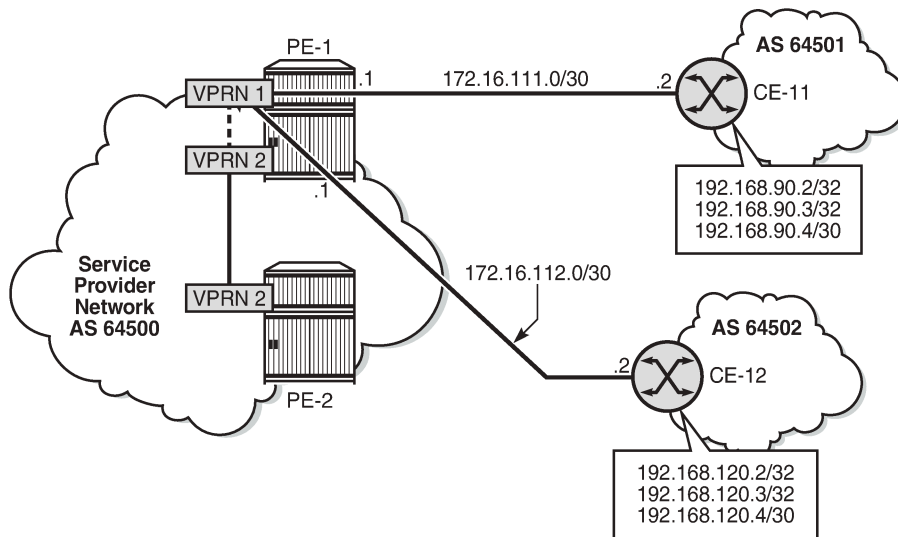
The nodes in the example topology have the following initial configuration:

- Cards, MDAs, ports
- Router interfaces
- IGP (IS-IS or OSPF) between the PEs
- LDP between the PEs
- VPRN "VPRN 1" on PE-1; VPRN "VPRN 2" on PE-1 and PE-2
- BGP (IBGP between the PEs; EBGP between PE-1 and the CEs)
 - On the PEs, BGP is configured in the base router and in the VPRNs.
- Loopback addresses and black-hole static routes in the CEs. Different routes are exported to GRT and VPRN 1 on PE-1

Example 1 - BGP IPv4 route leaking between VPRNs. Global BGP policy

[Figure 122: BGP IPv4 route leaking between VPRNs](#) shows the topology for this example. CE-11 exports routes such as 192.168.90.2/32 to VPRN 1 on PE-1, and CE-12 exports routes such as 192.168.120.2/32 to VPRN 1 on PE-1.

Figure 122: BGP IPv4 route leaking between VPRNs



25965

In MD-CLI, all EBGP routes are by default rejected unless policies are configured. The following import policy accepts all BGP routes and is applied in VPRN 1.

```
# on PE-1:
configure {
  policy-options {
    policy-statement "import-bgp" {
      entry 10 {
        from {
          protocol {
            name [bgp]
          }
        }
        action {
          action-type accept
        }
      }
    }
  }
  service {
    vprn "VPRN 1" {
      bgp {
        import {
          policy ["import-bgp"]
        }
      }
    }
  }
}
```

The routing table for VPRN 1 on PE-1 includes routes that are learned from CE-11 and CE-12, as follows:

```
[/]
A:admin@PE-1# show router 1 route-table

=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]                               Type      Proto    Age      Pref
=====
```

Next Hop[Interface Name]			Metric	
172.16.1.1/32	Local	Local	00h04m46s	0
system			0	
172.16.111.0/30	Local	Local	00h04m46s	0
int-PE-1-CE-11			0	
172.16.112.0/30	Local	Local	00h04m46s	0
int-PE-1-CE-12			0	
192.168.90.2/32	Remote	BGP	00h00m21s	170
172.16.111.2			0	
192.168.90.3/32	Remote	BGP	00h00m21s	170
172.16.111.2			0	
192.168.90.4/30	Remote	BGP	00h00m21s	170
172.16.111.2			0	
192.168.120.2/32	Remote	BGP	00h00m36s	170
172.16.112.2			0	
192.168.120.3/32	Remote	BGP	00h00m36s	170
172.16.112.2			0	
192.168.120.4/32	Remote	BGP	00h00m36s	170
172.16.112.2			0	

 No. of Routes: 9
 Flags: n = Number of times nexthop is repeated
 B = BGP backup route available
 L = LFA nexthop available
 S = Sticky ECMP requested
 =====

These BGP routes are not leakable, by default, as follows:

```
[/]
A:admin@PE-1# show router 1 bgp routes ipv4 leakable
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)        Path-Id    IGP Cost
      As-Path                  Label
-----
No Matching Entries Found.
=====
```

The routing table for VPRN 2 does not include any of these routes because BGP route leaking is disabled by default:

```
[/]
A:admin@PE-1# show router 2 route-table
=====
Route Table (Service: 2)
=====
Dest Prefix[Flags]          Type  Proto  Age  Pref
  Next Hop[Interface Name]                Metric
-----
```

```

172.16.2.1/32          Local   Local   00h04m46s  0
  system
172.16.2.2/32          Remote  BGP VPN 00h03m52s 170
  192.0.2.2 (tunneled)
172.16.12.0/30         Local   Local   00h04m46s  0
  int-PE-1-PE-2_VPN2
-----
No. of Routes: 3
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

To configure BGP route leaking, an import policy with **action>bgp-leak true** is required in VPRN 1. The BGP route leaking policy is configured on PE-1, as follows:

```

# on PE-1:
configure {
  policy-options {
    policy-statement "BGP-Leak" {
      entry 10 {
        from {
          protocol {
            name [bgp]
          }
        }
        action {
          action-type accept
          bgp-leak true
        }
      }
    }
  }
}

```

By adding **action>action-type accept** and **action>bgp-leak true**, BGP routes are imported and marked as BGP-leakable, meaning they are available to be copied—with their complete set of BGP path attributes—to the BGP RIB-IN of another routing instance.

In this example, the BGP route leaking policy replaces the import policy applied in VPRN 1 in the general BGP configuration, but the route leaking policy can also be configured in the group context, or per neighbor:

```

# on PE-1:
configure {
  service {
    vprn "VPRN 1" {
      bgp {
        delete import
        import {
          policy ["BGP-Leak"]
        }
      }
    }
  }
}

```

With the preceding configuration, SR OS is marking all the BGP routes imported into the VPRN as leakable. The BGP routes originate from CE-11 or CE-12 in this example.

The following command shows which BGP routes in VPRN 1 are marked as leakable:

```

[/]
A:admin@PE-1# show router 1 bgp routes ipv4 leakable
=====

```

```

BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                       Path-Id    IGP Cost
      As-Path                                Label
-----
u*>i  192.168.90.2/32                          None       None
      172.16.111.2                             None       0
      64501                                     -
u*>i  192.168.90.3/32                          None       None
      172.16.111.2                             None       0
      64501                                     -
u*>i  192.168.90.4/30                          None       None
      172.16.111.2                             None       0
      64501                                     -
u*>i  192.168.120.2/32                         None       None
      172.16.112.2                             None       0
      64502                                     -
u*>i  192.168.120.3/32                         None       None
      172.16.112.2                             None       0
      64502                                     -
u*>i  192.168.120.4/32                         None       None
      172.16.112.2                             None       0
      64502                                     -
-----
Routes : 6
=====

```

The routes learned from CE-11 and CE-12 are leakable. The following detailed output for one of the routes in the preceding list shows the flag "Leakable". The route source is external because the routes are imported (from CE-11 or CE-12):

```

[/]
A:admin@PE-1# show router 1 bgp routes 192.168.90.2/32 detail
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Original Attributes

Network       : 192.168.90.2/32
Nexthop       : 172.16.111.2
Path Id       : None
From          : 172.16.111.2
Res. Protocol : LOCAL                Res. Metric   : 0
Res. Nexthop  : 172.16.111.2
Local Pref.   : n/a                  Interface Name : int-PE-1-CE-11
---snip---

```

```

Originator Id   : None                Peer Router Id : 172.16.0.11
Fwd Class       : None                Priority       : None
Flags         : Used Valid Best IGP Leakable In-RTM
Route Source : External
AS-Path         : 64501
---snip---
    
```

BGP leakable routes can be imported into another VPRN. Prefix lists can be used to filter specific routes for BGP leaking, but that is not configured in this example. The following import policy is configured on PE-1 to import BGP leakable routes:

```

# on PE-1:
configure {
  policy-options {
    policy-statement "Import-Leakable-Routes" {
      entry 10 {
        from {
          protocol {
            name [bgp]
          }
        }
        action {
          action-type accept
        }
      }
    }
  }
}
    
```

In each of the examples, the same import policy will be used. The import policy to import BGP leakable routes is applied in the VPRN 2 on PE-1 as follows:

```

# on PE-1:
configure {
  service {
    vprn "VPRN 2" {
      bgp {
        rib-management {
          ipv4 {
            leak-import {
              policy ["Import-Leakable-Routes"]
            }
          }
        }
      }
    }
  }
}
    
```

The following command shows that VPRN 2 imported leaked BGP routes from VPRN 1. The status code "l" indicates that the route is leaked.

```

[/]
A:admin@PE-1# show router 2 bgp routes ipv4 leaked
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
    
```

Flag	Network Nexthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
u*>li	192.168.90.2/32 172.16.111.2 (VPRN 1) 64501	100 None	None 0 -
u*>li	192.168.90.3/32 172.16.111.2 (VPRN 1) 64501	100 None	None 0 -
u*>li	192.168.90.4/30 172.16.111.2 (VPRN 1) 64501	100 None	None 0 -
u*>li	192.168.120.2/32 172.16.112.2 (VPRN 1) 64502	100 None	None 0 -
u*>li	192.168.120.3/32 172.16.112.2 (VPRN 1) 64502	100 None	None 0 -
u*>li	192.168.120.4/32 172.16.112.2 (VPRN 1) 64502	100 None	None 0 -

Routes : 6
=====

The flags in the detailed output for a particular leaked BGP route from the preceding list include the flag "Leaked". The route source for this leaked route is VPRN 1 and all BGP attributes are preserved, as follows:

```
[/]
A:admin@PE-1# show router 2 bgp routes 192.168.90.2/32 detail
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Original Attributes

Network       : 192.168.90.2/32
Nexthop      : 172.16.111.2 (VPRN 1)
Path Id      : None
From         : BGP VPRN 1
Res. Protocol : LOCAL                      Res. Metric   : 0
Res. Nexthop : 172.16.111.2
Local Pref.  : 100                        Interface Name : int-PE-1-CE-11
Aggregator AS : None                      Aggregator    : None
Atomic Aggr. : Not Atomic                  MED           : None
AIGP Metric  : None                      IGP Cost      : 0
Connector    : None
Community    : No Community Members
Cluster      : No Cluster Members
Originator Id : None                      Peer Router Id : 0.0.0.0
Fwd Class    : None                      Priority       : None
Flags       : Used Valid Best IGP Leaked In-RTM
Route Source : Leaked from VPRN 1
AS-Path      : 64501
```



```
Route Tag      : 0
Neighbor-AS   : 64501
Orig Validation: NotFound
Source Class  : 0
Add Paths Send: Default
RIB Priority   : Normal
Last Modified : 00h00m36s
---snip---
```

The route table for VPRN 2 in the neighbor PE-2 contains the leaked routes, as follows:

```
[/]
A:admin@PE-2# show router 2 route-table

=====
Route Table (Service: 2)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]      Type   Proto   Age           Pref
                               Metric
-----
172.16.2.1/32                 Remote BGP VPN 00h12m14s    170
                               0
192.0.2.1 (tunneled)
172.16.2.2/32                 Local  Local   00h12m59s    0
                               0
system
172.16.12.0/30                Local  Local   00h12m59s    0
                               0
int-PE-2-PE-1_VPN2
192.168.90.2/32              Remote BGP     00h03m22s    170
                               0
172.16.12.1
192.168.90.3/32              Remote BGP     00h03m22s    170
                               0
172.16.12.1
192.168.90.4/30              Remote BGP     00h03m22s    170
                               0
172.16.12.1
192.168.120.2/32             Remote BGP     00h03m22s    170
                               0
172.16.12.1
192.168.120.3/32             Remote BGP     00h03m22s    170
                               0
172.16.12.1
192.168.120.4/32             Remote BGP     00h03m22s    170
                               0
172.16.12.1
-----
No. of Routes: 9
```

Example 2 - BGP IPv4 route leaking between VPRNs per neighbor

The topology used for this example is the same as for Example 1; see [Figure 122: BGP IPv4 route leaking between VPRNs](#). Both CEs export the same routes as in the preceding example, and the BGP route leaking policy is identical:

```
# on PE-1:
configure {
  policy-options {
    policy-statement "BGP-Leak" {
      entry 10 {
        from {
          protocol {
            name [bgp]
          }
        }
        action {
          action-type accept
          bgp-leak true
        }
      }
    }
  }
}
```

```

    }
  }
}

```

In the preceding example, the BGP route leaking policy was applied in the global **bgp** context in VPRN 1 and consequently, it applied to routes from all neighbors. In this example, the BGP route leaking policy is applied in VPRN 1 for neighbor CE-11 only, as follows:

```

# on PE-1:
configure {
  service {
    vprn "VPRN 1" {
      bgp {
        delete import
        neighbor "172.16.111.2" {
          import {
            policy ["BGP-Leak"]
          }
        }
      }
    }
  }
}

```

This import policy implies that only routes learned from CE-11 will be leakable. The following command shows all the BGP routes learned in VPRN 1 on PE-1.

```

[/]
A:admin@PE-1# show router 1 bgp routes
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               -          Label
-----
u*>i  192.168.90.2/32                          None       None
      172.16.111.2                          None       0
      64501                                   -          -
u*>i  192.168.90.3/32                          None       None
      172.16.111.2                          None       0
      64501                                   -          -
u*>i  192.168.90.4/30                          None       None
      172.16.111.2                          None       0
      64501                                   -          -
i     192.168.120.2/32                        None       None
      172.16.112.2                          None       0
      64502                                   -          -
i     192.168.120.3/32                        None       None
      172.16.112.2                          None       0
      64502                                   -          -
i     192.168.120.4/32                        None       None
      172.16.112.2                          None       0
      64502                                   -          -
-----
Routes : 6

```

Only the routes imported from CE-11 are accepted and leakable. The following command shows which IPv4 BGP routes are marked as leakable in VPRN 1 on PE-1:

```
[/]
A:admin@PE-1# show router 1 bgp routes ipv4 leakable
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                       Path-Id    IGP Cost
      As-Path                                Label
-----
u*>i  192.168.90.2/32                          None       None
      172.16.111.2                             None       0
      64501                                     -
u*>i  192.168.90.3/32                          None       None
      172.16.111.2                             None       0
      64501                                     -
u*>i  192.168.90.4/30                          None       None
      172.16.111.2                             None       0
      64501                                     -
-----
Routes : 3
=====
```

The BGP leakable routes can be imported into another VPRN instance. The import policy is the same as for Example 1:

```
# on PE-1:
configure {
  policy-options {
    policy-statement "Import-Leakable-Routes" {
      entry 10 {
        from {
          protocol {
            name [bgp]
          }
        }
        action {
          action-type accept
        }
      }
    }
  }
}
```

This import policy is applied in VPRN 2 in the same way as in example 1:

```
# on PE-1:
configure {
  service {
    vprn "VPRN 2" {
      bgp {
```

```

        rib-management {
            ipv4 {
                leak-import {
                    policy ["Import-Leakable-Routes"]
                }
            }
        }
    }
}

```

The following command shows the leaked routes in VPRN 2. Each of these routes is leaked from VPRN 1, as indicated between brackets in the following output. Only routes learned from CE-11 in VPRN 1 are leaked to VPRN 2.

```

[/]
A:admin@PE-1# show router 2 bgp routes ipv4 leaked
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                       Path-Id    IGP Cost
      As-Path                                Label
-----
u*>li 192.168.90.2/32                          100        None
      172.16.111.2 (VPRN 1)                    None        0
      64501
u*>li 192.168.90.3/32                          100        None
      172.16.111.2 (VPRN 1)                    None        0
      64501
u*>li 192.168.90.4/30                          100        None
      172.16.111.2 (VPRN 1)                    None        0
      64501
-----
Routes : 3
=====

```

The detailed output for any of these BGP routes shows that the flag "Leaked" is set and that the route source corresponds to VPRN 1, as follows for route 192.168.90.2/32:

```

[/]
A:admin@PE-1# show router 2 bgp routes 192.168.90.2/32 detail
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Original Attributes
Network       : 192.168.90.2/32

```

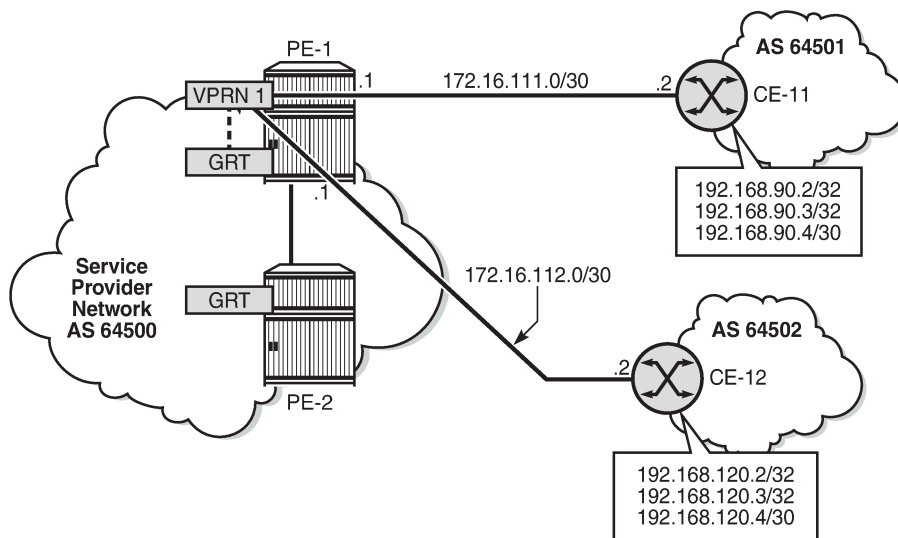
```

Nexthop      : 172.16.111.2 (VPRN 1)
Path Id      : None
From         : BGP VPRN 1
Res. Protocol : LOCAL                      Res. Metric   : 0
Res. Nexthop  : 172.16.111.2
Local Pref.   : 100                        Interface Name : int-PE-1-CE-11
Aggregator AS : None                      Aggregator    : None
Atomic Aggr.  : Not Atomic                 MED           : None
AIGP Metric   : None                      IGP Cost      : 0
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None                      Peer Router Id : 0.0.0.0
Fwd Class     : None                      Priority       : None
Flags       : Used Valid Best IGP Leaked In-RTM
Route Source : Leaked from VPRN 1
AS-Path       : 64501
---snip---
    
```

Example 3 - BGP IPv4 route leaking from VPRN to GRT per BGP group

Figure 123: BGP IPv4 route leaking from VPRN to GRT shows the topology for this example. CE-11 and CE-12 export the same routes to VPRN 1. The routes originating from CE-11 will be accepted, marked as leakable, and leaked to the GRT.

Figure 123: BGP IPv4 route leaking from VPRN to GRT



25966

The import policy is the same as in the preceding examples:

```

# on PE-1:
configure {
  policy-options {
    policy-statement "BGP-Leak" {
      entry 10 {
        from {
          protocol {
    
```

```

        name [bgp]
      }
    }
    action {
      action-type accept
      bgp-leak true
    }
  }
}

```

This policy is applied for BGP group "EBGP_64500to64501_IPv4", so the routes from CE-11 will be accepted and marked as leakable:

```

# on PE-1:
configure {
  service {
    vprn "VPRN 1" {
      bgp {
        group "EBGP_64500to64501_IPv4" {
          import {
            policy ["BGP-Leak"]
          }
        }
      }
    }
  }
}

```

The routing table for VPRN 1 in PE-1 contains the BGP routes exported by CE-11, as follows:

```

[/]
A:admin@PE-1# show router 1 route-table

=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]          Type   Proto   Age      Pref
Metric
-----
172.16.1.1/32
  system                          Local  Local   00h14m37s  0
  0
172.16.111.0/30
  int-PE-1-CE-11                  Local  Local   00h14m37s  0
  0
172.16.112.0/30
  int-PE-1-CE-12                  Local  Local   00h14m37s  0
  0
192.168.90.2/32
  172.16.111.2                    Remote BGP     00h01m45s  170
  0
192.168.90.3/32
  172.16.111.2                    Remote BGP     00h01m45s  170
  0
192.168.90.4/30
  172.16.111.2                    Remote BGP     00h01m45s  170
  0
-----
No. of Routes: 6
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====

```

The routing table of the base router does not include any of the BGP routes exported by the CEs, as follows:

```

[/]
A:admin@PE-1# show router route-table

```

```

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                                Type  Proto  Age      Pref
  Next Hop[Interface Name]                        Metric
-----
172.17.111.0/30                                   Local Local  00h14m37s 0
      int-PE-1-CE-11                               0
172.17.112.0/30                                   Local Local  00h14m37s 0
      int-PE-1-CE-12                               0
192.0.2.1/32                                       Local Local  00h14m37s 0
      system                                         0
192.0.2.2/32                                       Remote ISIS  00h14m13s 15
      192.168.12.2                                  10
192.168.12.0/30                                   Local Local  00h14m37s 0
      int-PE-1-PE-2                                 0
-----
No. of Routes: 5
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

The following command shows the leakable BGP routes in VPRN 1:

```

[/]
A:admin@PE-1# show router 1 bgp routes ipv4 leakable
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                   Path-Id    IGP Cost
      As-Path                             Label
-----
u*>i  192.168.90.2/32                         None       None
      172.16.111.2                         None       0
      64501                                 -
u*>i  192.168.90.3/32                         None       None
      172.16.111.2                         None       0
      64501                                 -
u*>i  192.168.90.4/30                         None       None
      172.16.111.2                         None       0
      64501                                 -
-----
Routes : 3
=====

```

The leakable BGP routes in VPRN 1 can be imported into the GRT. The import policy is identical to the import policy in the preceding examples, as follows:

```

# on PE-1:
configure {

```

```

policy-options {
  policy-statement "Import-Leakable-Routes" {
    entry 10 {
      from {
        protocol {
          name [bgp]
        }
      }
      action {
        action-type accept
      }
    }
  }
}

```

This import policy is applied in the base router, as follows:

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      rib-management {
        ipv4 {
          leak-import {
            policy ["Import-Leakable-Routes"]
          }
        }
      }
    }
  }
}

```

As a result, the leakable BGP routes in VPRN 1 are leaked to the GRT, as follows:

```

[/]
A:admin@PE-1# show router bgp routes ipv4 leaked
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Path-Id    Label
-----
u*>li 192.168.90.2/32                        100        None
      172.16.111.2 (VPRN 1)                 None       0
      64501                                   -
u*>li 192.168.90.3/32                        100        None
      172.16.111.2 (VPRN 1)                 None       0
      64501                                   -
u*>li 192.168.90.4/30                        100        None
      172.16.111.2 (VPRN 1)                 None       0
      64501                                   -
-----
Routes : 3
=====

```


The detailed information for any of these leaked routes shows that the flag "Leaked" is present and that the route source is VPRN 1, as follows:

```
[/]
A:admin@PE-1# show router bgp routes 192.168.90.2/32 detail
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Original Attributes

Network       : 192.168.90.2/32
Nexthop      : 172.16.111.2 (VPRN 1)
Path Id      : None
From         : BGP VPRN 1
Res. Protocol : LOCAL                Res. Metric   : 0
Res. Nexthop : 172.16.111.2
Local Pref.  : 100
Aggregator AS : None                Interface Name : int-PE-1-CE-11
Atomic Aggr. : Not Atomic          Aggregator    : None
AIGP Metric  : None                MED           : None
Connector    : None                IGP Cost      : 0
Community    : No Community Members
Cluster      : No Cluster Members
Originator Id : None                Peer Router Id : 0.0.0.0
Fwd Class    : None                Priority       : None
Flags      : Used Valid Best IGP Leaked In-RTM
Route Source : Leaked from VPRN 1
AS-Path      : 64501
---snip---
```

The GRT includes the leaked routes, as follows:

```
[/]
A:admin@PE-1# show router route-table
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
Next Hop[Interface Name]          Metric
-----
172.17.111.0/30                    Local Local  00h16m10s  0
int-PE-1-CE-11                    0
172.17.112.0/30                    Local Local  00h16m10s  0
int-PE-1-CE-12                    0
192.0.2.1/32                       Local Local  00h16m10s  0
system                             0
192.0.2.2/32                       Remote ISIS  00h15m47s  15
192.168.12.2                       10
192.168.12.0/30                    Local Local  00h16m10s  0
int-PE-1-PE-2                      0
192.168.90.2/32                    Remote BGP   00h00m36s  170
172.16.111.2                       0
192.168.90.3/32                    Remote BGP   00h00m36s  170
172.16.111.2                       0
```

```

192.168.90.4/30                               Remote BGP      00h00m36s 170
172.16.111.2                                 0
-----
No. of Routes: 8
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

The GRT on neighbor PE-2 also includes the leaked routes, as follows:

```

[/]
A:admin@PE-2# show router route-table

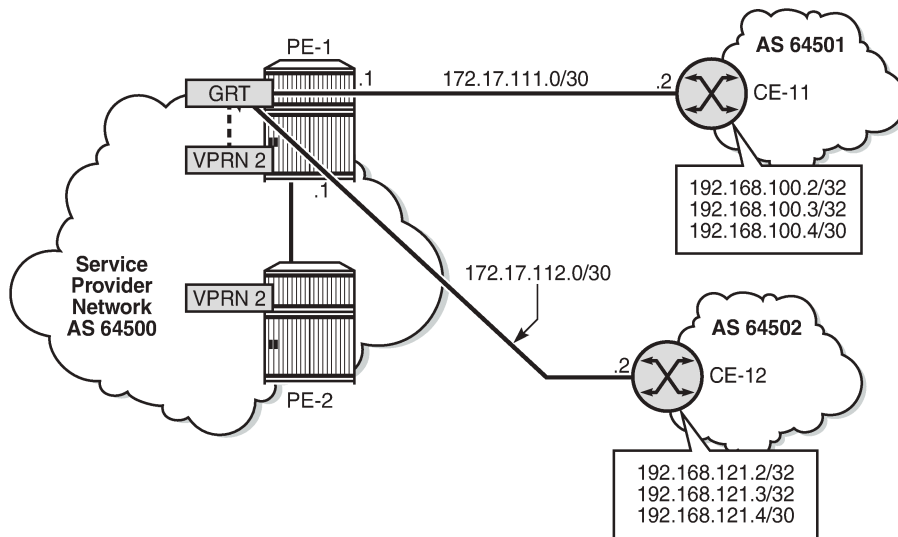
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                            Type  Proto  Age           Pref
  Next Hop[Interface Name]                    Metric
-----
192.0.2.1/32                                  Remote ISIS  00h15m47s 15
192.168.12.1                                  10
192.0.2.2/32                                  Local  Local  00h15m55s  0
system                                         0
192.168.12.0/30                               Local  Local  00h15m55s  0
int-PE-2-PE-1                                 0
192.168.90.2/32                               Remote BGP  00h00m07s 170
192.168.12.1                                  10
192.168.90.3/32                               Remote BGP  00h00m07s 170
192.168.12.1                                  10
192.168.90.4/30                               Remote BGP  00h00m07s 170
192.168.12.1                                  10
-----
No. of Routes: 6
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

Example 4 - BGP IPv4 route leaking from GRT to VPRN per neighbor

[Figure 124: BGP IPv4 route leaking from GRT to VPRN](#) shows the topology for this example, and the corresponding IP addresses. CE-11 exports routes such as 192.168.100.2/32 to the base router and CE-12 exports routes such as 192.168.121.2/32 to the base router. The routes will be leaked from the base router to VPRN 2 if matched by an import policy in the base router of PE-1.

Figure 124: BGP IPv4 route leaking from GRT to VPRN



25966

On PE-1, the following import policy accepts BGP routes and marks them as leakable:

```
# on PE-1:
configure {
  policy-options {
    policy-statement "BGP-Leak" {
      entry 10 {
        from {
          protocol {
            name [bgp]
          }
        }
        action {
          action-type accept
          bgp-leak true
        }
      }
    }
  }
}
```

This import policy is applied for neighbor 172.17.111.2 in the base router, as follows:

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      neighbor "172.17.111.2" {
        group "EBGP_64500to64501_IPv4"
        import {
          policy ["BGP-Leak"]
        }
      }
    }
  }
}
```

The policy is not applied for neighbor 172.17.112.2 on CE-12, so only the routes from CE-11 will be accepted and marked as leakable:

```
[/]
A:admin@PE-1# show router bgp routes
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
u*>i  192.168.100.2/32          None       None
      172.17.111.2          None       0
      64501                  -
u*>i  192.168.100.3/32          None       None
      172.17.111.2          None       0
      64501                  -
u*>i  192.168.100.4/30          None       None
      172.17.111.2          None       0
      64501                  -
i     192.168.121.2/32          None       None
      172.17.112.2          None       0
      64502                  -
i     192.168.121.3/32          None       None
      172.17.112.2          None       0
      64502                  -
i     192.168.121.4/30          None       None
      172.17.112.2          None       0
      64502                  -
-----
Routes : 6
=====
```

The GRT in PE-1 includes BGP routes learned from CE-11, as follows:

```
[/]
A:admin@PE-1# show router route-table
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age           Pref
  Next Hop[Interface Name]  Path-Id Metric
-----
172.17.111.0/30             Local Local  00h21m08s    0
  int-PE-1-CE-11           0
172.17.112.0/30             Local Local  00h21m08s    0
  int-PE-1-CE-12           0
192.0.2.1/32                Local Local  00h21m08s    0
  system                    0
192.0.2.2/32                Remote ISIS  00h20m48s    15
  192.168.12.2              10
192.168.12.0/30             Local Local  00h21m08s    0
  int-PE-1-PE-2            0
=====
```

```

192.168.100.2/32          Remote BGP      00h01m21s 170
    172.17.111.2          0
192.168.100.3/32          Remote BGP      00h01m21s 170
    172.17.111.2          0
192.168.100.4/30          Remote BGP      00h01m21s 170
    172.17.111.2          0
-----
No. of Routes: 8
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

The following command shows that only the routes imported from neighbor CE-11 are marked as leakable in the GRT:

```

[/]
A:admin@PE-1# show router bgp routes ipv4 leakable
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
     Nexthop (Router)                       Path-Id   IGP Cost
     As-Path                                Label
-----
u*>i 192.168.100.2/32                       None      None
     172.17.111.2                           None      0
     64501                                    -
u*>i 192.168.100.3/32                       None      None
     172.17.111.2                           None      0
     64501                                    -
u*>i 192.168.100.4/30                       None      None
     172.17.111.2                           None      0
     64501                                    -
-----
Routes : 3
=====

```

The leakable BGP routes in the GRT can be imported into VPRN 2. The import policy is identical to the import policy in the preceding examples, as follows:

```

# on PE-1:
configure {
  policy-options {
    policy-statement "Import-Leakable-Routes" {
      entry 10 {
        from {
          protocol {
            name [bgp]
          }
        }
        action {
          action-type accept
        }
      }
    }
  }
}

```

```

    }
  }
}

```

This import policy is applied in VPRN 2, as follows:

```

# on PE-1:
configure {
  service {
    vprn "VPRN 2" {
      bgp {
        rib-management {
          ipv4 {
            leak-import {
              policy ["Import-Leakable-Routes"]
            }
          }
        }
      }
    }
  }
}

```

The following command shows the imported leaked BGP routes in VPRN 2. The source of these leaked routes is the base router, not a VPRN.

```

[/]
A:admin@PE-1# show router 2 bgp routes ipv4 leaked
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
      Nexthop (Router)                     Path-Id  IGP Cost
      As-Path                               Label
-----
u*>li 192.168.100.2/32                      100      None
      172.17.111.2 (Base)                 None     0
      64501                                  -
u*>li 192.168.100.3/32                      100      None
      172.17.111.2 (Base)                 None     0
      64501                                  -
u*>li 192.168.100.4/30                      100      None
      172.17.111.2 (Base)                 None     0
      64501                                  -
-----
Routes : 3
=====

```

Any of these leaked BGP routes has the flag "leaked", and the route source is the base router (leaked from base), as follows:

```

[/]
A:admin@PE-1# show router 2 bgp routes 192.168.100.2/32 detail
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -

```

```
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid  
              l - leaked, x - stale, > - best, b - backup, p - purge  
Origin codes  : i - IGP, e - EGP, ? - incomplete
```

```
=====
```

BGP IPv4 Routes

```
=====
```

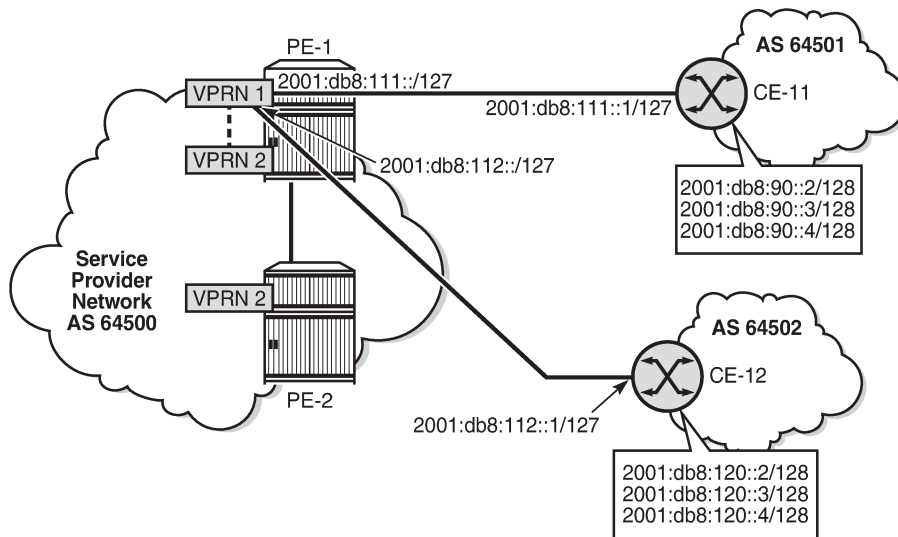
Original Attributes

```
Network       : 192.168.100.2/32  
Nextthop     : 172.17.111.2 (Base)  
Path Id      : None  
From         : BGP Base  
Res. Protocol : LOCAL                Res. Metric   : 0  
Res. Nextthop : 172.17.111.2  
Local Pref.   : 100                  Interface Name : int-PE-1-CE-11  
Aggregator AS : None                Aggregator    : None  
Atomic Aggr.  : Not Atomic          MED           : None  
AIGP Metric   : None                IGP Cost      : 0  
Connector     : None  
Community     : No Community Members  
Cluster       : No Cluster Members  
Originator Id : None                Peer Router Id : 0.0.0.0  
Fwd Class     : None                Priority       : None  
Flags       : Used Valid Best IGP Leaked In-RTM  
Route Source : Leaked from Base  
AS-Path       : 64501  
---snip---
```

Example 5 - BGP IPv6 route leaking between VPRNs. Global VPRN BGP configuration

[Figure 125: BGP IPv6 route leaking between VPRNs](#) shows the topology and the IP addresses used for this example. CE-11 exports routes such as 2001:db8:90::2/128 to VPRN 1 on PE-1, and CE-12 exports routes such as 2001:db8:120::2/128 to VPRN 1 on PE-1.

Figure 125: BGP IPv6 route leaking between VPRNs



25968

The following policy imports BGP routes and marks them as leakable:

```
# on PE-1:
configure {
  policy-options {
    policy-statement "BGP-Leak" {
      entry 10 {
        from {
          protocol {
            name [bgp]
          }
        }
        action {
          action-type accept
          bgp-leak true
        }
      }
    }
  }
}
```

This import policy is applied in the general **bgp** context in VPRN 1:

```
# on PE-1:
configure {
  service {
    vprn "VPRN 1" {
      bgp {
        import {
          policy ["BGP-Leak"]
        }
      }
    }
  }
}
```

The following route table includes three BGP routes exported by CE-11 and three BGP routes exported by CE-12:

```
[/]
```



```
A:admin@PE-1# show router 1 route-table family ipv6

=====
IPv6 Route Table (Service: 1)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]                Type   Proto   Age           Pref
Metric
-----
2001:db8::1:1/128                        Local  Local   00h25m40s    0
system
2001:db8:90::2/128                       Remote BGP     00h01m22s    170
2001:db8:111::1
2001:db8:90::3/128                       Remote BGP     00h01m22s    170
2001:db8:111::1
2001:db8:90::4/126                       Remote BGP     00h01m22s    170
2001:db8:111::1
2001:db8:111::/127                      Local  Local   00h25m40s    0
int-PE-1-CE-11
2001:db8:112::/127                      Local  Local   00h25m40s    0
int-PE-1-CE-12
2001:db8:120::2/128                      Remote BGP     00h01m04s    170
2001:db8:112::1
2001:db8:120::3/128                      Remote BGP     00h01m04s    170
2001:db8:112::1
2001:db8:120::4/126                      Remote BGP     00h01m04s    170
2001:db8:112::1
-----
No. of Routes: 9
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

All the routes imported into the VPRN using BGP are marked as leakable. The following command shows which BGP IPv6 routes are marked as leakable in VPRN 1:

```
[/]
A:admin@PE-1# show router 1 bgp routes ipv6 leakable

=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete

=====
BGP IPv6 Routes
=====
Flag Network
Nexthop (Router)
As-Path                LocalPref MED
Path-Id               IGP Cost
Label
-----
u*>i 2001:db8:90::2/128
2001:db8:111::1
64501                None      None
None                0
-
u*>i 2001:db8:90::3/128
2001:db8:111::1
64501                None      None
None                0
-
u*>i 2001:db8:90::4/126
2001:db8:111::1
64501                None      None
None                0
-
```

```

u*>i 2001:db8:120::2/128      None      None
      2001:db8:112::1        None      0
      64502                   -        -
u*>i 2001:db8:120::3/128      None      None
      2001:db8:112::1        None      0
      64502                   -        -
u*>i 2001:db8:120::4/126      None      None
      2001:db8:112::1        None      0
      64502                   -        -
-----
Routes : 6
=====

```

The BGP leakable routes can be imported into VPRN 2 when the following import policy is configured and applied in VPRN 2:

```

# on PE-1:
configure {
  policy-options {
    policy-statement "Import-Leakable-Routes" {
      entry 10 {
        from {
          protocol {
            name [bgp]
          }
        }
        action {
          action-type accept
        }
      }
    }
  }
}

```

The only difference from IPv4 routes is that the policy is applied to the IPv6 context of the RIB management:

```

# on PE-1:
configure {
  service {
    vprn "VPRN 2" {
      bgp {
        rib-management {
          ipv6 {
            leak-import {
              policy ["Import-Leakable-Routes"]
            }
          }
        }
      }
    }
  }
}

```

The following command shows that the VPRN is importing the leaked BGP IPv6 routes from another VPRN instance:

```

[/]
A:admin@PE-1# show router 2 bgp routes ipv6 leaked
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====

```

```

BGP IPv6 Routes
=====
Flag Network                               LocalPref MED
      Nexthop (Router)                     Path-Id  IGP Cost
      As-Path                               -        Label
-----
u*>li 2001:db8:90::2/128                    100      None
      2001:db8:111::1 (VPRN 1)             None     0
      64501                                  -
u*>li 2001:db8:90::3/128                    100      None
      2001:db8:111::1 (VPRN 1)             None     0
      64501                                  -
u*>li 2001:db8:90::4/126                    100      None
      2001:db8:111::1 (VPRN 1)             None     0
      64501                                  -
u*>li 2001:db8:120::2/128                   100      None
      2001:db8:112::1 (VPRN 1)             None     0
      64502                                  -
u*>li 2001:db8:120::3/128                   100      None
      2001:db8:112::1 (VPRN 1)             None     0
      64502                                  -
u*>li 2001:db8:120::4/126                   100      None
      2001:db8:112::1 (VPRN 1)             None     0
      64502                                  -
-----
Routes : 6
=====

```

The BGP routes have the flag "leaked" and the route source is VPRN 1, as follows:

```

[/]
A:admin@PE-1# show router 2 bgp routes 2001:db8:90::2/128 detail
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv6 Routes
=====
Original Attributes

Network       : 2001:db8:90::2/128
Nexthop      : 2001:db8:111::1 (VPRN 1)
Path Id      : None
From         : BGP VPRN 1
Res. Protocol : LOCAL           Res. Metric   : 0
Res. Nexthop : 2001:db8:111::1
Local Pref.  : 100              Interface Name : int-PE-1-CE-11
Aggregator AS : None           Aggregator    : None
Atomic Aggr. : Not Atomic      MED           : None
AIGP Metric   : None           IGP Cost      : 0
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None           Peer Router Id : 0.0.0.0
Fwd Class     : None           Priority       : None
Flags       : Used Valid Best IGP Leaked In-RTM
Route Source : Leaked from VPRN 1
AS-Path       : 64501

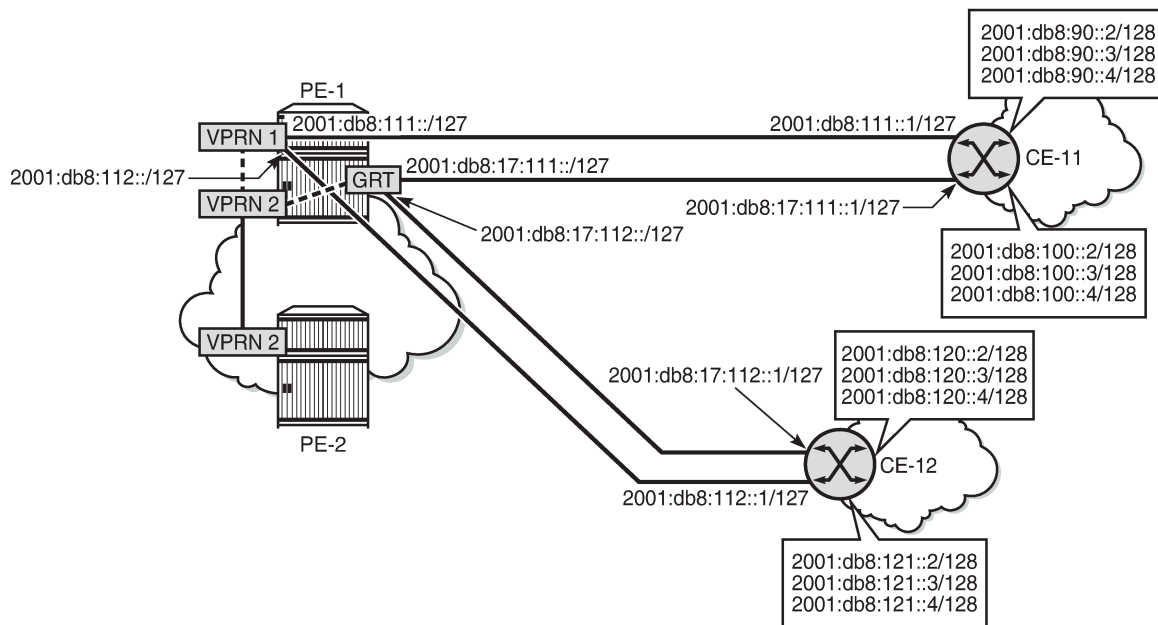
```

---snip---

Example 6 - BGP IPv6 route leaking from GRT to VPRN and from VPRN to VPRN

Figure 126: BGP IPv6 route leaking from GRT and VPRN to VPRN shows the topology and the IPv6 addresses used in this example. CE-11 exports IPv6 routes such as 2001:db8:90::2/128 to VPRN 1 and IPv6 routes such as 2001:db8:100::2/128 to the GRT. CE-12 exports IPv6 routes such as 2001:db8:120::2/128 to VPRN 1 and IPv6 routes such as 2001:db8:121::2/128 to the GRT.

Figure 126: BGP IPv6 route leaking from GRT and VPRN to VPRN



25969

The policy to mark imported BGP routes as leakable can be identical to the policy used in the preceding examples. However, in this case, prefix-lists are added as a filter. VPRN 1 may accept routes such as 2001:db8:90::2/128 and 2001:db8:120::2/128.

```
# on PE-1:
configure {
  policy-options {
    prefix-list "2001:db8:120::" {
      prefix 2001:db8:120::/100 type longer {
      }
    }
    prefix-list "2001:db8:90::" {
      prefix 2001:db8:90::/100 type longer {
      }
    }
  }
  policy-statement "BGP-Leak-VPRN1_90_120" {
    entry 10 {
      from {
        prefix-list ["2001:db8:90::"]
        protocol {
          name [bgp]
        }
      }
    }
  }
}
```

```

    }
  }
  action {
    action-type accept
    bgp-leak true
  }
}
entry 20 {
  from {
    prefix-list ["2001:db8:120::"]
    protocol {
      name [bgp]
    }
  }
  action {
    action-type accept
    bgp-leak true
  }
}
}
}

```

This import policy is applied in the general BGP settings for VPRN 1, as follows:

```

# on PE-1:
configure {
  service {
    vprn "VPRN 1" {
      bgp {
        import {
          policy ["BGP-Leak-VPRN1_90_120"]
        }
      }
    }
  }
}

```

In a similar way, the base router may accept routes such as 2001:8db:100::2/128 and 2001:8db:121::2/128:

```

# on PE-1:
configure {
  policy-options {
    prefix-list "2001:db8:100::" {
      prefix 2001:db8:100::/100 type longer {
      }
    }
    prefix-list "2001:db8:121::" {
      prefix 2001:db8:121::/100 type longer {
      }
    }
  }
  policy-statement "BGP-Leak-Base_100_121" {
    entry 10 {
      from {
        prefix-list ["2001:db8:100::"]
        protocol {
          name [bgp]
        }
      }
      action {
        action-type accept
        bgp-leak true
      }
    }
  }
  entry 20 {
    from {

```

```

        prefix-list ["2001:db8:121::"]
        protocol {
            name [bgp]
        }
    }
    action {
        action-type accept
        bgp-leak true
    }
}

```

This policy is applied in the base router for BGP neighbor 2001:db8:17:111::1 (CE-11), as follows:

```

# on PE-1:
configure {
    router "Base" {
        bgp {
            neighbor "2001:db8:17:111::1" {
                import {
                    policy ["BGP-Leak-Base_100_121"]
                }
            }
        }
    }
}

```

The import policy in the base router is not applied for BGP neighbor 2001:db8::112::1 (CE-12), so only the routes exported by CE-11 will be accepted and marked as leakable. The IPv6 routing table in the base router is as follows:

```

[/]
A:admin@PE-1# show router route-table family ipv6
=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]          Type   Proto   Age      Pref
                                   Metric
-----
2001:db8::1/128
system                             Local  Local   00h31m24s  0
                                   0
2001:db8::2/128
fe80::b:1ff:fe01:1-"int-PE-1-PE-2" Remote  ISIS    00h31m04s  15
                                   10
2001:db8:12::/126
int-PE-1-PE-2                      Local  Local   00h31m23s  0
                                   0
2001:db8:17:111::/127
int-PE-1-CE-11                     Local  Local   00h31m22s  0
                                   0
2001:db8:17:112::/127
int-PE-1-CE-12                     Local  Local   00h31m23s  0
                                   0
2001:db8:100::2/128
2001:db8:17:111::1                 Remote  BGP     00h01m18s  170
                                   0
2001:db8:100::3/128
2001:db8:17:111::1                 Remote  BGP     00h01m18s  170
                                   0
2001:db8:100::4/126
2001:db8:17:111::1                 Remote  BGP     00h01m18s  170
                                   0
-----
No. of Routes: 8
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====

```

The IPv6 routing table for VPRN 1 contains routes exported by CE-11 and CE-12, as follows:

```
[/]
A:admin@PE-1# show router 1 route-table family ipv6

=====
IPv6 Route Table (Service: 1)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]                Type   Proto   Age      Pref
Metric
-----
2001:db8::1:1/128                        Local  Local   00h31m22s  0
system                                  0
2001:db8:90::2/128                       Remote BGP     00h02m03s  170
2001:db8:111::1                          0
2001:db8:90::3/128                       Remote BGP     00h02m03s  170
2001:db8:111::1                          0
2001:db8:90::4/126                       Remote BGP     00h02m03s  170
2001:db8:111::1                          0
2001:db8:111::/127                      Local  Local   00h31m22s  0
int-PE-1-CE-11                          0
2001:db8:112::/127                      Local  Local   00h31m21s  0
int-PE-1-CE-12                          0
2001:db8:120::2/128                     Remote BGP     00h02m03s  170
2001:db8:112::1                          0
2001:db8:120::3/128                     Remote BGP     00h02m03s  170
2001:db8:112::1                          0
2001:db8:120::4/126                     Remote BGP     00h02m03s  170
2001:db8:112::1                          0
-----
No. of Routes: 9
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

The following command shows which routes are marked as leakable in the GRT:

```
[/]
A:admin@PE-1# show router bgp routes ipv6 leakable

=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv6 Routes
=====
Flag Network
Nexthop (Router)                LocalPref  MED
As-Path                          Path-Id    IGP Cost
Label
-----
u*>i 2001:db8:100::2/128          None      None
      2001:db8:17:111::1      None      0
      64501                    -
u*>i 2001:db8:100::3/128          None      None
      2001:db8:17:111::1      None      0
      64501                    -
u*>i 2001:db8:100::4/126          None      None
```

```

2001:db8:17:111::1          None          0
64501                      -
-----
Routes : 3
=====

```

The following command shows which routes are marked as leakable in VPRN 1:

```

[/]
A:admin@PE-1# show router 1 bgp routes ipv6 leakable
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv6 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
u*>i  2001:db8:90::2/128        None       None
      2001:db8:111::1        None       0
      64501                   -
u*>i  2001:db8:90::3/128        None       None
      2001:db8:111::1        None       0
      64501                   -
u*>i  2001:db8:90::4/126        None       None
      2001:db8:111::1        None       0
      64501                   -
u*>i  2001:db8:120::2/128      None       None
      2001:db8:112::1        None       0
      64502                   -
u*>i  2001:db8:120::3/128      None       None
      2001:db8:112::1        None       0
      64502                   -
u*>i  2001:db8:120::4/126      None       None
      2001:db8:112::1        None       0
      64502                   -
-----
Routes : 6
=====

```

On PE-1, a policy is created to import the BGP leakable routes (the same as in the preceding examples), as follows:

```

# on PE-1:
configure {
  policy-options {
    policy-statement "Import-Leakable-Routes" {
      entry 10 {
        from {
          protocol {
            name [bgp]
          }
        }
        action {
          action-type accept
        }
      }
    }
  }
}

```



```

    }
  }
}

```

This import policy is configured for IPv6 routes in VPRN 2, as follows:

```

# on PE-1:
configure {
  service {
    vprn "VPRN 2" {
      bgp {
        rib-management {
          ipv6 {
            leak-import {
              policy ["Import-Leakable-Routes"]
            }
          }
        }
      }
    }
  }
}

```

The following command shows the leaked IPv6 routes in VPRN 2:

```

[/]
A:admin@PE-1# show router 2 bgp routes ipv6 leaked
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv6 Routes
=====
Flag Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>li 2001:db8:90::2/128                    100        None
      2001:db8:111::1 (VPRN 1)             None       0
      64501
u*>li 2001:db8:90::3/128                    100        None
      2001:db8:111::1 (VPRN 1)             None       0
      64501
u*>li 2001:db8:90::4/126                    100        None
      2001:db8:111::1 (VPRN 1)             None       0
      64501
u*>li 2001:db8:100::2/128                   100        None
      2001:db8:17:111::1 (Base)            None       0
      64501
u*>li 2001:db8:100::3/128                   100        None
      2001:db8:17:111::1 (Base)            None       0
      64501
u*>li 2001:db8:100::4/126                   100        None
      2001:db8:17:111::1 (Base)            None       0
      64501
u*>li 2001:db8:120::2/128                   100        None
      2001:db8:112::1 (VPRN 1)            None       0
      64502
u*>li 2001:db8:120::3/128                   100        None
      2001:db8:112::1 (VPRN 1)            None       0
      64502
u*>li 2001:db8:120::4/126                   100        None

```

```

2001:db8:112::1 (VPRN 1)          None          0
64502                             -
-----
Routes : 9
=====

```

Some of these routes are leaked from the base router and some routes are leaked from VPRN 1. The detailed information for any of these leaked routes shows that the flag "Leaked" is present. For route 2001:db8:100::2/128, the route source is the base router, as follows:

```

[/]
A:admin@PE-1# show router 2 bgp routes 2001:db8:100::2/128 detail
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv6 Routes
=====
Original Attributes

Network       : 2001:db8:100::2/128
NextHop       : 2001:db8:17:111::1 (Base)
Path Id       : None
From          : BGP Base
Res. Protocol : LOCAL                Res. Metric   : 0
Res. NextHop  : 2001:db8:17:111::1
Local Pref.   : 100                  Interface Name : int-PE-1-CE-11
Aggregator AS : None                 Aggregator    : None
Atomic Aggr.  : Not Atomic           MED           : None
AIGP Metric   : None                 IGP Cost      : 0
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None                 Peer Router Id : 0.0.0.0
Fwd Class     : None                 Priority       : None
Flags       : Used Valid Best IGP Leaked In-RTM
Route Source : Leaked from Base
AS-Path       : 64501
---snip---

```

For route 2001:db8:90::2/128, the route source is VPRN 1, as follows:

```

[/]
A:admin@PE-1# show router 2 bgp routes 2001:db8:90::2/128 detail
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv6 Routes
=====
Original Attributes

```

```
Network      : 2001:db8:90::2/128
Nexthop     : 2001:db8:111::1 (VPRN 1)
Path Id     : None
From        : BGP VPRN 1
Res. Protocol : LOCAL                      Res. Metric   : 0
Res. Nexthop : 2001:db8:111::1
Local Pref.  : 100                          Interface Name : int-PE-1-CE-11
Aggregator AS : None                       Aggregator    : None
Atomic Aggr. : Not Atomic                   MED           : None
AIGP Metric  : None                         IGP Cost      : 0
Connector    : None
Community    : No Community Members
Cluster      : No Cluster Members
Originator Id : None                       Peer Router Id : 0.0.0.0
Fwd Class    : None                         Priority       : None
Flags      : Used Valid Best IGP Leaked In-RTM
Route Source : Leaked from VPRN 1
AS-Path      : 64501
---snip---
```

Conclusion

BGP provides many ways to manipulate routes. In this example, IPv4/IPv6 routes learned from BGP neighbors could be marked as "leakable" and imported into other routing instances (VPRN to VPRN, VPRN to GRT, GRT to VPRN) without the use of communities in the network policy.

BGP Route Refresh

This chapter describes BGP Route Refresh.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information and configuration in this chapter are based on SR OS Release 20.5.R2. The option to manually trigger BGP ROUTE_REFRESH messages to a BGP peer is supported in SR OS Release 19.7.R1, and later.

In SR OS releases earlier than 19.7.R1, only the automatic route refresh mechanism for VPN routes that carry Route Target extended communities, such as VPN-IPv4, VPN-IPv6, L2-VPN, MVPN-IPv4, or MVPN-IPv6 routes, is supported.

In SR OS releases earlier than 19.7.R1, soft reconfiguration inbound is supported for all non-VPN and VPN address families, using a **clear** command with **soft-inbound** option. With soft reconfiguration inbound, incoming routes are continuously retained in memory (RIB-IN), exactly as they were originally received from a BGP peer. Therefore, when an import policy change happens, the reevaluation of these routes can happen locally. There is no need to involve the peer node, because no route-refresh is involved. The disadvantage is the extra resource consumption to retain a copy of all original routes in memory, even if they are not needed at the current time.

Overview

RFC 2918, *Route Refresh Capability for BGP-4*, describes the BGP ROUTE_REFRESH message type and capability for BGP-4. When BGP router PE-1 sends a route refresh message for a specific address family to its BGP peer PE-2, PE-2 re-advertises all its RIB-OUT routes for PE-1 belonging to that address family. Manually-triggered BGP route refresh can be used for any BGP address family. However, if PE-2 did not advertise the route refresh capability in the BGP OPEN message to PE-1, then PE-2 ignores the incoming ROUTE_REFRESH message from PE-1.

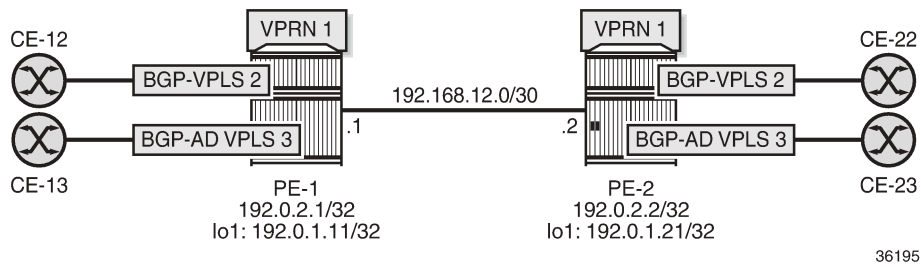
In this chapter, the following use cases are shown:

- Automatic route refresh for VPN-IP and L2-VPN routes after an import policy is modified
- Block automatic route refresh for VPN-IP routes (**mp-bgp-keep** option)
- Manual route refresh for BGP routes for different address families (**soft-route-refresh** option in **clear** command)

Configuration

[Figure 127: Example topology](#) shows the example topology with two nodes.

Figure 127: Example topology



The initial configuration on the nodes includes:

- Cards, MDAs, ports
- Router interfaces
- SR-ISIS

The following route policies are configured on PE-1; the policies on PE-2 are similar.

```
# on PE-1:
configure {
  policy-options {
    community "target:64500:1" {
      member "target:64500:1" { }
    }
    community "target:64500:2" {
      member "target:64500:2" { }
    }
    prefix-list "192.0.1.0/24" {
      prefix 192.0.1.0/24 type range {
        start-length 32
        end-length 32
      }
    }
  }
  policy-statement "export-VPLS2" {
    entry 10 {
      action {
        action-type accept
        community {
          add ["target:64500:2"]
        }
      }
    }
  }
  policy-statement "export-VPRN1" {
    entry 10 {
      action {
        action-type accept
        next-hop 192.0.1.11
        community {
          add ["target:64500:1"]
        }
      }
    }
  }
  policy-statement "export-bgp" {
    entry 10 {
      from {
        prefix-list ["192.0.1.0/24"]
      }
    }
  }
}
```

```

    }
    action {
        action-type accept
    }
}
}
policy-statement "import-VPLS2" {
    entry 10 {
        from {
            family [l2-vpn]
            community {
                name "target:64500:2"
            }
        }
        action {
            action-type accept
        }
    }
    default-action {
        action-type reject
    }
}
policy-statement "import-VPRN1" {
    entry 10 {
        from {
            community {
                name "target:64500:1"
            }
            protocol {
                name [bgp-vpn]
            }
        }
        action {
            action-type accept
        }
    }
    default-action {
        action-type reject
    }
}
}
}

```

Two BGP groups are configured: one for the VPN-IPv4 and Label-IPv4 address families and another for the L2-VPN address family. The BGP configuration for the base router on PE-1 is as follows:

```

# on PE-1:
configure {
    router "Base" {
        autonomous-system 64500
        bgp {
            split-horizon true
            next-hop-resolution {
                labeled-routes {
                    transport-tunnel {
                        family label-ipv4 {
                            resolution-filter {
                                ldp false
                                sr-isis true
                            }
                        }
                    }
                }
            }
        }
    }
    group "iBGP-L2" {

```

```

        type internal
        local-address 192.0.1.11
        family {
            l2-vpn true
        }
    }
    group "iBGPv4" {
        peer-as 64500
        family {
            vpn-ipv4 true
            label-ipv4 true
        }
    }
    neighbor "192.0.1.21" {
        group "iBGP-L2"
    }
    neighbor "192.0.2.2" {
        group "iBGPv4"
        export {
            policy ["export-bgp"]
        }
    }
}

```

The service configuration on PE-1 is as follows:

```

# on PE-1:
configure {
    service {
        pw-template "PW1" {
            pw-template-id 1
        }
        vpls "BGP-AD VPLS 3" {
            admin-state enable
            service-id 3
            customer "1"
            bgp 1 {
                route-distinguisher "64500:3"
                route-target {
                    export "target:64500:3"
                    import "target:64500:3"
                }
                pw-template-binding "PW1" {
                }
            }
            bgp-ad {
                admin-state enable
                vpls-id "64500:3"
                vsi-id-prefix 192.0.1.11
            }
            sap 1/2/1:3 {
            }
        }
        vpls "BGP-VPLS 2" {
            admin-state enable
            service-id 2
            customer "1"
            bgp 1 {
                route-distinguisher "64500:2"
                vsi-import ["import-VPLS2"]
                vsi-export ["export-VPLS2"]
                route-target {
                    export "target:64500:2"
                }
            }
        }
    }
}

```

```

        import "target:64500:2"
    }
    pw-template-binding "PW1" {
        import-rt ["target:64500:2"]
    }
}
bgp-vpls {
    admin-state enable
    maximum-ve-id 100
    ve {
        name "PE-1"
        id 1
    }
}
sap 1/2/1:2 {
}
}
vprn "VPRN 1" {
    admin-state enable
    service-id 1
    customer "1"
    route-distinguisher "64500:1"
    vrf-target {
        community "target:64500:1"
    }
    vrf-import {
        policy ["import-VPRN1"]
    }
    vrf-export {
        policy ["export-VPRN1"]
    }
    auto-bind-tunnel {
        resolution filter
    }
    bgp {
        next-hop-resolution {
            use-bgp-routes true
        }
    }
    interface "lo1" {
        loopback true
        ipv4 {
            primary {
                address 172.31.1.1
                prefix-length 32
            }
        }
    }
}
}
}

```

The following BGP OPEN message sent by PE-1 includes the route refresh capability for two BGP address families:

```

1 2020/06/23 09:03:58.168 UTC MINOR: DEBUG #2001 Base BGP
"BGP: OPEN
Peer 1: 192.0.2.2 - Send (Passive) BGP OPEN: Version 4
AS Num 64500: Holdtime 90: BGP_ID 192.0.2.1: Opt Length 26 (ExtOpt F)
Opt Para: Type CAPABILITY: Length = 24: Data:
  Cap_Code GRACEFUL-RESTART: Length 2
  Bytes: 0x0 0x78
  Cap_Code MP-BGP: Length 4
  Bytes: 0x0 0x1 0x0 0x80 # AFI / SAFI ; 1 / 128 ; vpn-ipv4
  Cap_Code MP-BGP: Length 4

```



```

Bytes: 0x0 0x1 0x0 0x4 # AFI / SAFI ; 1 / 4 ; label-ipv4
Cap_Code ROUTE-REFRESH: Length 0
Cap_Code 4-OCTET-ASN: Length 4
Bytes: 0x0 0x0 0xfb 0xf4
"

```

The BGP session between PE-1 and PE-2 includes the route refresh capability, as follows. No route refresh messages have been triggered manually yet.

```

[]
A:admin@PE-1# show router bgp neighbor 192.0.2.2 | match RtRefresh
Input RtRefresh      : 0          Output RtRefresh      : 0
Local Capability    : RtRefresh MPBGP 4byte ASN
Remote Capability   : RtRefresh MPBGP 4byte ASN

```

PE-1 receives the following BGP Labeled Unicast (BGP-LU) route:

```

[]
A:admin@PE-1# show router bgp routes label-ipv4
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP Routes
=====
Flag Network                               LocalPref MED
      Nexthop (Router)                     Path-Id  IGP Cost
      As-Path                               Label
-----
u*>i 192.0.1.21/32                          100     None
      192.0.2.2                             None    10
      No As-Path                            524274
-----
Routes : 1
=====

```

PE-1 receives the following VPN-IPv4 route for VPRN 1:

```

[]
A:admin@PE-1# show router bgp routes vpn-ipv4
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag Network                               LocalPref MED
      Nexthop (Router)                     Path-Id  IGP Cost
      As-Path                               Label
-----
u*>i 64500:1:172.31.1.2/32                  100     None
      192.0.1.21                           None    0
-----

```

```

No As-Path                                     524286
-----
Routes : 1
=====

```

PE-1 receives one L2-VPN route for BGP-VPLS 2 and one L2-VPN route for BGP-AD VPLS 3:

```

[]
A:admin@PE-1# show router bgp routes l2-vpn
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP L2VPN Routes
=====
Flag RouteType      Prefix      MED
      RD            SiteId
      Nexthop       VeId
      As-Path       BaseOffset  BlockSize  LocalPref
                        vplsLabelBa
                        se
-----
u*>i VPLS              -            0
      64500:2        -            -
      192.0.1.21    2            8            100
      No As-Path    1            524278
u*>i AutoDiscovery   192.0.1.21  -            0
      64500:3        -            -
      192.0.1.21    -            -            100
      No As-Path    -            -
-----
Routes : 2
=====

```

Automatic route refresh for VPN-IP and L2-VPN routes

The following import policy is modified on PE-1; the "import-VPRN1" policy action sets the local preference to a value of 200:

```

# on PE-1:
configure {
  policy-options {
    policy-statement "import-VPRN1" {
      entry 10 {
        from {
          community {
            name "target:64500:1"
          }
          protocol {
            name [bgp-vpn]
          }
        }
        action {
          action-type accept
          local-preference 200
        }
      }
    }
  }
}

```

```
    }  
    default-action {  
        action-type reject  
    }  
}
```

When one or more import policies are modified after the VPN-IP and L2-VPN routes have been received, the node automatically generates route refresh messages for VPN-IP and L2-VPN routes to its peers. In this case, PE-1 sends one route refresh message for VPN-IPv4 routes and one route refresh message for L2-VPN routes to its BGP peer PE-2. When debugging is enabled for BGP route refresh messages, the following debug messages are logged on PE-1:

```
18 2020/06/23 09:14:47.611 UTC MINOR: DEBUG #2001 Base Peer 1: 192.0.2.2  
"Peer 1: 192.0.2.2: ROUTE REFRESH  
Peer 1: 192.0.2.2 - Send BGP ROUTE REFRESH:  
Address Family AFI_IPV4: Sub AFI SAFI_VPN  
"  
  
19 2020/06/23 09:14:47.611 UTC MINOR: DEBUG #2001 Base Peer 1: 192.0.1.21  
"Peer 1: 192.0.1.21: ROUTE REFRESH  
Peer 1: 192.0.1.21 - Send BGP ROUTE REFRESH:  
Address Family AFI_L2VPN: Sub AFI SAFI_VPLS  
"
```

The first route refresh message triggers VPN-IPv4 routes to be re-advertised by the peer, while the second route refresh message triggers L2-VPN routes to be re-advertised. With these BGP route refresh messages, all VPN-IPv4 and L2-VPN routes are refreshed, even for services without an import policy, such as BGP-AD VPLS 3. The first of the following routes is related to VPRN 1 (with route-target target:64500:1), the second to BGP-VPLS 2 (with route-target target:64500:2), and the third to BGP-AD VPLS 3 (with route-target target:64500:3):

```
20 2020/06/23 09:14:47.614 UTC MINOR: DEBUG #2001 Base Peer 1: 192.0.2.2  
"Peer 1: 192.0.2.2: UPDATE  
Peer 1: 192.0.2.2 - Received BGP UPDATE:  
  Withdrawn Length = 0  
  Total Path Attr Length = 62  
  Flag: 0x90 Type: 14 Len: 33 Multiprotocol Reachable NLRI:  
    Address Family VPN_IPV4  
    NextHop len 12 NextHop 192.0.1.21  
    172.31.1.2/32 RD 64500:1 Label 524286  
  Flag: 0x40 Type: 1 Len: 1 Origin: 0  
  Flag: 0x40 Type: 2 Len: 0 AS Path:  
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100  
  Flag: 0xc0 Type: 16 Len: 8 Extended Community:  
    target:64500:1  
"
```

```
21 2020/06/23 09:14:47.614 UTC MINOR: DEBUG #2001 Base Peer 1: 192.0.1.21  
"Peer 1: 192.0.1.21: UPDATE  
Peer 1: 192.0.1.21 - Received BGP UPDATE:  
  Withdrawn Length = 0  
  Total Path Attr Length = 72  
  Flag: 0x90 Type: 14 Len: 28 Multiprotocol Reachable NLRI:  
    Address Family L2VPN  
    NextHop len 4 NextHop 192.0.1.21  
    [VPLS/VPWS] preflen 17, veid: 2, vbo: 1, vbs: 8, label-base: 524278, RD 64500:2  
  Flag: 0x40 Type: 1 Len: 1 Origin: 0  
  Flag: 0x40 Type: 2 Len: 0 AS Path:  
  Flag: 0x80 Type: 4 Len: 4 MED: 0  
"
```

```

Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
Flag: 0xc0 Type: 16 Len: 16 Extended Community:
      target:64500:2
      l2-vpn/vrf-imp:Encap=19: Flags=none: MTU=1514: PREF=0
"

```

```

22 2020/06/23 09:14:47.614 UTC MINOR: DEBUG #2001 Base Peer 1: 192.0.1.21
"Peer 1: 192.0.1.21: UPDATE
Peer 1: 192.0.1.21 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 67
  Flag: 0x90 Type: 14 Len: 23 Multiprotocol Reachable NLRI:
    Address Family L2VPN
    NextHop len 4 NextHop 192.0.1.21
    [AD] 192.0.1.21/32, RD 64500:3
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 0 AS Path:
  Flag: 0x80 Type: 4 Len: 4 MED: 0
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
  Flag: 0xc0 Type: 16 Len: 16 Extended Community:
    target:64500:3
    l2-vpn/vrf-imp:64500:3
"

```

Block automatic route refresh for VPN-IP routes

When the VPN-IP routes do not need to be re-advertised when an import policy is modified, the **mp-bgp-keep** option can be configured in the generic **bgp** context of the base router, as follows:

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      mp-bgp-keep true
    }
  }
}

```

Change the import policy back to the original configuration, as follows:

```

# on PE-1:
configure {
  policy-options {
    policy-statement "import-VPRN1" {
      entry 10 {
        from {
          community {
            name "target:64500:1"
          }
          protocol {
            name [bgp-vpn]
          }
        }
        action {
          action-type accept
          delete local-preference # do not modify LP
        }
      }
      default-action {
        action-type reject
      }
    }
  }
}

```

The **mp-bgp-keep true** option blocks the route refresh message for the VPN-IP routes, but not for the L2-VPN routes. The following route refresh message is sent by PE-1:

```
35 2020/06/23 09:21:33.951 UTC MINOR: DEBUG #2001 Base Peer 1: 192.0.1.21
"Peer 1: 192.0.1.21: ROUTE REFRESH
Peer 1: 192.0.1.21 - Send BGP ROUTE REFRESH:
  Address Family AFI_L2VPN: Sub AFI SAFI_VPLS
"
```

Therefore, PE-1 receives the following refreshed L2-VPN routes from PE-2:

```
36 2020/06/23 09:21:33.954 UTC MINOR: DEBUG #2001 Base Peer 1: 192.0.1.21
"Peer 1: 192.0.1.21: UPDATE
Peer 1: 192.0.1.21 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 67
  Flag: 0x90 Type: 14 Len: 23 Multiprotocol Reachable NLRI:
    Address Family L2VPN
    NextHop len 4 NextHop 192.0.1.21
    [AD] 192.0.1.21/32, RD 64500:3
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 0 AS Path:
  Flag: 0x80 Type: 4 Len: 4 MED: 0
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
  Flag: 0xc0 Type: 16 Len: 16 Extended Community:
    target:64500:3
    l2-vpn/vrf-imp:64500:3
"
```

```
37 2020/06/23 09:21:33.954 UTC MINOR: DEBUG #2001 Base Peer 1: 192.0.1.21
"Peer 1: 192.0.1.21: UPDATE
Peer 1: 192.0.1.21 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 72
  Flag: 0x90 Type: 14 Len: 28 Multiprotocol Reachable NLRI:
    Address Family L2VPN
    NextHop len 4 NextHop 192.0.1.21
    [VPLS/VPWS] preflen 17, veid: 2, vbo: 1, vbs: 8, label-base: 524278, RD 64500:2
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 0 AS Path:
  Flag: 0x80 Type: 4 Len: 4 MED: 0
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
  Flag: 0xc0 Type: 16 Len: 16 Extended Community:
    target:64500:2
    l2-vpn/vrf-imp:Encap=19: Flags=none: MTU=1514: PREF=0
"
```

Manually-triggered route refresh for any BGP address family

A manual route refresh can be triggered by the **soft-route-refresh** option using the **clear** operation. This command can be launched for any address family. The command will look like the following:

```
[ ]
A:admin@PE-1# clear router bgp neighbor {<ip-address>|as <as-number>|external|all}
  soft-route-refresh [<family>]
```

```
<family> : ipv4|vpn-ipv4|ipv6|mcast-ipv4|vpn-ipv6|l2-vpn|mvpn-ipv4|mdt-safi|flow-
ipv4|ms-pw|route-target|mcast-vpn-ipv4|mvpn-ipv6|flow-ipv6|evpn|mcast-ipv6|label-ipv4|label-
ipv6|mcast-vpn-ipv6|bgp-ls|sr-policy-ipv4
```

For example, the following command on PE-1 clears the BGP-LU routes from PE-1:

```
[ ]
A:admin@PE-1# clear router bgp neighbor 192.0.2.2 soft-route-refresh label-ipv4
```

The preceding command triggers the following route refresh message for the BGP-LU routes:

```
38 2020/06/23 09:23:48.951 UTC MINOR: DEBUG #2001 Base Peer 1: 192.0.2.2
"Peer 1: 192.0.2.2: ROUTE REFRESH
Peer 1: 192.0.2.2 - Send BGP ROUTE REFRESH:
Address Family AFI_IPV4: Sub AFI SAFI_MPLS_LABEL
"
```

The following BGP-LU route is received by PE-1:

```
39 2020/06/23 09:23:48.954 UTC MINOR: DEBUG #2001 Base Peer 1: 192.0.2.2
"Peer 1: 192.0.2.2: UPDATE
Peer 1: 192.0.2.2 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 35
  Flag: 0x90 Type: 14 Len: 17 Multiprotocol Reachable NLRI:
    Address Family LBL-IPV4
    NextHop len 4 NextHop 192.0.2.2
    192.0.1.21/32 Label 524274
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 0 AS Path:
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
"
```

The following command on PE-1 shows that one output route refresh message is sent:

```
[ ]
A:admin@PE-1# show router bgp neighbor 192.0.2.2 | match RtRefresh
Input RtRefresh      : 0          Output RtRefresh      : 1
Local Capability     : RtRefresh MPBGP 4byte ASN
Remote Capability    : RtRefresh MPBGP 4byte ASN
```

A similar command on PE-2 shows that one input route refresh message has been received:

```
[ ]
A:admin@PE-2# show router bgp neighbor 192.0.2.1 | match RtRefresh
Input RtRefresh      : 1          Output RtRefresh      : 0
Local Capability     : RtRefresh MPBGP 4byte ASN
Remote Capability    : RtRefresh MPBGP 4byte ASN
```

When the **soft-route-refresh** option is executed without a specific address family, the BGP routes are refreshed for all negotiated address families with that neighbor:

```
[ ]
A:admin@PE-1# clear router bgp neighbor 192.0.2.2 soft-route-refresh
# BGP- LU, BGP-VPN

[ ]
A:admin@PE-1# clear router bgp neighbor 192.0.1.21 soft-route-refresh # L2-VPN
```

The preceding **clear** commands trigger the following BGP ROUTE_REFRESH messages:

```
42 2020/06/23 09:39:53.836 CEST MINOR: DEBUG #2001 Base Peer 1: 192.0.1.21
"Peer 1: 192.0.1.21: ROUTE_REFRESH
Peer 1: 192.0.1.21 - Send BGP ROUTE_REFRESH:
Address Family AFI_L2VPN: Sub AFI SAFI_VPLS
"
```

```
43 2020/06/23 09:39:53.836 CEST MINOR: DEBUG #2001 Base Peer 1: 192.0.2.2
"Peer 1: 192.0.2.2: ROUTE_REFRESH
Peer 1: 192.0.2.2 - Send BGP ROUTE_REFRESH:
Address Family AFI_IPV4: Sub AFI SAFI_VPN
"
```

```
44 2020/06/23 09:39:53.836 CEST MINOR: DEBUG #2001 Base Peer 1: 192.0.2.2
"Peer 1: 192.0.2.2: ROUTE_REFRESH
Peer 1: 192.0.2.2 - Send BGP ROUTE_REFRESH:
Address Family AFI_IPV4: Sub AFI SAFI_MPLS_LABEL
"
```

Conclusion

The **soft-route-refresh** option in the **clear router bgp neighbor** command keeps a BGP session up and sends one or more ROUTE_REFRESH messages to the peer, each requesting the peer to resend all RIB-OUT routes for a specific address family (or for all established address families for a BGP neighbor). This option can be used to debug and troubleshoot route advertisement issues.

BGP Unresolved Route Leaking from Base Router to VPRN

This chapter describes BGP unresolved route leaking from base router to VPRN.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

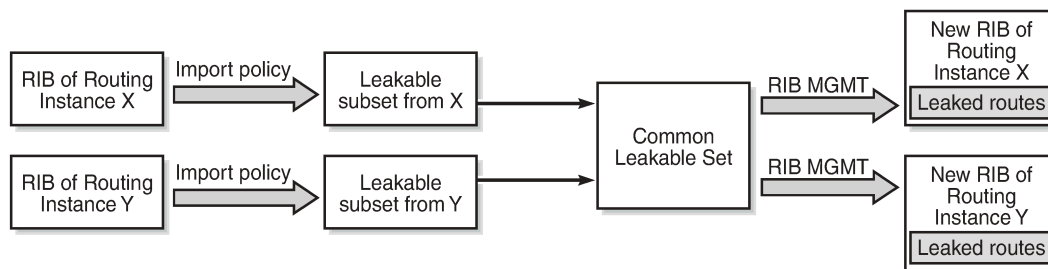
Applicability

The information and configuration in this chapter are based on SR OS Release 22.10.R2. BGP resolved route leaking between BGP routing instances is supported in SR OS Release 12.0.R7, and later; BGP unresolved route leaking from base router to VPRN is supported in SR OS Release 19.10.R1, and later.

Overview

The [BGP Route Leaking](#) chapter describes how BGP resolved routes can be leaked from one BGP routing instance to other BGP routing instances; for example, from the base router to a VPRN, from one VPRN to another VPRN, or from a VPRN to the base router. The first BGP routing instance (source) makes selected BGP routes in its RIB-IN leakable, so that these routes are available for import by BGP in other routing instances (destinations). [Figure 128: BGP route leaking process between BGP routing instances X and Y](#) shows the BGP route leaking process between BGP routing instances.

Figure 128: BGP route leaking process between BGP routing instances X and Y



25963

In SR OS Releases earlier than 19.10.R1, a BGP route is leakable if it meets the following conditions:

- It must have been received from a BGP neighbor and matched by a BGP import policy that accepts the route with a **bgp-leak true** action.
- It must have a BGP next-hop that is resolved by a route or tunnel belonging to the source routing instance.

Those leakable BGP routes can be imported into other destination BGP routing instances. A BGP RIB imports a leakable BGP route when it has a **leak-import** policy that matches and accepts the route.

Leaked BGP routes are compared to other (leaked and non-leaked) BGP routes for the same prefix to come up with the best path, Equal Cost Multi-Path (ECMP), backup path, and so on. A leaked route can be advertised to BGP peers of the importing BGP instance. A leaked route imported into a VPRN BGP instance can even be re-advertised as a VPN-IP route subject to the **vrf-export** policies of the VPRN.

The following use cases require that unresolved BGP routes are leaked from base router to VPRN. To avoid per-VPRN BGP sessions, a Route Reflector (RR) advertises BGP routes toward a PE over a single BGP session with the base router, even though some of the routes belong to VPRNs of the PE. The PE can determine the VPRN owner of a route from an attached community value. The BGP routes that belong to VPRNs can be marked as leakable in the base router, then imported into the correct VPRN based on community matching in the **leak-import** policies.

When the RR advertises a BGP route intended for a VPRN, the BGP next-hop of the route is resolvable in the VPRN instance, but not in the base router. The **allow-unresolved-leaking true** command must be added to the **BGP next-hop-resolution** context for the base router to allow any leakable route to be imported into any VPRN, even when the BGP next-hop is unresolved. The BGP next-hop is resolved as follows:

- If the next-hop of a valid BGP route is resolvable in the base router, any VPRN that imports the route uses the next-hop resolution result of the base router, even if that VPRN is also able to resolve the BGP next-hop using its own routing table.
- If the next-hop of a valid BGP route is unresolvable in the base router and **allow-unresolved-leaking** is set to true, any VPRN can import the route. A VPRN that imports the route then uses its own routing table to resolve the BGP next-hop:
 - By default, the importing VPRN can only use IGP routes, such as OSPFv2, OSPFv3, IS-IS, RIP, RIPng, and static routes to resolve the BGP next-hop of the leaked route.
 - If **use-bgp-routes true** is configured in the **BGP next-hop-resolution** context, the importing VPRN can also use BGP and BGP-VPN routes to resolve the BGP next-hop of the leaked route.

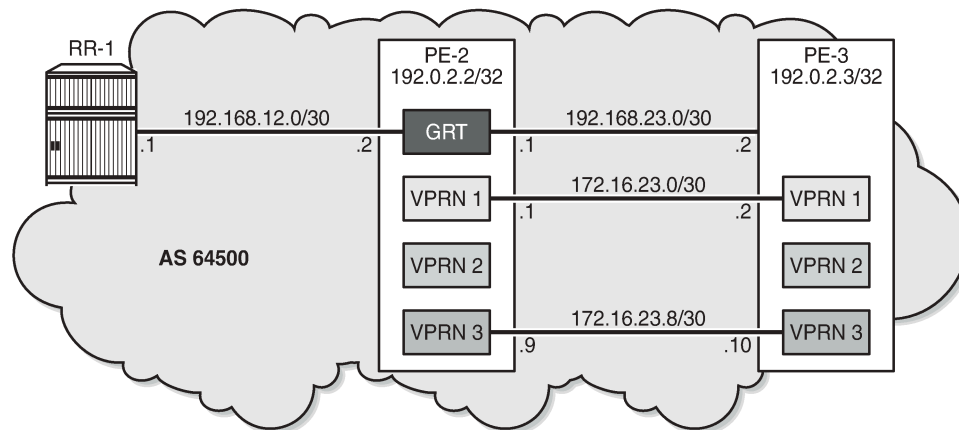
If a leaked BGP route is resolved by a VPRN, the VPRN can re-advertise the route to VPRN BGP peers or export the route as a VPN-IP route. However, if a leaked route is resolved over a BGP-VPN route, it can only be exported as a VPN-IP route if **allow-bgp-vpn-export** is enabled in the VPRN.

If a BGP route is invalid in the base router for reasons other than next-hop reachability, it is not leakable into any VPRN, regardless of the **allow-unresolved-leaking** setting.

Configuration

[Figure 129: Example topology](#) shows the example topology with an RR and two PEs.

Figure 129: Example topology



35961

The initial configuration on the PEs includes the following:

- Cards, MDAs, ports
- Router interfaces
- SR-ISIS

The initial configuration on PE-2 is as follows:

```
# on PE-2:
configure {
  router "Base" {
    autonomous-system 64500
    interface "int-PE-2-PE-3" {
      port 1/1/c1/1:100
      ipv4 {
        primary {
          address 192.168.23.1
          prefix-length 30
        }
      }
    }
    interface "int-PE-2-RR-1" {
      port 1/1/c1/3:100
      ipv4 {
        primary {
          address 192.168.12.2
          prefix-length 30
        }
      }
    }
    interface "system" {
      ipv4 {
        primary {
          address 192.0.2.2
          prefix-length 32
        }
      }
    }
  }
  mpls-labels {
    sr-labels {
```



```
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
```

```
=====
BGP IPv4 Routes
=====
```

Flag	Network NextHop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
i	10.14.0.0/16 10.13.0.1 64501	100 None	None 0 -
i	10.24.0.0/16 10.23.0.1 No As-Path	100 None	None 0 -
i	10.34.0.0/16 10.33.0.1 64503	100 None	None 0 -

```
-----
Routes : 3
=====
```

These routes are invalid in the base router because the next-hop is unresolved, as indicated by the flags in the BGP route details:

```
[/]
A:admin@PE-2# show router bgp routes hunt | match Flags
Flags          : Invalid IGP Nexthop-Unresolved
Flags          : Invalid IGP Nexthop-Unresolved
Flags          : Invalid IGP Nexthop-Unresolved
```

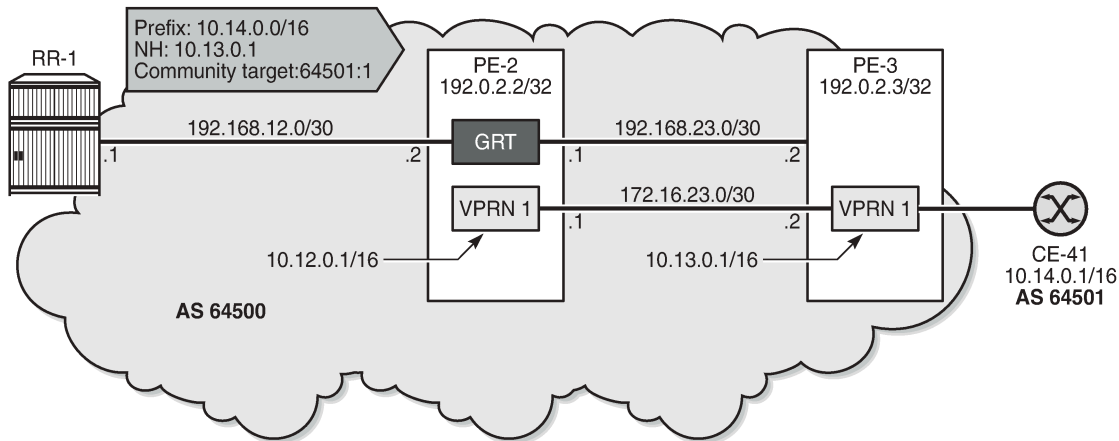
On PE-2, the following import policy is created to make the prefixes leakable:

```
# on PE-2:
configure {
  policy-options {
    prefix-list "10.0.0.0/8" {
      prefix 10.0.0.0/8 type longer {
      }
    }
    policy-statement "leak-10.x" {
      entry 10 {
        from {
          prefix-list ["10.0.0.0/8"]
        }
        action {
          action-type accept
          bgp-leak true
        }
      }
    }
  }
  router "Base" {
    bgp {
      group "iBGP" {
        peer-as 64500
        family {
          ipv4 true
        }
      }
    }
  }
}
```


Use case 1: BGP route leaked to VPRN 1 with next-hop resolved using IS-IS

Figure 130: Leaked route 10.14.0.0/16 with next-hop resolved in VPRN 1 using IS-IS shows that RR-1 advertises prefix 10.14.0.0/16 with next-hop 10.13.0.0/16, which is unresolvable in the base router of PE-2, but can be resolved in VPRN 1.

Figure 130: Leaked route 10.14.0.0/16 with next-hop resolved in VPRN 1 using IS-IS



35962

On PE-3, VPRN 1 has a loopback interface "lo1" configured with IP address 10.13.0.1/32. IS-IS on PE-3 is only enabled on the loopback interface and on the interface facing VPRN 1 on PE-2, not on the interface toward CE-41. VPRN 1 is configured as follows:

```
# on PE-3:
configure {
  service {
    vprn "VPRN 1" {
      admin-state enable
      service-id 1
      customer "1"
      autonomous-system 64500
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "64500:1"
          vrf-target {
            community "target:64500:1"
          }
        }
      }
    }
  }
  interface "int-VPRN1-PE-3-PE-2" {
    ipv4 {
      primary {
        address 172.16.23.2
        prefix-length 30
      }
    }
    sap 1/1/c1/2:1 {
    }
  }
  interface "int-VPRN3-PE-3-CE-41" {
    ipv4 {
      primary {
```


PE-2 receives the following BGP route from RR-1 in the base routing instance with community "target:64500:1":

```
[/]
A:admin@PE-2# show router bgp routes community target:64500:1
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
i     10.14.0.0/16           100        None
      10.13.0.1             None        0
      64501                  -
-----
Routes : 1
=====
```

This route is leakable:

```
[/]
A:admin@PE-2# show router bgp routes community target:64500:1 hunt | match Flags
Flags          : Invalid IGP NextHop-Unresolved Leakable
```

On PE-2, the following **leak-import** policy is configured in VPRN 1 to import the leakable routes with community "target:64500:1":

```
# on PE-2:
configure {
  policy-options
    community "target:64500:1" {
      member "target:64500:1" { }
    }
    policy-statement "leak-import-1" {
      entry 10 {
        from {
          community {
            name "target:64500:1"
          }
        }
        action {
          action-type accept
        }
      }
      default-action {
        action-type reject
      }
    }
  }
}
service {
  vprn "VPRN 1" {
    admin-state enable
    service-id 1
  }
}
```



```

customer "1"
  autonomous-system 64500
  bgp-ipvpn {
    mpls {
      admin-state enable
      route-distinguisher "64500:1"
      vrf-target {
        community "target:64500:1"
      }
    }
  }
  bgp {
    rib-management {
      ipv4 {
        leak-import {
          policy ["leak-import-1"]
        }
      }
    }
  }
}

```

By default, the base router does not leak unresolved routes, so the list of leaked BGP routes in VPRN 1 remains empty:

```

[/]
A:admin@PE-2# show router 1 bgp routes ipv4 leaked
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                       Path-Id    IGP Cost
      As-Path                                Label
-----
No Matching Entries Found.
=====

```

The **allow-unresolved-leaking true** command in the **BGP next-hop resolution** context of the base router allows unresolved BGP routes to be leaked:

```

# on PE-2:
configure {
  router "Base" {
    bgp {
      next-hop-resolution {
        allow-unresolved-leaking true
      }
    }
  }
}

```

When routes with unresolved BGP next-hop in the base router are leaked, VPRN 1 receives the BGP route for prefix 10.14.0.0/16, and the next-hop can be resolved in the VPRN, so the leaked route is valid, best, and used:

```
[/]
A:admin@PE-2# show router 1 bgp routes ipv4 leaked
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path              Label
-----
u*>li 10.14.0.0/16              100        None
      10.13.0.1 (Base)      None        10
      64501                  -
-----
Routes : 1
=====
```

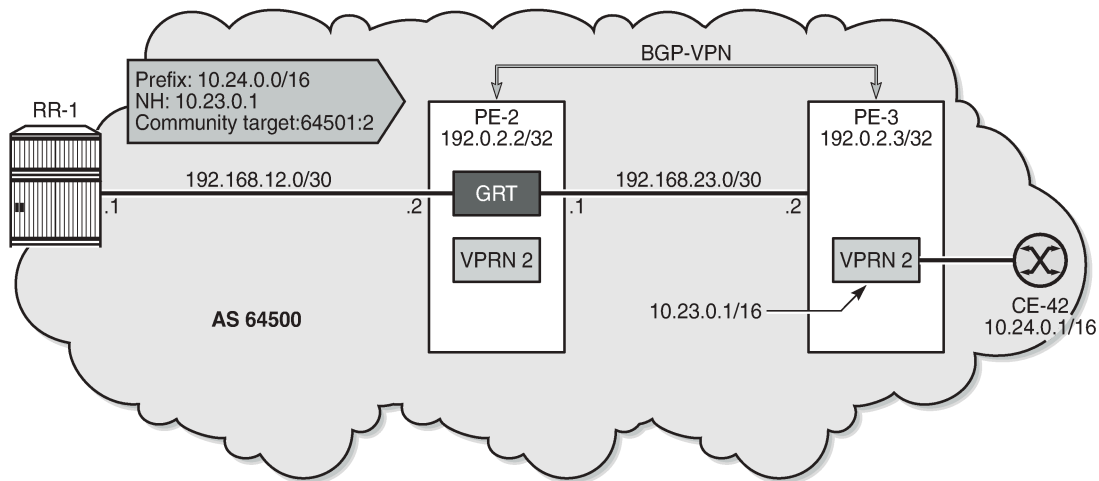
The route table for VPRN 1 includes a BGP route for prefix 10.14.0.0/16 with next-hop 172.16.23.2:

```
[/]
A:admin@PE-2# show router 1 route-table
=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]          Type  Proto  Age          Pref
      Next Hop[Interface Name]      Metric
-----
10.12.0.1/32                Local  Local  00h16m47s    0
      lo1
10.13.0.1/32                Remote  ISIS   00h16m14s    15
      172.16.23.2
10.14.0.0/16              Remote  BGP   00h00m17s    170
      172.16.23.2
172.16.23.0/30              Local  Local  00h16m47s    0
      int-VPRN1-PE-2-PE-3
-----
No. of Routes: 4
---snip---
=====
```

Use case 2: BGP route leaked to VPRN 2 with next-hop resolved using VPN-IP

Figure 131: Leaked route 10.24.0.0/16 with next-hop resolved in VPRN 2 using VPN-IP shows that RR-1 advertises prefix 10.24.0.0/16 with next-hop 10.23.0.1 while PE-3 advertises prefix 10.23.0.1/32 in a VPN-IP route to PE-2.

Figure 131: Leaked route 10.24.0.0/16 with next-hop resolved in VPRN 2 using VPN-IP



35963

On PE-3, VPRN 2 has a loopback interface "lo1" configured with IP address 10.23.0.1/32, which is the BGP next-hop of the leakable route received from RR-1. VPRN 2 is configured with auto-bind-tunnel with resolution to SR-ISIS tunnels.

```
# on PE-3:
configure {
  service {
    vprn "VPRN 2" {
      admin-state enable
      service-id 2
      customer "1"
      autonomous-system 64500
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "64500:2"
          vrf-target {
            community "target:64500:2"
          }
          auto-bind-tunnel {
            resolution filter
            resolution-filter {
              sr-isis true
            }
          }
        }
      }
    }
  }
  interface "lo1" {
    loopback true
    ipv4 {
      primary {
        address 10.23.0.1
        prefix-length 32
      }
    }
  }
}
}
```

```
}

```

Prefix 10.23.0.1/32 is advertised in a VPN-IPv4 route to PE-2. On PE-3, the BGP configuration is as follows:

```
# on PE-3:
configure {
  router "Base" {
    bgp {
      split-horizon true
      group "iBGP-VPN" {
        peer-as 64500
        family {
          vpn-ipv4 true
        }
      }
      neighbor "192.0.2.2" {
        group "iBGP-VPN"
      }
    }
  }
}

```

When the prefix 10.23.0.1/32 is advertised by PE-3, the route table for VPRN 2 on PE-2 is as follows:

```
[/]
A:admin@PE-2# show router 2 route-table

=====
Route Table (Service: 2)
=====
Dest Prefix[Flags]                                Type   Proto   Age           Pref
  Next Hop[Interface Name]                        Metric
-----
10.22.0.1/32                                       Local  Local   00h19m19s    0
  lo1                                               0
10.23.0.1/32                                       Remote BGP VPN 00h17m59s   170
  192.0.2.3 (tunneled:SR-ISIS:524290)             10
-----
No. of Routes: 2
---snip---
=====

```

RR-1 advertises the following BGP route for prefix 10.24.0.0/16 with next-hop 10.23.0.1 and community "target:64500:2":

```
[/]
A:admin@PE-2# show router bgp routes community target:64500:2

=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete

=====
BGP IPv4 Routes
=====
Flag Network                                LocalPref  MED
  Nexthop (Router)                          Path-Id    IGP Cost

```

	As-Path		Label
i	10.24.0.0/16	100	None
	10.23.0.1	None	0
	No As-Path		-

Routes : 1			
=====			

This route is not resolved in BGP, as indicated by the flags:

```
[/]
A:admin@PE-2# show router bgp routes community target:64500:2 hunt | match Flags
Flags          : Invalid IGP NextHop-Unresolved Leakable
```

The route is leakable and, by configuration, routes with unresolved next-hop can be leaked. The following **leak-import** policy is configured on PE-2 to import routes with community "target:64500:2":

```
# on PE-2:
configure {
  policy-options {
    community "target:64500:2" {
      member "target:64500:2" { }
    }
    policy-statement "leak-import-2" {
      entry 10 {
        from {
          community {
            name "target:64500:2"
          }
        }
        action {
          action-type accept
        }
      }
      default-action {
        action-type reject
      }
    }
  }
  service {
    vprn "VPRN 2" {
      admin-state enable
      service-id 2
      customer "1"
      autonomous-system 64500
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "64500:2"
          vrf-target {
            community "target:64500:2"
          }
        }
        auto-bind-tunnel {
          resolution filter
          resolution-filter {
            sr-isis true
          }
        }
      }
    }
  }
  bgp {
```

```

        rib-management {
            ipv4 {
                leak-import {
                    policy ["leak-import-2"]
                }
            }
        }
    }
}

```

The route is now leaked even though the next-hop is not only unresolved in the base router, but also unresolved in VPRN 2:

```

[/]
A:admin@PE-2# show router 2 bgp routes ipv4 leaked
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
li    10.24.0.0/16           100        None
      10.23.0.1 (Base)      None        0
      No As-Path            -
-----
Routes : 1
=====

```

```

[/]
A:admin@PE-2# show router 2 bgp routes hunt | match Flags
Flags          : Invalid IGP NextHop-Unresolved Leaked

```

By default, the BGP next-hop in the VPRN is resolved using IGP or static routes, but in this example, the route for 10.23.0.1/23 is resolved using the BGP VPN-IPv4 address family. Therefore, the **BGP next-hop resolution** context in VPRN 2 must be configured to allow the use of BGP routes:

```

# on PE-2:
configure {
    service {
        vprn "VPRN 2" {
            admin-state enable
            service-id 2
            customer "1"
            autonomous-system 64500
            bgp-ipvpn {
                mpls {
                    admin-state enable
                    route-distinguisher "64500:2"
                    vrf-target {
                        community "target:64500:2"
                    }
                }
            }
        }
    }
}

```



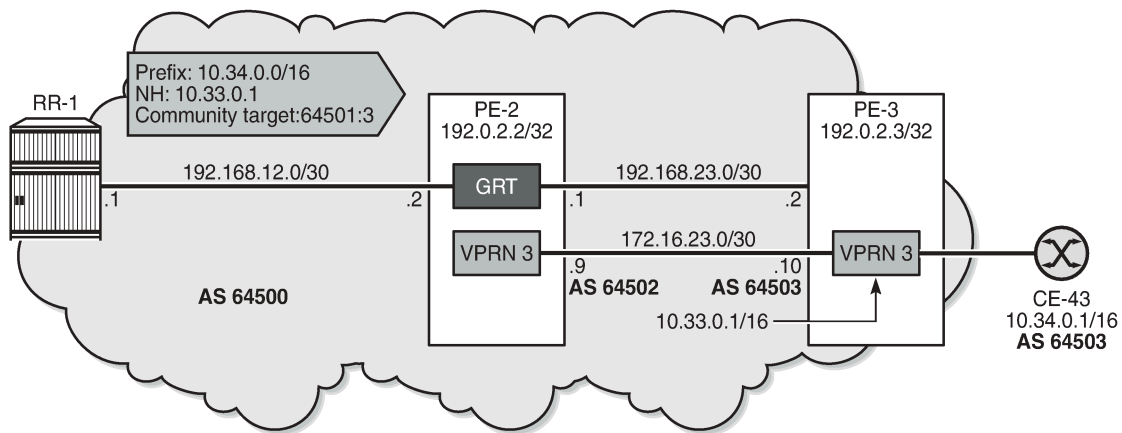
```

=====
Route Table (Service: 2)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]
Type Proto Age Pref
Metric
-----
10.22.0.1/32
lo1 Local Local 00h25m07s 0
0
10.23.0.1/32
192.0.2.3 (tunneled:SR-ISIS:524290) Remote BGP VPN 00h23m46s 170
10
10.24.0.0/16
192.0.2.3 (tunneled:SR-ISIS:524290) Remote BGP 00h00m22s 170
10
-----
No. of Routes: 3
---snip---
=====
    
```

Use case 3: BGP route leaked to VPRN 3 with next-hop resolved using eBGP

Figure 132: Leaked route 10.34.0.0/16 with next-hop resolved in VPRN 2 using eBGP shows that RR-1 advertises prefix 10.34.0.0/16 with next-hop 10.33.0.1. A BGP session is established within VPRN 3 on PE-2 and PE-3.

Figure 132: Leaked route 10.34.0.0/16 with next-hop resolved in VPRN 2 using eBGP



35964

On PE-3, VPRN 3 has a loopback Interface "lo1" configured with IP address 10.33.0.1/32, which is the BGP next-hop of the leakable route received from RR-1. Prefix 10.33.0.0/16 is advertised by BGP in VPRN 3.

```

# on PE-3:
configure {
  policy-options {
    prefix-list "10.33.0.0/16" {
      prefix 10.33.0.0/16 type longer {
      }
    }
  }
  policy-statement "export_10.33" {
    entry 10 {
      from {
        prefix-list ["10.33.0.0/16"]
      }
    }
  }
}
    
```



```
    }
    to {
        protocol {
            name [bgp]
        }
    }
    action {
        action-type accept
    }
}
}
}
service {
    vprn "VPRN 3" {
        admin-state enable
        service-id 3
        customer "1"
        autonomous-system 64503
        bgp-ipvpn {
            mpls {
                admin-state enable
                route-distinguisher "64503:3"
                vrf-target {
                    community "target:64500:3"
                }
            }
        }
    }
    bgp {
        router-id 10.33.0.1
        split-horizon true
        group "eBGP" {
            peer-as 64502
        }
        neighbor "172.16.23.9" {
            group "eBGP"
            export {
                policy ["export_10.33"]
            }
        }
    }
}
interface "int-VPRN3-PE-3-CE-43" {
    ipv4 {
        primary {
            address 172.16.34.9
            prefix-length 30
        }
    }
    sap 1/1/c1/1:3 {
    }
}
interface "int-VPRN3-PE-3-PE-2" {
    ipv4 {
        primary {
            address 172.16.23.10
            prefix-length 30
        }
    }
    sap 1/1/c1/2:3 {
    }
}
interface "lo1" {
    loopback true
    ipv4 {
        primary {
```


This route is leakable, but the next-hop 10.33.0.1 cannot be resolved in the base router of PE-2:

```
[/]
A:admin@PE-2# show router bgp routes community target:64500:3 hunt | match Flags
Flags          : Invalid IGP Nexthop-Unresolved Leakable
```

The only BGP route used in VPRN 3 on PE-2 is for prefix 10.33.0.1/32:

```
[/]
A:admin@PE-2# show router 3 bgp routes
=====
BGP Router ID:10.32.0.1      AS:64502      Local AS:64502
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i 10.33.0.1/32            None       None
      172.16.23.10         None       0
      64503                 -
-----
Routes : 1
=====
```

The following **leak-import** policy is configured on PE-2 to import leakable BGP routes with community "64500:3":

```
# on PE-2:
configure {
  policy-options {
    community "target:64500:3" {
      member "target:64500:3" { }
    }
    policy-statement "leak-import-3" {
      entry 10 {
        from {
          community {
            name "target:64500:3"
          }
        }
        action {
          action-type accept
        }
      }
      default-action {
        action-type reject
      }
    }
  }
}
```

This **leak-import** policy is applied in VPRN 3 and the **BGP next-hop-resolution** is configured as **use-bgp-routes true**:

```
# on PE-2:
configure {
  service {
    vprn "VPRN 3" {
      admin-state enable
      service-id 3
      customer "1"
      autonomous-system 64502
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "64502:3"
          vrf-target {
            community "target:64500:3"
          }
        }
      }
    }
  }
  bgp {
    next-hop-resolution {
      use-bgp-routes true      # for BGP and BGP-VPN routes
    }
    rib-management {
      ipv4 {
        leak-import {
          policy ["leak-import-3"]
        }
      }
    }
  }
}
}
```

With this configuration, the received RR-1 route for prefix 10.34.0.0/16 is leaked to VPRN 3 and the next-hop is resolved using a BGP route. The BGP routes in VPRN 3 on PE-2 are the following:

```
[/]
A:admin@PE-2# show router 3 bgp routes
=====
BGP Router ID:10.32.0.1      AS:64502      Local AS:64502
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  10.33.0.1/32                             None       None
      172.16.23.10                          None       0
      64503                                  -
u*>li 10.34.0.0/16                          100        None
      10.33.0.1 (Base)                      None       0
      64503                                  -
```

```
-----  
Routes : 2  
=====
```

The route table for VPRN 3 on PE-2 now includes a route for prefix 10.34.0.0/16:

```
[/]  
A:admin@PE-2# show router 3 route-table  
  
=====
```

Dest Prefix[Flags] Next Hop[Interface Name]	Type	Proto	Age Metric	Pref
10.32.0.1/32 lo1	Local	Local	00h32m42s 0	0
10.33.0.1/32 172.16.23.10	Remote	BGP	00h31m38s 0	170
10.34.0.0/16 172.16.23.10	Remote	BGP	00h00m09s 0	170
172.16.23.8/30 int-VPRN3-PE-2-PE-3	Local	Local	00h32m42s 0	0

```
-----  
No. of Routes: 4  
---snip---  
=====
```

Conclusion

BGP routes can be leaked from the base router to a VPRN routing instance, even when the next-hop is unresolved in the base router. This feature reduces the number of BGP sessions toward an RR, because all VPRN-related routes can now be leaked from the base router using a single BGP session. The VPRNs distinguish the routes based on the community value.

BGP Weighted ECMP

This chapter provides information about BGP weighted ECMP.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

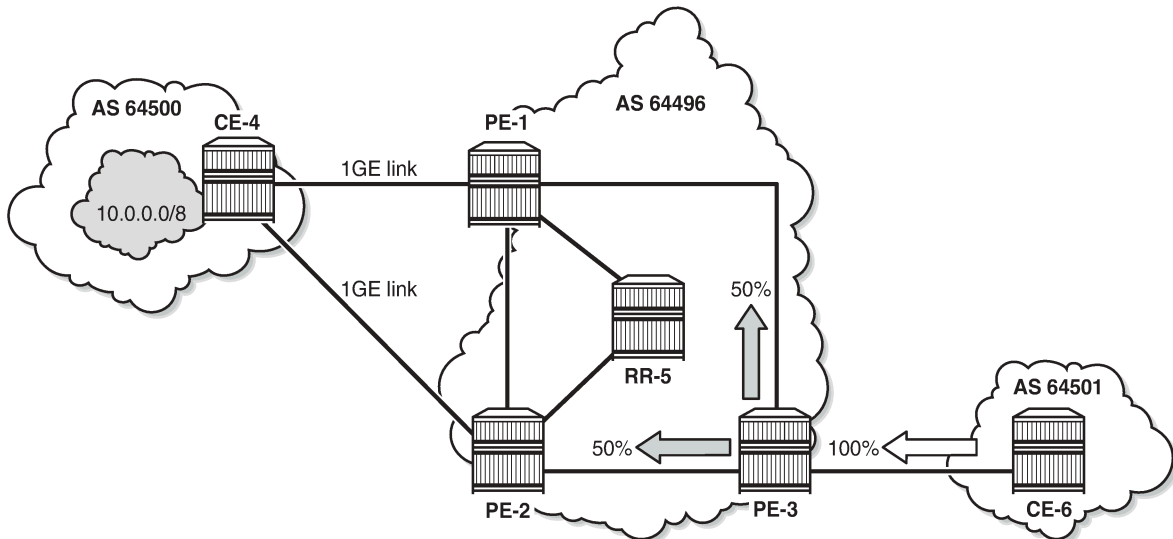
The information and configuration in this chapter was originally based on SR OS Release 15.0.R4. The CLI in the current edition is based on SR OS Release 23.3.R2.

Overview

Equal-cost multipath (ECMP) is a routing strategy that allows the installation of multiple next hops for an IP destination in the routing table. When used in conjunction with BGP multipath, the ingress router can forward traffic to an IP prefix destination in a load-balanced fashion across the available ECMP next hops. For more information about the implementation, see the [BGP Multipath](#) chapter.

In the standard implementation, ECMP distributes traffic as evenly as possible across all the ECMP next hops. [Figure 133: Standard ECMP - Equal Bandwidth Links](#) shows an example scenario where CE-4 is dual-homed to two PE routers and advertises the prefix 10.0.0.0/8. This prefix is then advertised within AS 64496 and received by PE-3, which in turn advertises it to CE-6 in AS 64501. PE-3 has BGP multipath and ECMP enabled, so the traffic toward destinations in 10.0.0.0/8 sent by CE-6 is load-balanced toward PE-1 and PE-2 as evenly as possible.

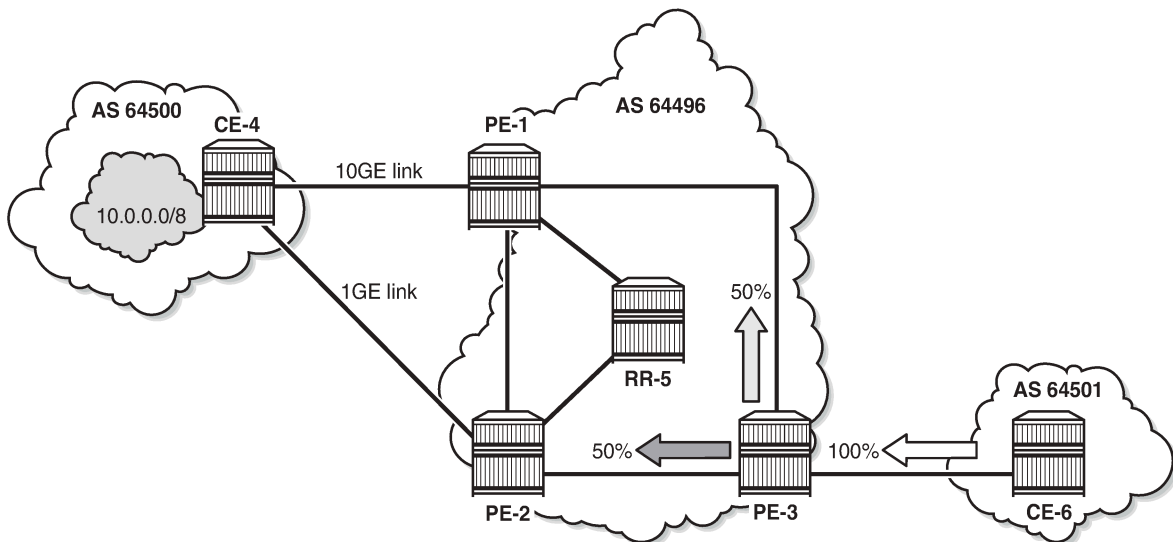
Figure 133: Standard ECMP - Equal Bandwidth Links



26854

The behavior of equally distributing across the ECMP next hops may not be suitable under specific circumstances. Consider the same topology with the connection between CE-4 and PE-1 replaced with a 10GE link, while the CE-4 to PE-2 connection still is a 1GE link, as shown in [Figure 134: Standard ECMP - Unequal Bandwidth Links](#). In standard ECMP operation, when PE-3 sends 50% of traffic to PE-1 and 50% to PE-2, this may result in an under-utilization of the link between CE-4 and PE-1 or an over-utilization of the link between CE-4 and PE-2.

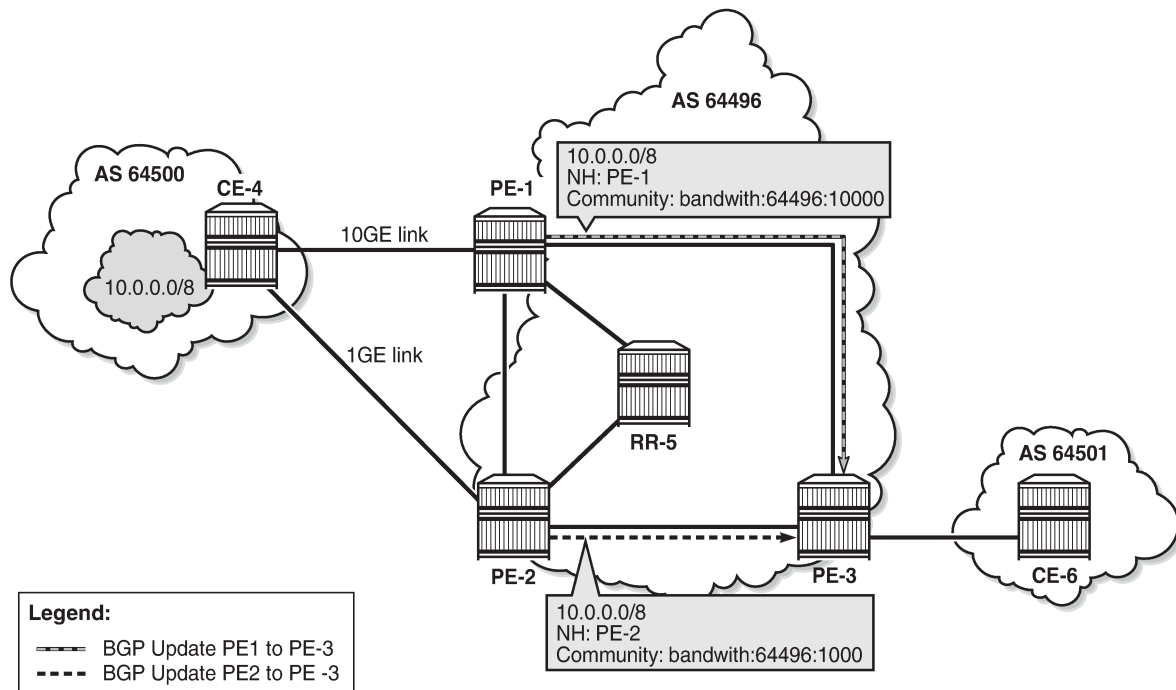
Figure 134: Standard ECMP - Unequal Bandwidth Links



26855

BGP Weighted ECMP, also known as Unequal-Cost Multipath (UCMP), allows for the distribution of traffic in proportion to the relative bandwidth of each equal-cost path. This feature uses a BGP community called the Link Bandwidth Extended Community. [Figure 135: Link Bandwidth Extended Community Advertisement](#) shows that PE-1 and PE-2, with this functionality, can add a Link Bandwidth Extended Community to the BGP routes advertised toward other routers within AS 64496 that indicates the bandwidth of their PE-CE link.

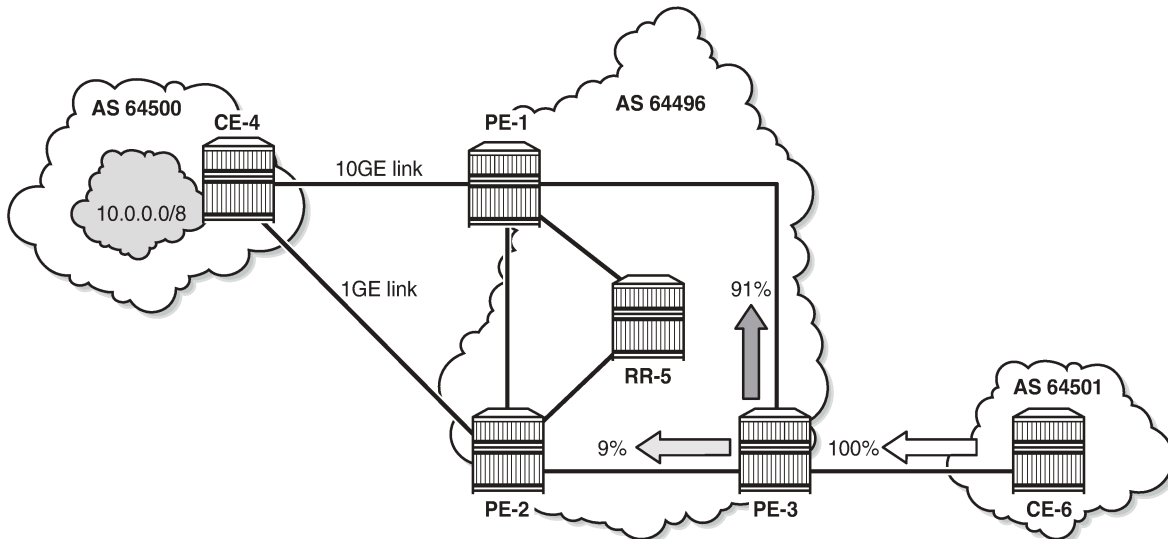
Figure 135: Link Bandwidth Extended Community Advertisement



26856

PE-3 can use the information in the Link Bandwidth Extended Community to distribute the traffic according to the relative bandwidth, or the "weight" of each path. [Figure 136: Weighted ECMP - Unequal Bandwidth Links](#) shows that 91% of traffic is sent toward PE-1 with the 10GE link and 9% is sent toward PE-2 with the 1GE link.

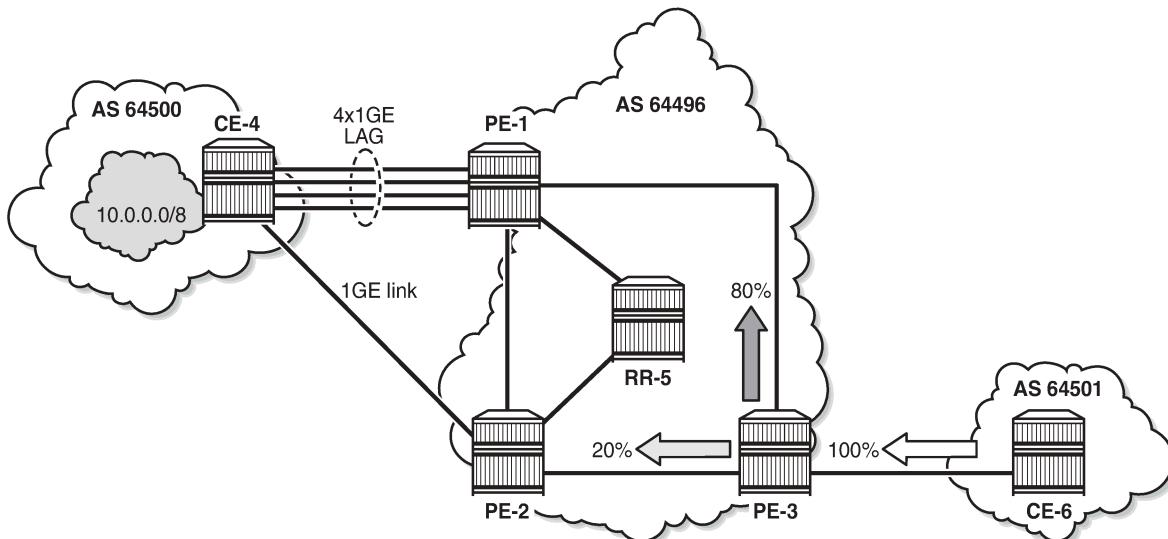
Figure 136: Weighted ECMP - Unequal Bandwidth Links



26857

Figure 137: Weighted ECMP - Link Aggregation Group shows another example where the CE-4-to-PE-1 link is composed of four 1GE links that are part of a Link Aggregation Group (LAG) and the CE-4-to-PE-2 link is 1GE. Weighted ECMP can be used here to achieve an 80% to 20% distribution of traffic sent from PE-3 to PE-1 and PE-2, respectively.

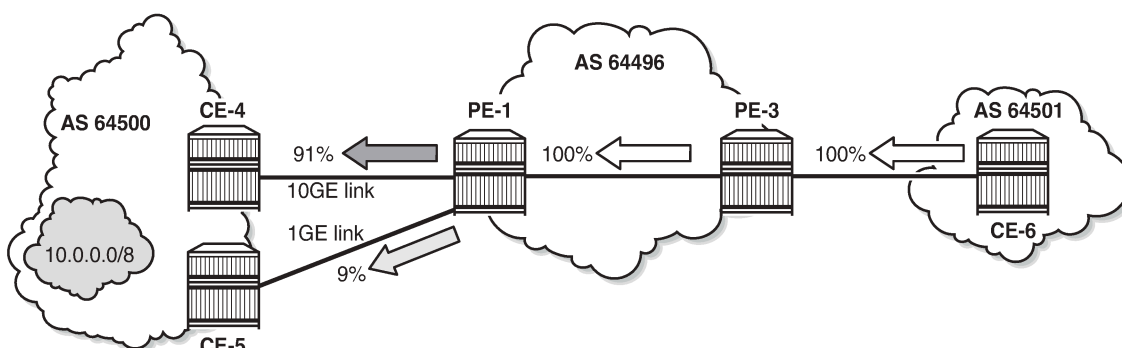
Figure 137: Weighted ECMP - Link Aggregation Group



26858

Figure 138: Standard ECMP - Unequal Bandwidth Links with eBGP shows an example where PE-1 is connected to two eBGP routers in neighbor AS 64500. Using the weighted ECMP functionality, 91% of traffic is sent to CE-4 and 9% to CE-5, according to the relative bandwidth values.

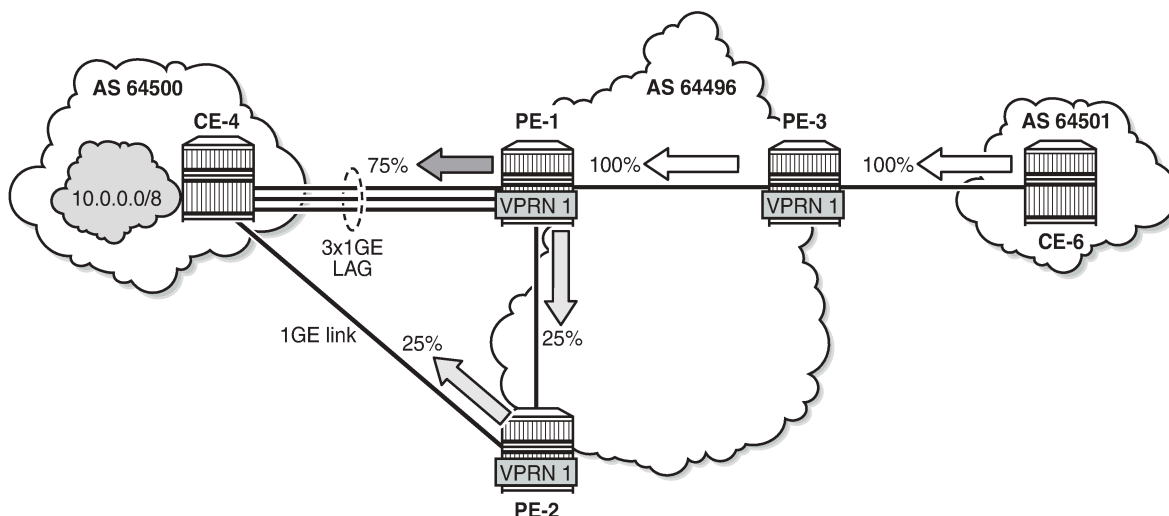
Figure 138: Standard ECMP - Unequal Bandwidth Links with eBGP



26859

Figure 139: Weighted ECMP - Unequal Bandwidth Links with VPRN shows an example with a Layer 3 VPRN service. PE-1 receives prefix 10.0.0.0/8 from CE-4 via eBGP, and also from PE-2 via iBGP. PE-1 sets the Link Bandwidth Extended Community indicating 3GE on the route received from CE-4. PE-2 sets the community value indicating 1GE on the route it advertises to PE-1. With Exterior Interior Border Gateway Protocol (EIBGP) multipath (described in the [BGP Multipath](#) chapter) and ECMP within the VPRN, PE-1 can send 75% of traffic on the direct LAG link to CE-4 and 25% to PE-2, which then forwards that traffic to CE-4.

Figure 139: Weighted ECMP - Unequal Bandwidth Links with VPRN



26860

Link Bandwidth Extended Community is defined in *draft-ietf-idr-link-bandwidth-06* and has the following characteristics:

- Signals the link bandwidth of a BGP path
- Has the following format: bandwidth:<as-number>:<value>
 - bandwidth is the community type
 - <as-number> is the local AS number

- <value> is a fixed/static bandwidth in Mb/s (converted to IEEE floating point format in a BGP Update message)
- Optional and non-transitive attribute (not sent to other eBGP peers upon receipt)
- If a router changes the route next hop, it does not propagate the Link Bandwidth Extended Community
- A route can only have a single Link Bandwidth Extended Community
- SR OS routers automatically perform weighted load balancing if all the BGP updates received for a destination contain the Link Bandwidth Extended Community

Link Bandwidth Extended Community can be added to a BGP route with the following methods:

- **link-bandwidth** command
- BGP import policy action
- VRF import policy action
- BGP export policy action

The **link-bandwidth** command has the following characteristics:

- Configurable per BGP group or neighbor in base router or VPRN
- Adds a Link Bandwidth Extended Community to all (IPv4, IPv6, VPN-IPv4, VPN-IPv6, label-IPv4, label-IPv6) routes received from directly connected EBGP peers
- Bandwidth value is based on the speed of port or active LAG members
- Bandwidth is automatically adjusted for LAG interfaces based on the number of active LAG member ports

SR OS uses the following rules when BGP paths are received with Link Bandwidth Extended Communities:

1. If BGP multipath and ECMP are configured and all the eligible multipaths have a Link Bandwidth Extended Community, then weighted ECMP is performed on the relative bandwidth of each path.
2. If EIBGP multipath and ECMP are enabled in a VPRN and all the eligible next hops have a Link Bandwidth Extended Community, then weighted ECMP is performed based on the relative bandwidth of each path.
3. The Link Bandwidth Extended Community is not used as a criterion for two or more paths to be considered equal for BGP/EIBGP multipath purposes.

Configuration

The following configuration examples for BGP weighted ECMP are covered in this chapter:

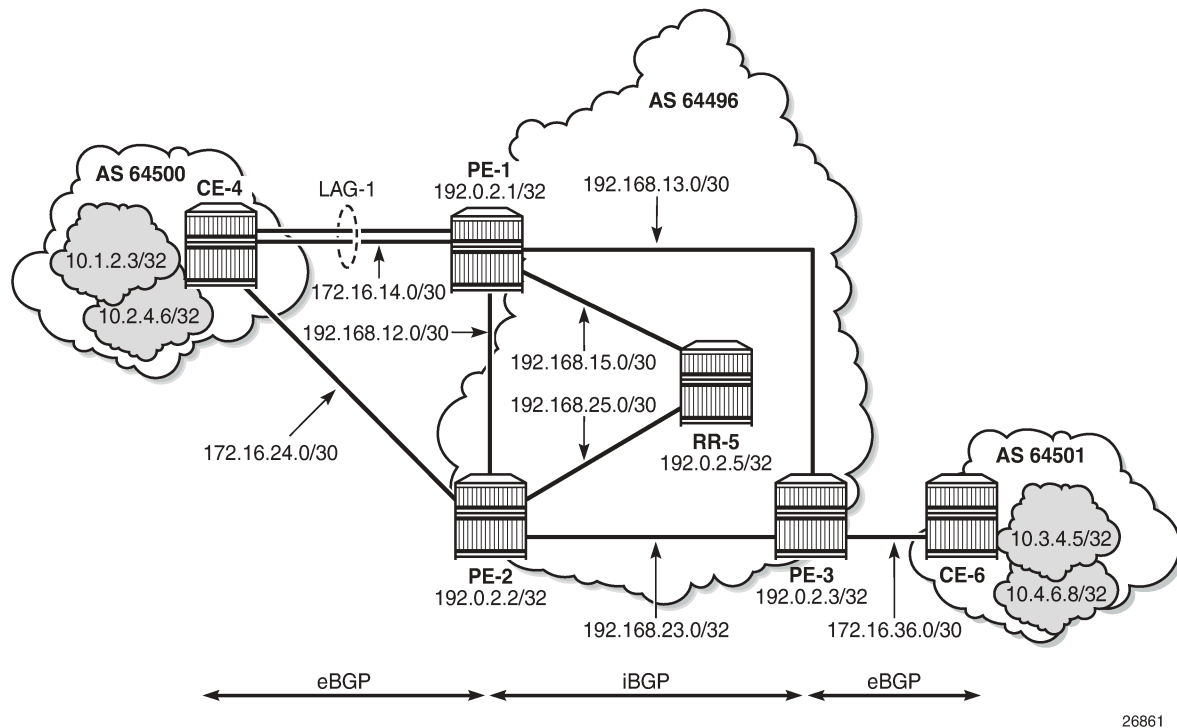
- [BGP Weighted ECMP for IPv4 Family using the link-bandwidth command](#)
- [BGP Weighted ECMP for IPv4 Family using BGP Import Policy](#)

[Figure 140: Example Topology - BGP Weighted ECMP for IPv4 Family](#) shows the example topology for BGP Weighted ECMP for IPv4 family with the following characteristics:

- CE-4 in AS 64500 advertises both prefixes 10.1.2.3/32 and 10.2.4.6/32 to its eBGP peers PE-1 and PE-2 in AS 64496.
- RR-5 is route reflector for all PEs in AS 64496.
- **add-paths** is configured on all PE routers and RR-5 with a **send** limit of 2.

- CE-6 in AS 64501 advertises both prefixes 10.3.4.5/32 and 10.4.6.8/32 to its eBGP peer PE-3 in AS 64496.

Figure 140: Example Topology - BGP Weighted ECMP for IPv4 Family



26861

Initial Configuration

The initial configuration on all nodes includes:

- Cards, MDAs, ports
- LAG configured for the link between CE-4 and PE-1 with two member links
- Router interfaces
- IS-IS as IGP on all interfaces within AS 64496 (alternatively, OSPF can be used)

BGP is configured on all the nodes. CE-4 peers with PE-1 and PE-2 and exports the 10.1.2.3/32 and 10.2.4.6/32 loopback prefixes to both eBGP peers, as follows:

```
# on CE-4:
configure {
  router "Base" {
    interface "int-loopback-1" {
      loopback
      ipv4 {
        primary {
          address 10.1.2.3
          prefix-length 32
        }
      }
    }
  }
}
```

```

}
interface "int-loopback-2" {
  loopback
  ipv4 {
    primary {
      address 10.2.4.6
      prefix-length 32
    }
  }
}
autonomous-system 64500
bgp {
  admin-state enable
  rapid-withdrawal true
  split-horizon true
  ebgp-default-reject-policy {
    import false
    export false
  }
  group "eBGP" {
    peer-as 64496
    export {
      policy ["policy-export-bgp"]
    }
  }
  neighbor "172.16.14.1" {
    group "eBGP"
  }
  neighbor "172.16.24.1" {
    group "eBGP"
  }
}
}
policy-options {
  prefix-list "10.0.0.0/8" {
    prefix 10.0.0.0/8 type longer {
    }
  }
  policy-statement "policy-export-bgp" {
    entry 10 {
      from {
        prefix-list ["10.0.0.0/8"]
      }
      action {
        action-type accept
      }
    }
  }
}
}
}

```

The BGP configuration on CE-6 is identical, except for the loopback interface addresses.

PE-1 peers with CE-4 in AS 65400 and RR-5 in AS 64496. **add-paths** is enabled on the iBGP group to advertise redundant BGP paths to the route reflector. The BGP configuration on PE-1 is as follows:

```

# on PE-1:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      admin-state enable
      rapid-withdrawal true
    }
  }
}

```

```

split-horizon true
ebgp-default-reject-policy {
    import false
    export false
}
group "eBGP" {
    peer-as 64500
}
neighbor "172.16.14.2" {
    group "eBGP"
}
group "iBGP" {
    family {
        ipv4 true
    }
    next-hop-self true
    peer-as 64496
    add-paths {
        ipv4 {
            send 2
            receive true
        }
    }
}
neighbor "192.0.2.5" {
    group "iBGP"
}
}
exit all

```

The BGP configuration on PE-2 and PE-3 is similar to that on PE-1.

RR-5 acts as a route reflector to all the PEs in AS 64496 with a cluster ID of 5.5.5.5. **add-paths** is enabled to advertise redundant BGP paths to the PEs. The configuration on RR-5 is as follows:

```

# on RR-5:
configure {
    router "Base" {
        autonomous-system 64496
        bgp {
            admin-state enable
            rapid-withdrawal true
            split-horizon true
            ebgp-default-reject-policy {
                import false
                export false
            }
            group "iBGP" {
                family {
                    ipv4 true
                }
                cluster {
                    cluster-id 5.5.5.5
                }
                peer-as 64496
                add-paths {
                    ipv4 {
                        send 2
                        receive true
                    }
                }
            }
        }
        neighbor "192.0.2.1" {

```

```

        group "iBGP"
        }
        neighbor "192.0.2.2" {
            group "iBGP"
        }
        neighbor "192.0.2.3" {
            group "iBGP"
        }
    }
}

```

BGP Weighted ECMP for IPv4 Family using the link-bandwidth command

PE-3 receives the prefixes 10.1.2.3/32 and 10.2.4.6/32 from PE-1 and PE-2 via the route reflector and indicates the ones received from PE-1 as the "used" or active routes, as follows:

```

[/]
A:admin@PE-3# show router bgp routes
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
u*>i  10.1.2.3/32              100        None
      192.0.2.1              7          10
      64500                   -
*i    10.1.2.3/32              100        None
      192.0.2.2              8          10
      64500                   -
u*>i  10.2.4.6/32              100        None
      192.0.2.1              9          10
      64500                   -
*i    10.2.4.6/32              100        None
      192.0.2.2              10         10
      64500                   -
u*>i  10.3.4.5/32              None       None
      172.16.36.2             None       0
      64501                   -
u*>i  10.4.6.8/32              None       None
      172.16.36.2             None       0
      64501                   -
-----
Routes : 6
=====

```

ECMP and BGP multipath are enabled on PE-3 with the following commands:

```

# on PE-3:
configure {

```

```

router "Base" {
  ecmp 2
  bgp {
    multipath {
      max-paths 2
    }
  }
}

```

As a result, PE-3 installs the routes from PE-2 as active, in addition to those from PE-1:

```

[/]
A:admin@PE-3# show router bgp routes
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref  MED
  NextHop (Router)                          Path-Id    IGP Cost
  As-Path                                     -          Label
-----
u*>i 10.1.2.3/32                             100        None
      192.0.2.1                              7          10
      64500                                   -          -
u*>i 10.1.2.3/32                             100        None
      192.0.2.2                              8          10
      64500                                   -          -
u*>i 10.2.4.6/32                             100        None
      192.0.2.1                              9          10
      64500                                   -          -
u*>i 10.2.4.6/32                             100        None
      192.0.2.2                              10         10
      64500                                   -          -
u*>i 10.3.4.5/32                             None       None
      172.16.36.2                            None       0
      64501                                   -          -
u*>i 10.4.6.8/32                             None       None
      172.16.36.2                            None       0
      64501                                   -          -
-----
Routes : 6
=====

```

The multiple next hops are also visible in the route table of PE-3:

```

[/]
A:admin@PE-3# show router route-table protocol bgp
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type  Proto  Age           Pref
  Next Hop[Interface Name]                       -     -      Metric
-----
10.1.2.3/32                                       Remote BGP    00h00m55s  170

```



```

192.168.13.1                               10
10.1.2.3/32                               Remote BGP 00h00m55s 170
192.168.23.1                               10
10.2.4.6/32                               Remote BGP 00h00m55s 170
192.168.13.1                               10
10.2.4.6/32                               Remote BGP 00h00m55s 170
192.168.23.1                               10
10.3.4.5/32                               Remote BGP 00h02m47s 170
172.16.36.2                               0
10.4.6.8/32                               Remote BGP 00h02m47s 170
172.16.36.2                               0
-----
No. of Routes: 6
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

The following command shows that the routes received on PE-3 have no community added (string within single quotes in the match statement to indicate that it is an expression).

```

[/]
A:admin@PE-3# show router bgp routes 10.1.2.3/32 hunt brief | match '^Nexthop|Community'
Nexthop      : 192.0.2.1
Community    : No Community Members
Nexthop      : 192.0.2.2
Community    : No Community Members

```

The following command output shows the ECMP-weight outputs assigned to next hops 192.0.2.1 and 192.0.2.2. Both have a value of 1.

```

[/]
A:admin@PE-3# show router fib 1 ip-prefix-prefix-length 10.1.2.3/32 extensive
=====
FIB Display (Router: Base)
=====
Dest Prefix      : 10.1.2.3/32
Protocol         : BGP
Installed        : Y
Indirect Next-Hop : 192.0.2.1
  QoS            : Priority=n/c, FC=n/c
  Source-Class   : 0
  Dest-Class     : 0
  ECMP-Weight   : 1
  Resolving Next-Hop : 192.168.13.1
    Interface     : int-PE-3-PE-1
    ECMP-Weight   : 1
  Indirect Next-Hop : 192.0.2.2
  QoS            : Priority=n/c, FC=n/c
  Source-Class   : 0
  Dest-Class     : 0
  ECMP-Weight   : 1
  Resolving Next-Hop : 192.168.23.1
    Interface     : int-PE-3-PE-2
    ECMP-Weight   : 1
=====
Total Entries : 1
=====

```

The following command is executed on both PE-1 and PE-2 to automatically add a Link Bandwidth Extended Community on routes received from their eBGP neighbor CE-4:

```
# on PE-1 and on PE-2:
configure {
  router "Base" {
    bgp {
      group "eBGP" {
        link-bandwidth {
          add-to-received-ebgp ipv4
        }
      }
    }
  }
}
exit all
```

PE-3 now receives the routes from PE-1 and PE-2 with Link Bandwidth Extended Communities corresponding to the interface bandwidth for each CE-PE link:

```
[/]
A:admin@PE-3# show router bgp routes 10.1.2.3/32 hunt brief | match '^NextHop|Community'
NextHop      : 192.0.2.1
Community    : bandwidth:64496:200000
NextHop      : 192.0.2.2
Community    : bandwidth:64496:100000
```

The following command output now shows that the ECMP-Weight value assigned to next hop 192.0.2.1 is 2, relative to its two member interfaces in the LAG, whereas the ECMP-Weight value of 192.0.2.2 is still 1, because it has a single interface to CE-4:

```
[/]
A:admin@PE-3# show router fib 1 ip-prefix-prefix-length 10.1.2.3/32 extensive

=====
FIB Display (Router: Base)
=====
Dest Prefix      : 10.1.2.3/32
Protocol         : BGP
Installed        : Y
Indirect Next-Hop : 192.0.2.1
  QoS            : Priority=n/c, FC=n/c
  Source-Class   : 0
  Dest-Class     : 0
  ECMP-Weight   : 2
  Resolving Next-Hop : 192.168.13.1
    Interface     : int-PE-3-PE-1
    ECMP-Weight   : 1
  Indirect Next-Hop : 192.0.2.2
  QoS            : Priority=n/c, FC=n/c
  Source-Class   : 0
  Dest-Class     : 0
  ECMP-Weight   : 1
  Resolving Next-Hop : 192.168.23.1
    Interface     : int-PE-3-PE-2
    ECMP-Weight   : 1
=====
Total Entries : 1
=====
```

If a tester tool is available, it can be used to test the traffic load-balancing behavior by using it to replace CE-4 and CE-6 in the topology. This would be the preferred option to get better results in observing the effect of weighted ECMP. Multiple flows (preferably a couple of hundred or thousands) should be created and sent between the tester ports. For a simple test, the SR OS rapid ping tool can be used to create traffic between the loopback interfaces of CE-6 and CE-4.

At least three flows need to be created to see traffic distributed over the two LAG links between CE-4 and PE-1 and the single link between CE-4 and PE-2. The loopback IP addresses on CE-4 and CE-6 have been specifically chosen to demonstrate the expected load balancing. The behavior may be different if different loopback IP addresses are used, because it affects the load-balancing algorithm.

To facilitate the test, two more Telnet or SSH sessions are initiated to CE-6 (three in total) and the following commands are executed in each separate session:

First session:

```
[/]
A:admin@CE-6# ping 10.1.2.3 source-address 10.3.4.5 size 1200 count 100000 interval 1
```

Second session:

```
[/]
A:admin@CE-6# ping 10.1.2.3 source-address 10.3.4.5 size 1200 count 100000 interval 1
```

Third session:

```
[/]
A:admin@CE-6# ping 10.1.2.3 source-address 10.4.6.8 size 1200 count 100000 interval 1
```

The **monitor** command outputs on PE-1 and PE-2 show the traffic from CE-6 to CE-4 is being distributed over the two LAG links on PE-1 and the single link on PE-2. In the ideal case, PE-1 would receive 67% and PE-2 would receive 33% of total traffic; however, it may not be possible to observe this effectively with only three ICMP flows.

On the PE-1 LAG link to CE-4, the following traffic is monitored. In each interval of 3 seconds, the number of output bytes is 250000 (or more if other traffic is sent in parallel).

```
[/]
A:admin@PE-1# monitor lag 1 interval 3 repeat 999 rate

=====
Monitor statistics for LAG ID 1
=====
Port-id          Input packets      Output packets
                Input bytes        Output bytes
                Input errors [Input util %]  Output errors [Output util %]
-----snip-----
At time t = 6 sec (Mode: Rate)
-----
1/1/c2/1         4                  3
                3878              2628
                0                  ~0.00 0           ~0.00
1/1/c5/1         1                  1
                128               128
                0                  ~0.00 0           ~0.00
-----
Totals           5                  4
                4006             2756
                0                  ~0.00 0           ~0.00
-----
At time t = 9 sec (Mode: Rate)
-----
1/1/c2/1         4                  3
```

	3878		2628	
1/1/c5/1	0	~0.00	0	~0.00
	1		1	
	128		128	
	0	~0.00	0	~0.00

Totals	5		4	
	4006		2756	
	0	~0.00	0	~0.00

On the PE-2 to CE-4 link, the following traffic is monitored. In each interval of 3 seconds, the number of output bytes is 125000 (or more if other traffic is sent in parallel):

```
[/]
A:admin@PE-2# monitor port 1/1/c1/1 interval 3 repeat 999 rate

=====
Monitor statistics for Port 1/1/c1/1
=====
                                Input          Output
-----
---snip---
-----
At time t = 6 sec (Mode: Rate)
-----
Octets                0                1250
Packets               0                 1
Errors                0                 0
Bits                  0                10000
Utilization (% of port capacity) 0.00           ~0.00
-----
At time t = 9 sec (Mode: Rate)
-----
Octets                0                1250
Packets               0                 1
Errors                0                 0
Bits                  0                10000
Utilization (% of port capacity) 0.00           ~0.00
```

BGP Weighted ECMP for IPv4 Family using BGP Import Policy

The **link-bandwidth** command, which was enabled in the previous step, is removed on PE-1 and PE-2:

```
# on PE-1 and on PE-2:
configure {
  router "Base" {
    bgp {
      group "eBGP" {
        link-bandwidth {
          delete add-to-received-ebgp
        }
      }
    }
  }
  exit all
}
```

The following policy is configured on PE-1 to manually add the Link Bandwidth Extended Community "bandwidth:64500:4000" to routes received from CE-4:

```
# on PE-1:
configure {
  policy-options {
```


The policy is applied on PE-2 for the eBGP group in the import direction:

```
# on PE-2:
configure {
  router "Base" {
    bgp {
      group "eBGP" {
        import {
          policy ["policy-import-bandwidth-2G"]
        }
      }
    }
  }
}
```

PE-3 receives the routes from PE-1 and PE-2 with Link Bandwidth Extended Communities as configured in the previous step:

```
[/]
A:admin@PE-3# show router bgp routes 10.1.2.3/32 hunt brief | match '^NextHop|Community'
NextHop      : 192.0.2.1
Community    : bandwidth:64500:4000
NextHop      : 192.0.2.2
Community    : bandwidth:64500:2000
```

Again, the following command output shows that the ECMP-weight output assigned to next hop 192.0.2.1 has become 2:

```
[/]
A:admin@PE-3# show router fib 1 ip-prefix-prefix-length 10.1.2.3/32 extensive

=====
FIB Display (Router: Base)
=====
Dest Prefix      : 10.1.2.3/32
Protocol         : BGP
Installed        : Y
Indirect Next-Hop : 192.0.2.1
  QoS             : Priority=n/c, FC=n/c
  Source-Class    : 0
  Dest-Class      : 0
  ECMP-Weight    : 2
  Resolving Next-Hop : 192.168.13.1
    Interface      : int-PE-3-PE-1
    ECMP-Weight    : 1
  Indirect Next-Hop : 192.0.2.2
  QoS             : Priority=n/c, FC=n/c
  Source-Class    : 0
  Dest-Class      : 0
  ECMP-Weight    : 1
  Resolving Next-Hop : 192.168.23.1
    Interface      : int-PE-3-PE-2
    ECMP-Weight    : 1
=====
Total Entries : 1
=====
```



Note:

Any dynamic changes to the Link Bandwidth Extended Community upon failure or bandwidth change of a LAG link are not possible with the policy functionality, as opposed to using the **link-bandwidth** command.

Similar tests can be run using the rapid ping facility or an external tester tool as described in the previous section to check the packet forwarding behavior.

Conclusion

BGP Weighted ECMP allows modification of the standard load-balancing behavior to accommodate the relative link bandwidth values of different BGP next hops. This allows better utilization of the links in the network with different capacities. The bandwidth values are advertised by edge routers and carried within a BGP community called the Link Bandwidth Extended Community. SR OS routers automatically perform load balancing if all the BGP routes to a destination contain this community.

Dynamic BGP Peers

This chapter provides information about dynamic BGP peers.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 14.0.R7, but the MD-CLI in the current edition corresponds to SR OS Release 20.7.R1.

Overview

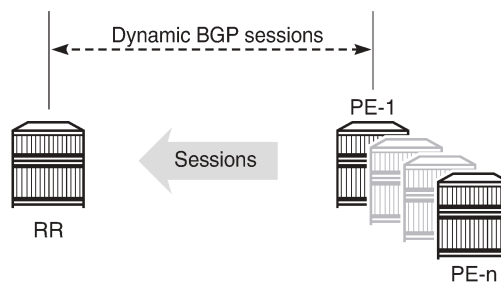
SR OS supports static and dynamic BGP sessions, where the static sessions are initiated toward explicitly configured non-passive neighbors, which are identified through an IPv4 or IPv6 address.

Neighbors must be part of a BGP peer group, and all neighbors in the same group share the same characteristics unless more specific characteristics are defined at the neighbor level.

SR OS will initiate TCP sessions toward explicitly configured non-passive neighbors, and listen for incoming TCP connections on port 179 for these configured neighbors. Sessions established with explicitly configured neighbors are considered static BGP sessions.

Dynamic BGP sessions can be established without explicitly configured neighbors; see [Figure 141: Establishing dynamic BGP sessions](#). The source address of a dynamic peer should match one of the configured IPv4 or IPv6 prefixes for the allowed peer Autonomous Systems (ASs). SR OS will only listen for incoming TCP connections on port 179 for these prefixes (which defines passive mode). SR OS will never initiate connections toward dynamic peers. This is consistent with RFC 4271, which allows a BGP speaker to accept connections from unconfigured BGP peers.

Figure 141: Establishing dynamic BGP sessions



26360

Dynamic BGP peering is also supported for ESM-routed subscriber hosts to improve deployment flexibility, but this is out of the scope of this chapter.

Characteristics

In SR OS, BGP groups and dynamic BGP peers have the following characteristics:

- A BGP group can support static and dynamic peers simultaneously.
- To support dynamic, unconfigured peers, multiple prefixes (IPv4/IPv6) in multiple allowed peer ASs can be associated with a group.
- A dynamic peer will be associated with a group, based on the source IP address of an incoming TCP connection. If multiple overlapping prefixes match, the prefix with the longest prefix length is used.
- A maximum number of dynamic peers can be configured per group and for the entire BGP instance. Whenever an incoming connection for a new dynamic session would cause either a group limit or the overall BGP limit to be exceeded, the connection attempt is rejected with a BGP Notification message.
- Dynamic peers are supported in the base router as well as in VPRN BGP instances.

Behavior

When a dynamic session is established, the following behavior will be observed when changes are made:

- If a new **prefix** entry is added to a group and this entry will become the longest prefix match for the IP address, then the session remains up, without interruption, if the new entry belongs to the same group as the one previously used to set up the dynamic session.
- If a new **prefix** entry is added to a group and this entry becomes the longest prefix match for the IP address, then the session is torn down immediately if the new entry belongs to a different group from the one previously used to set up the dynamic session. When the remote end attempts to reestablish the session, the parameters used locally are inherited from the new group.
- If a **neighbor** command is added to any group and its IP address matches the source IP address of an established dynamic session, then the dynamic session is torn down and the new session that is established inherits its local parameters from the **neighbor** configuration.

Using dynamic BGP peers can reduce the configuration file size of an SR OS router considerably, and is mainly used on route reflectors.

Configuration

In this section, the following two examples are shown:

- Dynamic BGP peers on a route reflector in an AS
- Dynamic BGP peers in multiple ASs

Dynamic BGP peers on a route reflector in an AS

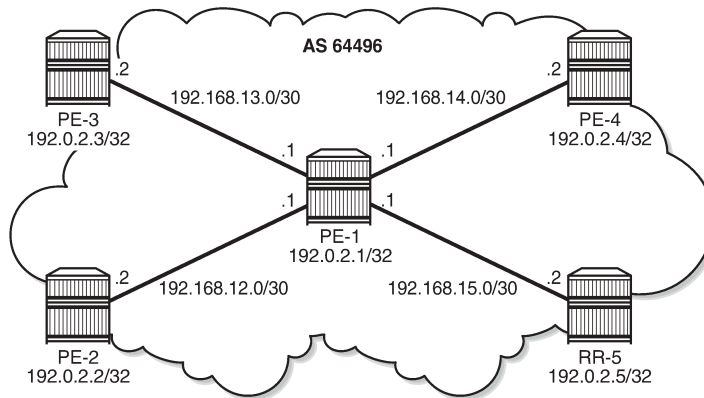
[Figure 142: Dynamic BGP peers](#) shows the example topology, and has the following characteristics:

- All nodes are part of AS 64496.
- BGP sessions are established between the routers of AS 64496, using RR-5 as route reflector with PE-1, PE-2, PE-3, and PE-4 being the route reflector clients.

The initial configuration on the nodes includes:

- cards, MDAs, and ports
- router interfaces
- IS-IS between the routers

Figure 142: Dynamic BGP peers



26361

BGP is configured between the route reflector clients and the route reflector for the IPv4 address family. The configuration on PE-1 is as follows:

```
# on PE-1:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      loop-detect discard-route
      split-horizon true
      group "iBGP" {
        peer-as 64496
      }
      neighbor "192.0.2.5" {
        group "iBGP"
      }
    }
  }
}
```

The BGP configuration on PE-2 is as follows. The BGP configuration on PE-3 and PE-4 is similar, but the prefix-lists are different.

```
# on PE-2:
configure {
  policy-options {
    prefix-list "local-lb" {
      prefix 172.31.2.0/24 type exact {
      }
    }
  }
  policy-statement "exp-local-lb" {
    entry 10 {
      from {
        prefix-list ["local-lb"]
      }
      action {
        action-type accept
      }
    }
  }
}
```

```

    }
  }
}
router "Base" {
  autonomous-system 64496
  bgp {
    loop-detect discard-route
    split-horizon true
    group "iBGP" {
      peer-as 64496
      export {
        policy ["exp-local-lb"]
      }
    }
  }
  neighbor "192.0.2.5" {
    group "iBGP"
  }
}

```

The initial route reflector RR-5 BGP configuration is as follows:

```

# on RR-5:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      loop-detect discard-route
      split-horizon true
      dynamic-neighbor-limit 20
      group "iBGP" {
        peer-as 64496
        dynamic-neighbor-limit 10
        cluster {
          cluster-id 5.5.5.5
        }
        dynamic-neighbor {
          match {
            prefix 192.0.2.0/24 {
              allowed-peer-as ["64496"]
            }
          }
        }
      }
    }
  }
}

```

Dynamic neighbors are shown with the "D" flag, as follows:

```

[]
A:admin@RR-5# show router bgp summary all

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====

Neighbor
Description
ServiceId          AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
                PktSent OutQ
-----
192.0.2.1(D)
Def. Instance 64496      64   0 00h30m53s 0/0/3 (IPv4)
                67   0
192.0.2.2(D)

```

```

Def. Instance 64496      66  0 00h31m11s 1/1/2 (IPv4)
                67  0
192.0.2.3(D)
Def. Instance 64496      67  0 00h31m49s 1/1/2 (IPv4)
                68  0
192.0.2.4(D)
Def. Instance 64496      65  0 00h30m47s 1/1/2 (IPv4)
                66  0
-----

```

The details for neighbor PE-2 show that the session is dynamic, as follows:

```

[]
A:admin@RR-5# show router bgp neighbor 192.0.2.2

=====
BGP Neighbor
=====
-----
Peer           : 192.0.2.2
Description    : (Not Specified)
Group          : iBGP
-----
Peer AS        : 64496           Peer Port      : 49704
Peer Address   : 192.0.2.2
Local AS       : 64496           Local Port     : 179
Local Address  : 192.0.2.5
Peer Type      : Internal       Dynamic Peer   : Yes
State         : Established     Last State     : Established
Last Event    : recv0pen
Last Error    : Cease (Connection Collision Resolution)
Local Family  : IPv4
Remote Family  : IPv4
Hold Time     : 90              Keep Alive     : 30
Min Hold Time : 0
Active Hold Time : 90          Active Keep Alive : 30
Cluster Id    : 5.5.5.5
---snip---

-----
Neighbors shown : 1
=====
* indicates that the corresponding row element may have been truncated.

```

The BGP configuration on route reflector RR-5 is modified with static BGP neighbor PE-1, as follows:

```

# on RR-5:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      loop-detect discard-route
      split-horizon true
      dynamic-neighbor-limit 20
      group "iBGP" {
        peer-as 64496
        dynamic-neighbor-limit 10
        cluster {
          cluster-id 5.5.5.5
        }
      }
      dynamic-neighbor {
        match {

```

```

        prefix 192.0.2.0/24 {
            allowed-peer-as ["64496"]
        }
    }
}
neighbor "192.0.2.1" {      # defines PE-1 as a static neighbor
    group "iBGP"
    keepalive 20
    hold-time {
        seconds 60
    }
}

```

Therefore, the properties of BGP group iBGP are as follows:

```

[]
A:admin@RR-5# show router bgp group "iBGP"
=====
BGP Group : iBGP
=====
Group           : iBGP
Description     : (Not Specified)
Group Type      : No Type           State           : Up
Peer AS        : 64496             Local AS        : 64496
Local Address   : n/a              Loop Detect     : Discard
Import Policy   : None Specified - Default Reject
Export Policy   : None Specified - Default Reject
Hold Time      : 90                 Keep Alive     : 30
Min Hold Time  : 0
Cluster Id     : 5.5.5.5           Client Reflect  : Enabled
NLRI           : Unicast           Preference     : 170
TTL Security   : Disabled          Min TTL Value  : n/a
Graceful Restart : Disabled        Stale Routes Time: n/a
Restart Time   : n/a
Auth key chain : n/a
Bfd Enabled    : Disabled          Disable Cap Nego : Disabled
Creation Origin : manual
Flowspec Validate: Disabled
Default Route Tgt: Disabled
Aigp Metric    : Disabled
Split Horizon  : Enabled
Damp Peer Oscill*: Disabled
GR Notification : Disabled          Fault Tolerance : Disabled
Next-Hop Unchang*: None
Routes Resolve T*: Disabled

List of Static Peers
- 192.0.2.1 :

List of Dynamic Peers
- 192.0.2.2
- 192.0.2.3
- 192.0.2.4

Total Peers      : 4                 Established     : 4
-----
Peer Groups : 1
=====
* indicates that the corresponding row element may have been truncated.

```

The BGP session toward PE-1 is static. The short session time is an indication that the BGP session toward PE-1 has been reestablished, as follows:

```
[ ]
A:admin@RR-5# show router bgp summary all

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId          AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
                PktSent OutQ
-----
192.0.2.1
Def. Instance 64496      95   0 00h01m33s 0/0/3 (IPv4)
                16   0
192.0.2.2(D)
Def. Instance 64496      7   0 00h47m44s 1/1/2 (IPv4)
                8   0
192.0.2.3(D)
Def. Instance 64496      94   0 00h45m04s 1/1/2 (IPv4)
                99   0
192.0.2.4(D)
Def. Instance 64496      92   0 00h44m02s 1/1/2 (IPv4)
                97   0
-----
```

Reestablishment of the BGP session is also indicated in log 99, as follows:

```
76 2020/08/19 16:41:37.265 CEST MINOR: BGP #2038 Base Peer 1: 192.0.2.1
"(ASN 64496) VR 1: Group iBGP: Peer 192.0.2.1: moved into established state"

75 2020/08/19 16:41:37.255 CEST WARNING: BGP #2011 Base Peer 1: 192.0.2.1
"(ASN 64496) VR 1: Group iBGP: Peer 192.0.2.1: remote end closed connection"

74 2020/08/19 16:41:37.255 CEST WARNING: BGP #2005 Base Peer 1: 192.0.2.1
"(ASN 64496) VR 1: Group iBGP: Peer 192.0.2.1: sending notification: code CEASE
subcode CONN_COLL_RES"

73 2020/08/19 16:41:37.234 CEST WARNING: BGP #2039 Base Peer 1: 192.0.2.1
"(ASN 64496) VR 1: Group iBGP: Peer 192.0.2.1: sending notification: code CEASE subcode CONFIG_
CHG"

72 2020/08/19 16:41:37.225 CEST WARNING: BGP #2011 Base Peer 1: 192.0.2.1
"(ASN 64496) VR 1: Group iBGP: Peer 192.0.2.1: moved from higher state ESTABLISHED to lower
state IDLE due to event CONFIG_CHG"
```

New and more specific settings apply to static neighbor PE-1, as follows:

```
[ ]
A:admin@RR-5# show router bgp neighbor 192.0.2.1

=====
BGP Neighbor
=====
-----
Peer          : 192.0.2.1
Description   : (Not Specified)
```

```

Group                : iBGP
-----
Peer AS              : 64496          Peer Port           : 49436
Peer Address         : 192.0.2.1
Local AS             : 64496          Local Port           : 179
Local Address        : 192.0.2.5
Peer Type            : Internal        Dynamic Peer       : No
State                : Established     Last State           : Established
Last Event           : recvOpen
Last Error           : Cease (Connection Collision Resolution)
Local Family         : IPv4
Remote Family        : IPv4
Hold Time            : 60              Keep Alive           : 20
Min Hold Time        : 0
Active Hold Time     : 60              Active Keep Alive    : 20
Cluster Id           : 5.5.5.5
---snip---
    
```

The properties of all dynamic peers can be displayed using a single command, as follows:

```

[]
A:admin@RR-5# show router bgp neighbor dynamic

=====
BGP Neighbor
=====
-----
Peer          : 192.0.2.2
Description   : (Not Specified)
Group         : iBGP
-----
Peer AS       : 64496          Peer Port       : 49704
Peer Address  : 192.0.2.2
Local AS      : 64496          Local Port      : 179
Local Address : 192.0.2.5
Peer Type     : Internal        Dynamic Peer   : Yes
State        : Established     Last State      : Established
---snip---
-----
Peer          : 192.0.2.3
Description   : (Not Specified)
Group         : iBGP
-----
Peer AS       : 64496          Peer Port       : 49636
Peer Address  : 192.0.2.3
Local AS      : 64496          Local Port      : 179
Local Address : 192.0.2.5
Peer Type     : Internal        Dynamic Peer   : Yes
State        : Established     Last State      : Established
---snip---
-----
Peer          : 192.0.2.4
Description   : (Not Specified)
Group         : iBGP
-----
Peer AS       : 64496          Peer Port       : 49840
Peer Address  : 192.0.2.4
Local AS      : 64496          Local Port      : 179
Local Address : 192.0.2.5
Peer Type     : Internal        Dynamic Peer   : Yes
State        : Established     Last State      : Established
---snip---
    
```

```
-----
Neighbors shown : 3
=====
* indicates that the corresponding row element may have been truncated.
```

Lowering the dynamic peer limit will not tear down any existing BGP sessions, as follows:

```
# on RR-5:
configure {
  router "Base" {
    bgp {
      group "iBGP" {
        dynamic-neighbor-limit 2
      }
    }
  }
}
```

A hard reset of a running BGP session will result in that BGP session being torn down, as follows:

```
[ ]
A:admin@RR-5# clear router bgp neighbor 192.0.2.4 hard
```

The BGP peer fails to reconnect to the route reflector, because the peer limit has been reached, as follows:

```
80 2020/08/19 17:12:39.585 CEST MINOR: BGP #2037 Base VR 1: Group iBGP
"192.0.2.4: Closing connection: reached dynamic peer limit (2) for BGP group iBGP"

79 2020/08/19 17:12:39.574 CEST WARNING: BGP #2005 Base Peer 1: 192.0.2.4
"(ASN 64496) VR 1: Group iBGP: Peer 192.0.2.4: sending notification: code CEASE
subcode HARD_RESET"

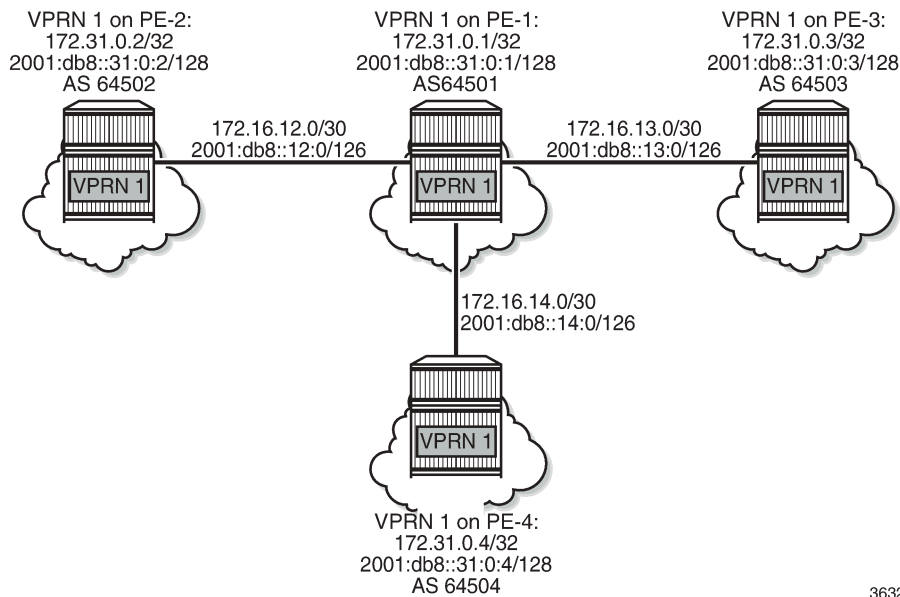
78 2020/08/19 17:12:39.574 CEST WARNING: BGP #2039 Base Peer 1: 192.0.2.4
"(ASN 64496) VR 1: Group iBGP: Peer 192.0.2.4: moved from higher state ESTABLISHED
to lower state IDLE due to event ADMIN_RESET_HARD"

77 2020/08/19 17:12:39.562 CEST INDETERMINATE: LOGGER #2010 Base Clear BGP
"Clear function clearRtrBgpNbr has been run with parameters: rtr-name="Base"
neighbor="192.0.2.4" type="hard". The completion result is: success.
Additional error text, if any, is: "
```

Dynamic BGP peers in multiple ASs

In SR OS Release 19.5.R1 and later, dynamic BGP sessions associated with a single BGP peer group can belong to different peer Autonomous Systems (ASs), both in the base router and in VPRNs. [Figure 143: Example topology with VPRN 1 in different ASs](#) shows the example topology with VPRN 1 configured in different ASs. Each interface in VPRN 1 has an IPv4 and an IPv6 address.

Figure 143: Example topology with VPRN 1 in different ASs



36327

EBGP sessions are established between VPRN 1 on PE-1 and VPRN 1 on the other nodes. In VPRN 1 on PE-2, PE-3, and PE-4, static BGP neighbors are configured. The VPRN configuration on PE-2 is as follows:

```
# on PE-2:
configure {
  service {
    vprn "VPRN 1" {
      admin-state enable
      service-id 1
      customer "1"
      autonomous-system 64502
      router-id 172.31.0.2
      route-distinguisher "1:1"
      vrf-target {
        community "target:1:1"
      }
    }
    bgp {
      router-id 172.31.0.2
      split-horizon true
      group "eBGPv4" {
        next-hop-self true
        peer-as 64501
        family {
          ipv4 true
        }
      }
      group "eBGPv6" {
        next-hop-self true
        peer-as 64501
        family {
          ipv6 true
        }
      }
    }
  }
  neighbor "172.16.12.1" {
```

```

        group "eBGPv4"
        export {
            policy ["exp-vprn-1-v4"]
        }
    }
    neighbor "2001:db8::12:1" {
        group "eBGPv6"
        export {
            policy ["exp-vprn-1-v6"]
        }
    }
}
interface "int-VPRN1-PE-2-PE-1" {
    ipv4 {
        primary {
            address 172.16.12.2
            prefix-length 30
        }
    }
    sap 1/1/1:1 {
    }
    ipv6 {
        address 2001:db8::12:2 {
            prefix-length 126
        }
    }
}
interface "system" {
    loopback true
    ipv4 {
        primary {
            address 172.31.0.2
            prefix-length 32
        }
    }
    ipv6 {
        address 2001:db8::31:0:2 {
            prefix-length 128
        }
    }
}
}
}
}

```

In VPRN 1 on PE-1, dynamic BGP peering is configured for IPv4 prefixes matching 172.16.0.0/16 in AS 64502 (PE-2) or AS 64504 (PE-4) and IPv6 prefixes matching 2001:db8::/107 ASN range from 64502 (PE-2) to 64503 (PE-3). The BGP configuration in VPRN 1 on PE-1 is as follows:

```

# on PE-1:
configure {
    service {
        vprn "VPRN 1" {
            bgp {
                router-id 172.31.0.1
                split-horizon true
                group "eBGPv4" {
                    next-hop-self true
                    dynamic-neighbor-limit 10
                    family {
                        ipv4 true
                    }
                }
                import {
                    policy ["1:1"]
                }
            }
        }
    }
}

```

```

export {
  policy ["exp-vprn-1-v4" "1:1"]
}
dynamic-neighbor {
  match {
    prefix 172.16.0.0/16 {
      allowed-peer-as ["64502" "64504"]
    }
  }
}
}
group "eBGPv6" {
  next-hop-self true
  dynamic-neighbor-limit 10
  family {
    ipv6 true
  }
  export {
    policy ["exp-vprn-1-v6" "1:1"]
  }
  import {
    policy ["1:1"]
  }
  dynamic-neighbor {
    match {
      prefix 2001:db8::/107 {
        allowed-peer-as ["64502..64503"]
      }
    }
  }
}
}
}

```

A dynamic BGP session can be rejected if receiving neighbor BGP OPEN message does not report an AS number in an allowed list: in the "eBGPv4" group, AS 64503 is not allowed and in the "eBGPv6" group, AS 64504 is not allowed. PE-1 sends a notification message with code OPEN and subcode INCORRECT_AS to PE-3 in AS 64503 and the following notification is logged in log 99:

```

14 2020/08/19 16:55:19.697 CEST WARNING: BGP #2005 vprn1 Peer 2: 172.16.13.2
"(ASN 0) VR 2: Group eBGPv4: Peer 172.16.13.2: sending notification: code OPEN subcode INCORRECT_AS"

```

When debugging is enabled for BGP OPEN messages and BGP notifications, the following messages are logged on PE-1: a BGP OPEN message received from PE-3 in AS 64503 and a BGP notification with code OPEN and subcode Bad Peer AS.

```

7 2020/08/19 16:55:19.697 CEST MINOR: DEBUG #2001 vprn1 Peer 2: 172.16.13.2
"Peer 2: 172.16.13.2: NOTIFICATION
Peer 2: 172.16.13.2 - Send BGP NOTIFICATION: Code = 2 (OPEN) Subcode = 2 (Bad Peer AS)"
6 2020/08/19 16:55:19.697 CEST MINOR: DEBUG #2001 vprn1 BGP
"BGP: OPEN
Peer 2: 172.16.13.2 - Received BGP OPEN: Version 4
AS Num 64503: Holdtime 90: BGP_ID 172.31.0.3: Opt Length 20 (ExtOpt F)
Opt Para: Type CAPABILITY: Length = 18: Data:
  Cap_Code GRACEFUL-RESTART: Length 2
  Bytes: 0x0 0x78
  Cap_Code MP-BGP: Length 4
  Bytes: 0x0 0x1 0x0 0x1

```

```

Cap_Code ROUTE-REFRESH: Length 0
Cap_Code 4-OCTET-ASN: Length 4
  Bytes: 0x0 0x0 0xfb 0xf7          # AS 64503
"
    
```

The following BGP summary on PE-1 shows four dynamic BGP neighbors: 172.16.12.2 (in AS 64502), 172.16.14.2 (in AS 64504), 2001:db8::12:2 (in AS 64502), and 2001:db8::13:2 (in AS 64503):

```

[]
A:admin@PE-1# show router bgp summary all

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId          AS PktRcvd InQ Up/Down  State|Rcv/Act/Sent (Addr Family)
                   PktSent OutQ
-----
192.0.2.5
Def. Instance     64496      19   0 00h04m34s 2/2/0 (IPv4)
                   17   0
172.16.12.2(D)
Svc: 1            64502      8   0 00h01m36s 1/1/2 (IPv4)
                   9   0
172.16.14.2(D)
Svc: 1            64504      8   0 00h01m56s 1/1/2 (IPv4)
                   9   0
2001:db8::12:2(D)
Svc: 1            64502      8   0 00h01m54s 1/1/2 (IPv6)
                   9   0
2001:db8::13:2(D)
Svc: 1            64503      8   0 00h01m57s 1/1/2 (IPv6)
                   9   0
-----
    
```

The following command shows that BGP group "eBGPv4" has two dynamic peers (172.16.12.2 and 172.16.14.2) and group "eBGPv6" has two dynamic peers (2001:db8::12:2 and 2001:db8::13:2):

```

[]
A:admin@PE-1# show router 1 bgp group

=====
BGP Group
=====
Group           : eBGPv4
Description      : (Not Specified)
Group Type       : No Type           State           : Up
Peer AS          : n/a              Local AS        : 64501
Local Address    : n/a              Loop Detect     : Ignore
Import Policy    : 1:1
Export Policy    : Default Reject
                 : exp-vprn-1-v4
                 : 1:1
                 : Default Reject
---snip---

List of Static Peers
    
```

```
List of Dynamic Peers
- 172.16.12.2
- 172.16.14.2

Total Peers      : 2                Established      : 2
Group           : eBGPv6
Description      : (Not Specified)
Group Type       : No Type          State           : Up
Peer AS         : n/a              Local AS        : 64501
Local Address    : n/a              Loop Detect     : Ignore
Import Policy    : 1:1
Export Policy    : Default Reject
                  : exp-vprn-1-v6
                  : 1:1
                  : Default Reject
---snip---

List of Static Peers

List of Dynamic Peers
- 2001:db8::12:2
- 2001:db8::13:2

Total Peers      : 2                Established      : 2
-----
Peer Groups : 2
=====
* indicates that the corresponding row element may have been truncated.
```

Conclusion

The use of dynamic BGP peers provides ISPs the means to reduce the configuration file size for routers. This reduces the number of configuration changes to be made to the network over time, which lowers the operational cost of running the network.

EBGP Default Reject Policy

This chapter describes EBGP Default Reject Policy.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information and MD-CLI configuration in this chapter are based on SR OS Release 20.7.R2. The eBGP default reject policy is supported in SR OS Release 19.5.R1 and later.

Overview

To improve security and reliability of Internet routing in the base router and in VPRN routing instances, a default eBGP reject policy rejects all BGP routes when no import or export policies are configured. This policy prevents accidental route leaks. In MD-CLI, the implicit **ebgp-default-reject-policy** is used by default for import and export, which is compliant with RFC 8212, *Default External BGP (EBGP) Route Propagation Behavior without Policies*.

The **ebgp-default-reject-policy** command can be configured in the general **bgp** context, in the BGP **group** context, and in the BGP **neighbor** context. It can be enabled for import direction only, for export direction only, or for both directions. The syntax of the command is as follows:

```
[ex:configure router "Base" bgp group "eBGP"]
A:admin@PE-2# ebgp-default-reject-policy

ebgp-default-reject-policy

export          - Enable default reject export policy for external peers
import         - Enable default reject import policy for external peers
```

The eBGP default reject policy is the last policy in a policy chain.



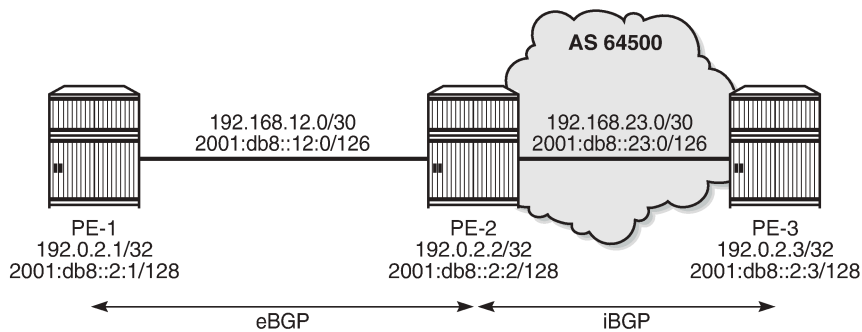
Note:

In MD-CLI, the default behavior is compliant with RFC 8212 (**ebgp-default-reject-policy import/export true**). However, when BGP was initially configured in classic CLI and afterward converted to MD-CLI, the insecure behavior remains for backward compatibility (**ebgp-default-reject-policy import/export false**).

Configuration

[Figure 144: Example topology](#) shows the example topology with three nodes. An eBGP session is established between PE-1 and PE-2; an iBGP session between PE-2 and PE-3.

Figure 144: Example topology



36517

The initial configuration includes:

- Cards, MDAs, ports
- Router interfaces
- SR-ISIS on PE-2 and PE-3 in AS 64500

On PE-1, BGP is configured as follows:

```
# on PE-1:
configure {
  router "Base" {
    autonomous-system 64501
    bgp {
      split-horizon true
      group "eBGP" {
        peer-as 64500
        local-as {
          as-number 64501
        }
      }
    }
    neighbor "192.168.12.2" {
      group "eBGP"
      family {
        ipv4 true
        ipv6 true
        label-ipv4 true
        label-ipv6 true
      }
      export {
        policy ["export-10.1" "export-10.2" "export-10.131" "export-10.132"]
      }
    }
  }
}
```

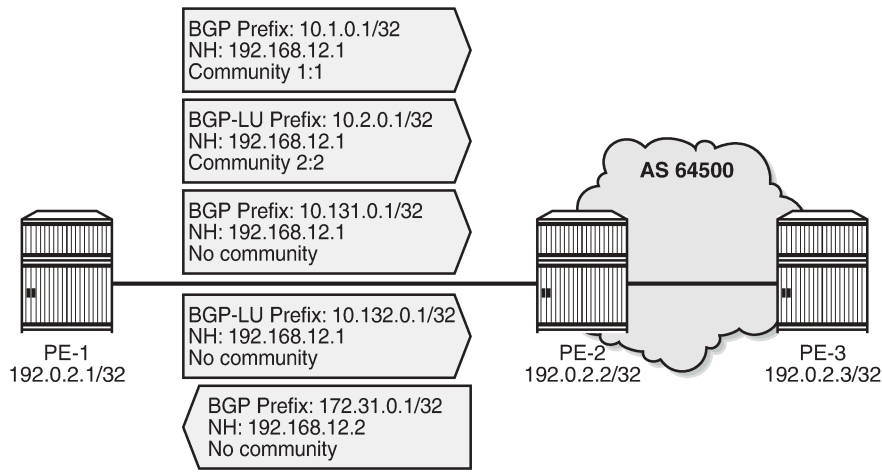
On PE-2, BGP is configured as follows:

```
# on PE-2:
configure {
  router "Base" {
    autonomous-system 64500
    bgp {
      split-horizon true
      next-hop-resolution {
```

```
        labeled-routes {
            transport-tunnel {
                family label-ipv4 {
                    resolution-filter {
                        ldp false
                        sr-isis true
                    }
                }
            }
        }
    }
}
group "eBGP" {
    peer-as 64501
    local-as {
        as-number 64500
    }
}
group "iBGP-IPv4" {
    peer-as 64500
    family {
        ipv4 true
        label-ipv4 true
    }
}
group "iBGP-IPv6" {
    peer-as 64500
    family {
        ipv6 true
        label-ipv6 true
    }
}
neighbor "192.0.2.3" {
    group "iBGP-IPv4"
    next-hop-self true
}
neighbor "192.168.12.1" {
    group "eBGP"
    family {
        ipv4 true
        ipv6 true
        label-ipv4 true
        label-ipv6 true
    }
    export {
        policy ["export-bgp"]
    }
}
neighbor "2001:db8::2:3" {
    group "iBGP-IPv6"
    next-hop-self true
}
}
```

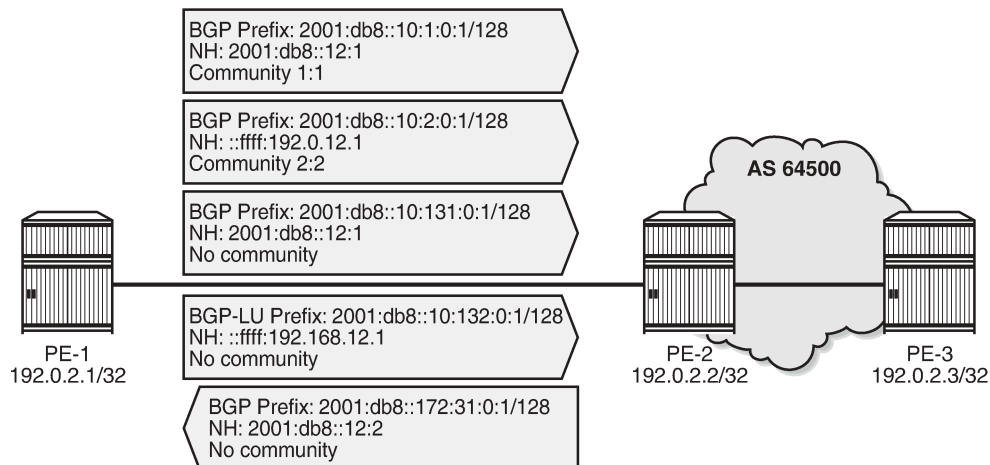
Figure 145: Advertised BGP and BGP-LU IPv4 routes and Figure 146: Advertised BGP and BGP-LU IPv6 routes show the advertised BGP and BGP Labeled Unicast (BGP-LU) routes between PE-1 and PE-2:

Figure 145: Advertised BGP and BGP-LU IPv4 routes



36518

Figure 146: Advertised BGP and BGP-LU IPv6 routes



36519

In this example, an export policy is configured toward eBGP peer 192.168.12.1 on PE-2:

```
[ ]
A:admin@PE-2# show router bgp neighbor 192.168.12.1 | match "Export Policy"
Export Policy          : export-bgp
```

By default, in MD-CLI, the eBGP default reject policy is used. When no eBGP import-policy is configured on PE-2, any route received from an eBGP peer is rejected, as follows:

```
[ ]
A:admin@PE-2# show router bgp neighbor 192.168.12.1 | match "Import Policy"
Import Policy          : None Specified - Default Reject
```

This behavior only applies to eBGP sessions. For iBGP sessions, the reverse is true and the default behavior is to accept. When no iBGP export-policy is configured on PE-2, any received eBGP route is advertised to the iBGP peer (PE-3 in this example), as follows:

```
[ ]
A:admin@PE-2# show router bgp neighbor 192.0.2.3 | match "Export Policy"
Export Policy      : None Specified - Default Accept
```

The BGP default reject policy is implicitly used for import and export. There is no need to configure it explicitly. Both on PE-1 and PE-2, export policies are configured, so the corresponding routes are advertised. However, no import policies are configured, so any route received from an eBGP peer is rejected. The BGP summary on PE-2 shows that for each of the address families, two routes are received from eBGP peer 192.168.12.1, but these routes are rejected:

```
[ ]
A:admin@PE-2# show router bgp summary all

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId      AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
                PktSent OutQ
-----
192.0.2.3
Def. Instance  64500      7   0 00h01m34s 0/0/0 (IPv4)
                12   0           0/0/0 (Lbl-IPv4)
192.168.12.1
Def. Instance  64501     16   0 00h01m55s 2/0/1 (IPv4)
                11   0           2/0/1 (IPv6)
                2/0/0 (Lbl-IPv4)
                2/0/0 (Lbl-IPv6)
2001:db8::2:3
Def. Instance  64500      7   0 00h01m34s 0/0/0 (IPv6)
                12   0           0/0/0 (Lbl-IPv6)
-----
```

The following output shows that the received BGP routes are invalid:

```
[ ]
A:admin@PE-2# show router bgp routes

=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                   Path-Id    IGP Cost
      As-Path                             Label
-----
```

```

i    10.1.0.1/32          None      None
    192.168.12.1        None      0
    64501                -
i    10.131.0.1/32       None      None
    192.168.12.1        None      0
    64501                -
-----
Routes : 2
=====

```

The status of the IPv6, BGP-LU IPv4, and BGP-LU IPv6 routes is the same. The flags for the received routes for the different address families include the 'Rejected' flag:

```

[ ]A:admin@PE-2# show router bgp routes hunt | match Flags
Flags          : Invalid IGP Rejected
Flags          : Invalid IGP Rejected

```

```

[ ]
A:admin@PE-2# show router bgp routes ipv6 hunt | match Flags
Flags          : Invalid IGP Rejected
Flags          : Invalid IGP Rejected

```

```

[ ]
A:admin@PE-2# show router bgp routes label-ipv4 hunt | match Flags
Flags          : Invalid IGP Rejected
Flags          : Invalid IGP Rejected

```

```

[ ]
A:admin@PE-2# show router bgp routes label-ipv6 hunt | match Flags
Flags          : Invalid IGP Rejected
Flags          : Invalid IGP Rejected

```

Import policy

When an import policy is configured, it is possible that some of these routes are accepted. The following import policy on PE-2 accepts incoming routes with communities "1:1" or "2:2":

```

# on PE-2:
configure {
  policy-options {
    community "1:1" {
      member "1:1" { }
    }
    community "2:2" {
      member "2:2" { }
    }
  }
  policy-statement "import-1:1-2:2" {
    entry 10 {
      from {
        community {
          name "1:1"
        }
      }
      action {
        action-type accept
      }
    }
  }
}

```

```

    entry 20 {
      from {
        community {
          name "2:2"
        }
      }
      action {
        action-type accept
      }
    }
  }
}
router "Base" {
  bgp {
    group "eBGP" {
      peer-as 64501
      local-as {
        as-number 64500
      }
      import {
        policy ["import-1:1-2:2"]
      }
    }
  }
}

```

PE-2 accepts BGP route 10.1.0.1/32 with community "1:1", but it rejects route 10.131.0.1/32 because this route has no communities:

```

[]
A:admin@PE-2# show router bgp routes
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i 10.1.0.1/32                             None       None
      192.168.12.1                          None       0
      64501                                   -
i     10.131.0.1/32                          None       None
      192.168.12.1                          None       0
      64501                                   -
-----
Routes : 2
=====

```

The BGP summary on PE-2 shows that one route is accepted and one route is rejected for the IPv4, IPv6, BGP-LU IPv4, and BGP-LU IPv6 address families:

```

[]
A:admin@PE-2# show router bgp summary all
=====

```

```
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId          AS PktRcvd InQ Up/Down  State|Rcv/Act/Sent (Addr Family)
                   PktSent OutQ
-----
192.0.2.3
Def. Instance 64500      81   0 00h38m35s 0/0/1 (IPv4)
                   94   0
192.168.12.1
Def. Instance 64501      90   0 00h38m56s 2/1/1 (IPv4)
                   85   0
                   2/1/1 (IPv6)
                   2/1/0 (Lbl-IPv4)
                   2/1/0 (Lbl-IPv6)
2001:db8::2:3
Def. Instance 64500      81   0 00h38m35s 0/0/1 (IPv6)
                   96   0
-----
```

The following shows that the routes with communities "1:1" or "2:2" are accepted while the other routes are rejected. For each of the address families, there are two routes in the RIB-IN: a first one with community "1:1" or "2:2" (with flags "Used Valid Best IGP") and second one with "No community members" (with flags "Invalid IGP Rejected"), as follows:

```
[ ]
A:admin@PE-2# show router bgp routes hunt | match expression "Comm|Flags"
Community      : 1:1
Flags          : Used Valid Best IGP
Community      : No Community Members
Flags          : Invalid IGP Rejected
Community      : 1:1 # RIB-OUT
Community      : No Community Members # RIB-OUT (172.31.0.1/32)
```

```
[ ]
A:admin@PE-2# show router bgp routes ipv6 hunt | match expression "Comm|Flags"
Community      : 1:1
Flags          : Used Valid Best IGP
Community      : No Community Members
Flags          : Invalid IGP Rejected
Community      : 1:1 # RIB-OUT
Community      : No Community Members # RIB-OUT (172.31.0.1/32)
```

```
[ ]
A:admin@PE-2# show router bgp routes label-ipv4 hunt | match expression "Comm|Flags"
Community      : 2:2
Flags          : Used Valid Best IGP
Community      : No Community Members
Flags          : Invalid IGP Rejected
Community      : 2:2 # RIB-OUT
```

```
[ ]
A:admin@PE-2# show router bgp routes label-ipv6 hunt | match expression "Comm|Flags"
Community      : 2:2
Flags          : Used Valid Best IGP
Community      : No Community Members
Flags          : Invalid IGP Rejected
```

```
Community      : 2:2                # RIB-OUT
```

Conclusion

The eBGP default reject policy is used to improve the security and reliability of Internet routing. The eBGP default reject policy can be combined with other policies and is always evaluated last in the list of policies.

EBGP Route Resolution to a Static Route

This chapter provides information about EBGP route resolution to a static route.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 14.0.R7, but the MD-CLI in the current edition is based on SR OS Release 20.10.R1. EBGP route resolution to a static route is supported in SR OS Release 14.0.R1, and later.

Overview

The configuration in this chapter resembles the configuration in chapter "Inter-AS VPRN Model C" in the Layer 3 Services volume of *7450 ESS, 7750 SR, and 7950 XRS Advanced Configuration Guide — Part II*, but in this chapter, the eBGP peering between the ASBRs is using loopback addresses instead of interface addresses.

Typically, service providers use interface IP addresses in eBGP sessions toward an Autonomous System Border Router (ASBR) of an untrusted ISP, but it is possible to use loopback addresses, such as system IP addresses. This requires the ASBRs to provide visibility on each other's loopback address; for example, by defining static routes. EBGP route resolution to a static route only works for ASBRs that are directly connected. As an alternative, MPLS (for example, RSVP-TE or LDP) can be configured on the interfaces between the ASBRs, which is the only viable solution when the peering ASBRs are multiple hops away.

Configuring MPLS on the interface toward an ASBR of an untrusted ISP is considered insecure. For directly connected ASBRs, EBGP route resolution to a static route mitigates these security issues. On each ASBR, static routes are configured toward the loopback address of the peer ASBR. Additionally, the following command enables labeled routes to be resolved via a static route:

```
configure {
  router "Base" {
    bgp {
      next-hop-resolution {
        labeled-routes {
          allow-static true
        }
      }
    }
  }
}
```

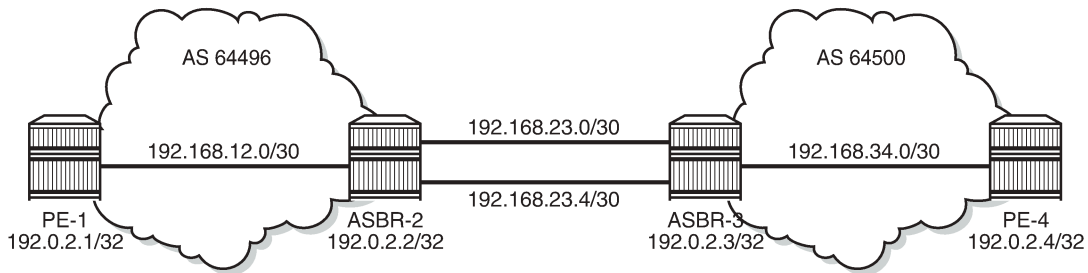
Even with this feature enabled, the system will first try to resolve the BGP next-hop to LDP or RSVP LSPs before the IP route table is attempted. The option is supported for the following address families:

- Labeled IPv4 routes
- VPN-IPv4 and VPN-IPv6 routes

Configuration

Figure 147: Example topology shows the example topology with four routers in two different ASs. ASBR-2 and ASBR-3 are connected via two links, which implies that there will be multiple next-hops configured for the static route entry toward the loopback IP address of the eBGP peer. Also, Equal Cost Multi-Path (ECMP) and BGP multipath need to be enabled between these ASBRs.

Figure 147: Example topology



26311

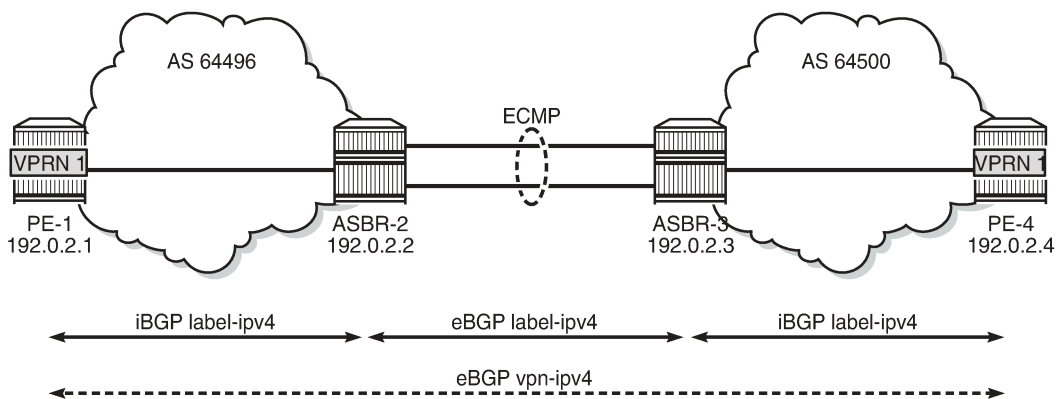
The initial configuration on the nodes includes the following:

- Cards, MDAs, ports
- Router interfaces
- IS-IS as IGP on the interfaces within an AS (alternatively, OSPF could be used)
- LDP on the interfaces within an AS

Figure 148: BGP peering shows the BGP sessions to be configured:

- iBGP sessions for address family labeled IPv4 between the PEs within each AS
- eBGP sessions for address family labeled IPv4 between ASBR-2 and ASBR-3
- a multi-hop eBGP session for address family VPN-IPv4 between PE-1 and PE-4

Figure 148: BGP peering



26312

On PE-1, iBGP is configured for address family labeled IPv4, as follows. The configuration on PE-4 is similar.

```
# on PE-1:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      split-horizon true
      group "iBGP" {
        peer-as 64496
      }
      neighbor "192.0.2.2" {
        export {
          policy "export-bgp"
        }
        group "iBGP"
        family {
          label-ipv4 true
        }
      }
    }
  }
}
```

The following export policy exports the loopback IP prefixes from PE-1 to ASBR-2 (and from PE-4 to ASBR-3):

```
# on PE-1, PE-4:
configure {
  policy-options {
    community "1:0" {
      member "1:0" { }
    }
    prefix-list "PE-sys" {
      prefix 192.0.2.0/28 type longer {
      }
    }
  }
  policy-statement "export-bgp" {
    entry 10 {
      from {
        prefix-list ["PE-sys"]
        protocol {
          name [direct]
        }
      }
      action {
        action-type accept
        community {
          add ["1:0"]
        }
      }
    }
  }
}
```

On ASBR-2, iBGP and eBGP are configured for address family labeled IPv4, as follows. Two links are connecting ASBR-2 to ASBR-3 and, therefore, ECMP and BGP multipath are enabled. For more information about BGP multipath, see chapter [BGP Multipath](#). The BGP configuration on ASBR-3 is similar.

```
# on ASBR-2:
configure {
  policy-options {
    community "1:0" {
      member "1:0" { }
    }
  }
}
```

```
    }
    policy-statement "1:0" {
        entry 10 {
            from {
                community {
                    name "1:0"
                }
            }
            action {
                action-type accept
            }
        }
    }
}
router "Base" {
    autonomous-system 64496
    ecmp 2
    bgp {
        split-horizon true
        multipath {
            max-paths 2
            ebgp 2
        }
        group "eBGP" {
            peer-as 64500
            import {
                policy "1:0"
            }
            export {
                policy "1:0"
            }
        }
        group "iBGP" {
            peer-as 64496
        }
        neighbor "192.0.2.1" {
            group "iBGP"
            family {
                label-ipv4 true
            }
        }
        neighbor "192.0.2.3" {
            advertise-inactive true
            group "eBGP"
            family {
                label-ipv4 true
            }
        }
    }
}
```

On eBGP sessions, routes are by default rejected unless import and export policies are defined (RFC 8212). In this example, the policy "1:0" accepts routes with community "1:0".

On the ASBRs, the BGP routes with the loopback IP addresses of the local AS PEs are not active because IGP routes are preferred. The **advertise-inactive true** option ensures that the ASBRs will also advertise these inactive routes to each other. ASBR-2 advertises prefix 192.0.2.1/32 to ASBR-3; ASBR-3 advertises prefix 192.0.2.4/32 to ASBR-2.

However, no prefixes can be exchanged between the ASBRs because the eBGP session is not in the established state yet; they still lack routing to each other's loopback IP address.

Eventually, the labeled IPv4 routes for prefixes PE-1 and PE-4 will be exchanged between ASBRs and forwarded to the PEs in the peer AS. PE-1 will have a route toward PE-4 in its routing table, and PE-4 will

have a route toward PE-1. Both PEs can then set up a multi-hop eBGP session to each other for address family VPN-IPv4; for example, on PE-1, as follows:

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      group "eBGP_multihop" {
        peer-as 64500
        local-address "192.0.2.1"
        family {
          vpn-ipv4 true
        }
      }
      neighbor 192.0.2.4 {
        group "eBGP_multihop"
        multihop 10
        ebgp-default-reject-policy {
          import false
          export false
        }
      }
    }
  }
}
```

The **ebgp-default-reject-policy export/import false** command allows to advertise to and receive routes from eBGP peer 192.0.2.4 when no export or import policies are defined.

On PE-1, VPRN 1 is configured with loopback address 10.1.1.1/32, as follows:

```
# on PE-1:
configure {
  service {
    vprn "VPRN 1" {
      admin-state enable
      service-id 1
      customer "1"
      route-distinguisher "64496:1"
      vrf-target {
        import-community "target:64500:1"
        export-community "target:64496:1"
      }
      auto-bind-tunnel {
        resolution filter
        resolution-filter {
          ldp true
        }
      }
    }
    interface "loopback" {
      loopback true
      ipv4 {
        primary {
          address 10.1.1.1
          prefix-length 32
        }
      }
    }
  }
}
```

The configuration of PE-4 resembles the configuration of PE-1, whereas the configuration of ASBR-3 resembles that of ASBR-2.

This configuration is almost identical to the configuration in chapter "Inter-AS VPRN Model C" in the Layer 3 Services volume of *7450 ESS, 7750 SR, and 7950 XRS Advanced Configuration Guide — Book II*, with the difference that the eBGP session between the ASBRs does not use interface IP addresses, but

loopback addresses. The problem is that the ASBRs cannot reach each other's loopback IP address, so the eBGP session between the ASBRs cannot be established, which can be verified in the BGP summary, as follows:

```
[ ]
A:admin@ASBR-2# show router bgp summary all

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId          AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
                   PktSent OutQ
-----
192.0.2.1
Def. Instance 64496      14   0 00h04m49s 1/0/0 (Lbl-IPv4)
              14   0
192.0.2.3
Def. Instance 64500      0   0 00h04m49s Connect
              1   0
-----
```

The state of the BGP session toggles between Active and Connect. The last event is an openFail, as follows:

```
[ ]
A:admin@ASBR-2# show router bgp neighbor 192.0.2.3 detail | match "BGP Neighbor"
                                                post-lines 15

BGP Neighbor
=====
-----
Peer          : 192.0.2.3
Description   : (Not Specified)
Group         : eBGP
-----
Peer AS       : 64500           Peer Port      : 0
Peer Address  : 192.0.2.3
Local AS      : 64496           Local Port     : 0
Local Address : 0.0.0.0
Peer Type     : External       Dynamic Peer   : No
State       : Active           Last State    : Connect
Last Event : openFail
Last Error    : Cease (Other Configuration Change)
Local Family  : LABEL-IPv4
```

When the eBGP session between the ASBRs is not established, no IP prefixes will be learned from the peer AS. This implies that PE-1 will not have a route toward PE-4 in its routing table. Therefore, no multi-hop eBGP session can be established between PE-1 and PE-4, which can be shown as follows:

```
[ ]
A:admin@PE-1# show router route-table

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age  Pref
      Next Hop[Interface Name]      Metric
-----
```

```

192.0.2.1/32          Local   Local   00h10m50s  0
  system
192.0.2.2/32          Remote  ISIS    00h10m40s  15
  192.168.12.2
192.168.12.0/30      Local   Local   00h10m50s  0
  int-PE-1-ASBR-2
-----
No. of Routes: 3
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

```

[]
A:admin@PE-1# show router bgp summary all

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId          AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
                  PktSent OutQ
-----
192.0.2.2
Def. Instance 64496      10   0 00h03m19s 0/0/1 (Lbl-IPv4)
                  12   0
192.0.2.4
Def. Instance 64500      0   0 00h02m42s Connect
                  0   0
-----

```

The state of the multi-hop eBGP session toggles between Active and Connect. The last event is openFail, as follows:

```

[]
A:admin@PE-1# show router bgp neighbor 192.0.2.4 detail | match "BGP Neighbor" post-lines 15
BGP Neighbor
=====
Peer           : 192.0.2.4
Description    : (Not Specified)
Group          : eBGP_multihop
-----
Peer AS        : 64500           Peer Port      : 0
Peer Address   : 192.0.2.4
Local AS       : 64496           Local Port     : 0
Local Address  : 0.0.0.0
Peer Type      : External       Dynamic Peer   : No
State        : Connect       Last State    : Active
Last Event   : openFail
Last Error     : Unrecognized Error
Local Family   : VPN-IPv4

```

The loopback IP addresses of the ASBRs can be made reachable by configuring static routes on each ASBR to the loopback IP address of the peer ASBR. This will be sufficient to establish the eBGP session between the ASBRs, but no BGP labeled IPv4 routes will be advertised to PE-1 and PE-4 yet. ASBR-2

and ASBR-3 are connected by two links and the static route entry contains two next-hops; for example, for ASBR-2, as follows. The configuration is similar for ASBR-3.

```
# on ASBR-2:
configure {
  router "Base" {
    static-routes {
      route 192.0.2.3/32 route-type unicast {
        next-hop "192.168.23.2" {
          admin-state enable
        }
        next-hop "192.168.23.6" {
          admin-state enable
        }
      }
    }
  }
}
```

The routing table in ASBR-2 contains two routes toward ASBR-3, as follows:

```
[ ]
A:admin@ASBR-2# show router route-table 192.0.2.3/32

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type  Proto  Age          Pref
  Next Hop[Interface Name]                       Metric
-----
192.0.2.3/32                                     Remote Static  00h00m13s  5
  192.168.23.2                                     1
192.0.2.3/32                                     Remote Static  00h00m13s  5
  192.168.23.6                                     1
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

The eBGP session between the ASBRs is established; for example, on ASBR-2, as follows:

```
[ ]
A:admin@ASBR-2# show router bgp summary all

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId          AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
                  PktSent OutQ
-----
192.0.2.1
Def. Instance 64496      40   0 00h17m38s 1/0/0 (Lbl-IPv4)
                  40   0
192.0.2.3
Def. Instance 64500    5   0 00h00m58s 1/0/1 (Lbl-IPv4)
                  6   0
-----
```

However, the multi-hop eBGP session between PE-1 and PE-4 is not established yet. The state of the multi-hop eBGP session toggles between active and connect and the following output from PE-1 shows that the last event was openFail:

```
[ ]
A:admin@PE-1# show router bgp neighbor 192.0.2.4 detail | match "BGP Neighbor"
                                                    post-lines 15
BGP Neighbor
=====
-----
Peer          : 192.0.2.4
Description   : (Not Specified)
Group         : eBGP_multihop
-----
Peer AS       : 64500           Peer Port      : 0
Peer Address  : 192.0.2.4
Local AS      : 64496           Local Port     : 0
Local Address : 0.0.0.0
Peer Type     : External       Dynamic Peer   : No
State       : Connect       Last State   : Active
Last Event : openFail
Last Error    : Unrecognized Error
Local Family  : VPN-IPv4
```

ASBR-2 advertised an inactive route for prefix 192.0.2.1/32 to ASBR-3 and received from ASBR-3 an inactive route for prefix 192.0.2.4/32. The following output shows that the route for prefix 192.0.2.4/32 is not valid on ASBR-2:

```
[ ]
A:admin@ASBR-2# show router bgp routes label-ipv4
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP Routes
=====
Flag  Network          LocalPref  MED
      Nexthop (Router) Path-Id     IGP Cost
      As-Path
-----
*i   192.0.2.1/32      100        None
      192.0.2.1        None        10
      No As-Path       524285
i   192.0.2.4/32      None       None
      192.0.2.3       None       0
      64500           524285
-----
Routes : 2
=====
```

Consequently, ASBR-2 does not advertise this invalid route to its iBGP peer PE-1 and PE-1 will not have a route toward PE-4 in its routing table, as follows:

```
[ ]
A:admin@PE-1# show router route-table
```

```

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                                Type  Proto  Age          Pref
  Next Hop[Interface Name]                        Metric
-----
192.0.2.1/32                                       Local  Local  00h23m51s  0
  system                                           0
192.0.2.2/32                                       Remote  ISIS   00h23m41s  15
  192.168.12.2                                     10
192.168.12.0/30                                    Local  Local  00h23m51s  0
  int-PE-1-ASBR-2                                0
-----
No. of Routes: 3
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

PE-1 and PE-4 cannot set up a multi-hop eBGP session to one another to exchange routes for VPRN 1. This problem can be solved in two different ways:

1. Enable MPLS (in this example, LDP) on the interfaces between the ASBRs.
2. Enable the following option: **configure router bgp next-hop-resolution labeled-routes allow-static**.

It is risky to enable MPLS toward a peer ASBR belonging to an untrusted ISP, but it is required between distant ASBRs if loopback addresses are used in eBGP peering.

In the following section, the first solution is described (LDP is enabled on the interfaces between the ASBRs); the section after that describes how to enable eBGP route resolution to a static route.

Enable LDP toward peer ASBR

LDP is configured on the interfaces between the ASBRs; for example, on ASBR-2, as follows. The configuration is similar on ASBR-3.

```

# on ASBR-2:
configure {
  router "Base" {
    ldp {
      interface-parameters {
        interface "int-ASBR-2-ASBR-3_1st" {
          ipv4 {
            }
          }
        interface "int-ASBR-2-ASBR-3_2nd" {
          ipv4 {
            }
          }
        }
      }
    }
  }
}

```

ASBR-2 now has a valid, best, and used route for prefix 192.0.2.4/32, as follows:

```

[]
A:admin@ASBR-2# show router bgp routes label-ipv4
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====

```



```

Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
BGP Routes
=====
Flag Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
*i 192.0.2.1/32             100        None
   192.0.2.1                None        10
   No As-Path                None        524285
u*>i 192.0.2.4/32          None        None
     192.0.2.3              None        1
     64500                   None        524285
-----
Routes : 2
=====

```

PE-1 has a valid route for prefix 192.0.2.4/32, as follows:

```

[]
A:admin@PE-1# show router bgp routes label-ipv4
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
BGP Routes
=====
Flag Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
u*>i 192.0.2.4/32          100        None
     192.0.2.2              None        10
     64500                   None        524282
-----
Routes : 1
=====

```

The following routing table shows that PE-1 has a BGP labeled route toward PE-4:

```

[]
A:admin@PE-1# show router route-table
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age           Pref
  Next Hop[Interface Name]  Path  Age      Metric
-----
192.0.2.1/32                Local  Local  00h43m23s    0
  system
192.0.2.2/32                Remote  ISIS   00h43m13s    15
  192.168.12.2
192.0.2.4/32                Remote  BGP_LABEL 00h17m57s    170

```

```

192.0.2.2 (tunneled)
192.168.12.0/30 Local Local 00h43m23s 0
  int-PE-1-ASBR-2 0
-----
No. of Routes: 4
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

A multi-hop eBGP session is established for address family VPN-IPv4 between PE-1 and PE-4, as follows:

```

[]
A:admin@PE-1# show router bgp summary all
=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId      AS PktRcvd InQ Up/Down State|Rcv/Act/Sent (Addr Family)
              PktSent OutQ
-----
192.0.2.2
Def. Instance 64496      93   0 00h44m01s 1/1/1 (Lbl-IPv4)
              94   0
192.0.2.4
Def. Instance 64500      46   0 00h21m06s 1/1/1 (VpnIPv4)
              47   0
-----

```

The loopback address defined in VPRN 1 on PE-4 (10.2.2.2/32) is advertised as VPN-IPv4 route in this multi-hop eBGP session on PE-1, as follows:

```

[]
A:admin@PE-1# show router bgp routes vpn-ipv4
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag Network LocalPref MED
      Nexthop (Router) Path-Id IGP Cost
      As-Path Label
-----
u*>i 64500:1:10.2.2.2/32 None None
      192.0.2.4 None 0
      64500 524284
-----
Routes : 1
=====

```

The routing table for VPRN 1 on PE-1 includes a BGP-VPN route to PE-4, as follows:

```
[ ]
A:admin@PE-1# show router 1 route-table

=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]
Type Proto Age Pref
Metric
-----
10.1.1.1/32 Local Local 00h44m02s 0
loopback 0
10.2.2.2/32 Remote BGP VPN 00h23m23s 170
192.0.2.4 (tunneled:BGP) 0
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
B = BGP backup route available
L = LFA nexthop available
S = Sticky ECMP requested
=====
```

To restore the configuration, LDP is disabled on the interfaces between the ASBRs, as follows for ASBR-2. The configuration is similar on ASBR-3.

```
# on ASBR-2:
configure {
  router "Base" {
    ldp {
      interface-parameters {
        delete interface "int-ASBR-2-ASBR-3_1st" {
        }
        delete interface "int-ASBR-2-ASBR-3_2nd" {
        }
      }
    }
  }
}
```

EBGP route resolution to a static route

The static routes are already configured on both ASBRs and the eBGP session between the ASBRs is established.

Multi-hop EBGP labeled IPv4 route resolution to a static route needs to be enabled on ASBR-2 and ASBR-3 using the following command:

```
# on ASBR-2, ASBR-3:
configure {
  router "Base" {
    bgp {
      next-hop-resolution {
        labeled-routes {
          allow-static true
        }
      }
    }
  }
}
```

On ASBR-2, the labeled IPv4 route for prefix 192.0.2.4/32 is now valid, best, and used, as follows:

```
[ ]
A:admin@ASBR-2# show router bgp routes label-ipv4
```

```

=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                  l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
*i    192.0.2.1/32                          100        None
      192.0.2.1                             None       10
      No As-Path                             524285
u*>i 192.0.2.4/32                          None       None
      192.0.2.3                             None       1
      64500                                   524285
-----
Routes : 2
=====

```

PE-1 learns the following BGP labeled IPv4 route for prefix 192.0.2.4/32 from ASBR-2:

```

[]
A:admin@PE-1# show router bgp routes label-ipv4
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                  l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i 192.0.2.4/32                          100        None
      192.0.2.2                             None       10
      64500                                   524284
-----
Routes : 1
=====

```

The routing table on PE-1 contains a BGP labeled IPv4 route to 192.0.2.4/32:

```

[]
A:admin@PE-1# show router route-table
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age           Pref
Next Hop[Interface Name]          Metric
-----
192.0.2.1/32                      Local Local  01h23m40s    0

```

```

system
192.0.2.2/32 Remote ISIS 01h23m30s 15
192.168.12.2 10
192.0.2.4/32 Remote BGP_LABEL 00h06m39s 170
192.0.2.2 (tunneled) 10
192.168.12.0/30 Local Local 01h23m40s 0
int-PE-1-ASBR-2 0
-----
No. of Routes: 4
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

The multi-hop eBGP session between PE-1 in AS 64496 and PE-4 in AS 64500 is established, as follows:

```

[]
A:admin@PE-1# show router bgp summary all

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId AS PktRcvd InQ Up/Down State|Rcv/Act/Sent (Addr Family)
          PktSent OutQ
-----
192.0.2.2
Def. Instance 64496 164 0 01h18m54s 1/1/1 (Lbl-IPv4)
                163 0
192.0.2.4
Def. Instance 64500 57 0 00h01m25s 1/1/1 (VpnIPv4)
                11 0
-----

```

The loopback address defined in VPRN 1 on PE-4 (10.2.2.2/32) is advertised as VPN-IPv4 route in this multi-hop eBGP session on PE-1, as follows:

```

[]
A:admin@PE-1# show router bgp routes vpn-ipv4

=====
BGP Router ID:192.0.2.1 AS:64496 Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag Network LocalPref MED
      Nexthop (Router) Path-Id IGP Cost
      As-Path Label
-----
u*>i 64500:1:10.2.2.2/32 None None
      192.0.2.4 None 0
      64500 524284

```

```
-----
Routes : 1
=====
```

The routing table for VPRN 1 on PE-1 includes the following BGP-VPN route to 10.2.2.2/32:

```
[ ]
A:admin@PE-1# show router 1 route-table

=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]          Metric
-----
10.1.1.1/32                        Local  Local   01h20m41s    0
   loopback
10.2.2.2/32                        Remote BGP VPN 00h05m26s    170
   192.0.2.4 (tunneled:BGP)         0
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```

The results are similar on PE-4 and PE-1, and on ASBR-3 and ASBR-2.

For directly connected ASBRs, inter-AS VPRN model C can be configured using loopback addresses on the ASBRs without the need to enable MPLS between the ASBRs.

Conclusion

Most service providers use interface IP addresses in eBGP sessions, in which case this feature is not needed. However, some providers build directly connected eBGP sessions based on loopback interfaces. The system interface of the peer ASBR must be reachable and the labeled IPv4 routes for the remote AS PEs must be advertised to the local AS PEs. This advertisement can be achieved by configuring static routes on the ASBRs to the loopback address of their eBGP peer and enabling the eBGP route resolution to a static route. Enabling eBGP route resolution to a static route is much more secure than enabling MPLS on the interface to the peer ASBR of an untrusted ISP. However, when the ASBRs are distant and loopback addresses are used for the eBGP peering, MPLS must be enabled between the ASBRs.

Flexible Algorithm for IS-IS

This chapter describes Flexible Algorithm for IS-IS.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information in this chapter is based on SR OS Release 20.7.R1. The MD-CLI configuration in the current edition is based on SR OS Release 20.10.R1.

Overview

By default, an IGP-computed path is based on the shortest IGP metric, but frequently these paths are accompanied by traffic-engineered paths that are used to meet the requirements of the network. These traffic-engineered paths are facilitated by RSVP-TE or SR-TE, both of which perform source routing based on a set of metrics and constraints. In many networks this works well, but for some operators the overhead of traffic engineering in this manner is perceived as complex or costly.

Flexible Algorithm (or Flex-Algorithm) — as described in *draft-ietf-isr-flex-algo* — provides a way for IGPs to compute constraint-based paths across a domain. Flex-Algorithm uses extensions to IS-IS and OSPF to advertise TLVs containing one or more Flexible Algorithm Definitions (FADs). Each FAD is associated with a numeric identifier and identifies a set of metrics and constraints to calculate the best path along the constrained topology.

When used with Segment Routing (SR), one or more Prefix Node-SIDs can be associated with a Flex-Algorithm identifier, thereby providing a level of traffic engineering without any associated control plane overhead or additional label stack imposition. The classic SPF technology used for shortest path calculation is referred to as algorithm 0. In SR OS Release 20.7.R1, up to five additional flexible algorithms can be supported.

This chapter provides an overview of the operation of Flex-Algorithm with IS-IS and how it is applicable to SR; specifically, SR-MPLS.

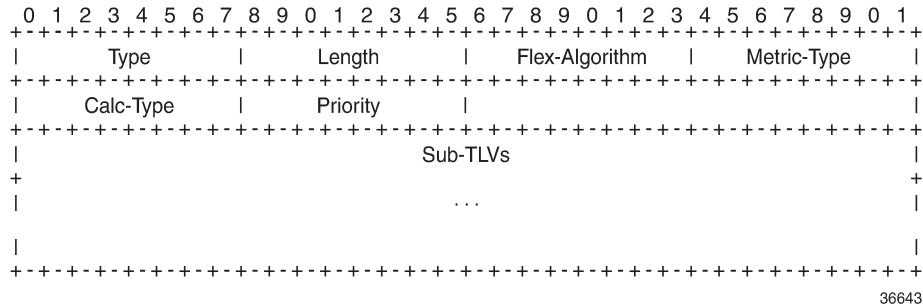
Flexible Algorithm Definition

A FAD is the construct that identifies how a path for a Flex-Algorithm will be computed, and consists of three components:

- A calculation type
- A metric type
- A set of constraints, such as include or exclude statements

To guarantee loop-free forwarding for paths computed with a Flex-Algorithm, all routers that participate in that Flex-Algorithm must receive the definition of it. In IS-IS, the definition of the Flex-Algorithm is advertised using the FAD sub-TLV, which is a sub-TLV of the Router Capability TLV and has area scope, as shown in [Figure 149: IS-IS FAD sub-TLV](#).

Figure 149: IS-IS FAD sub-TLV



The Type and Length fields are self-explanatory. The Flex-Algorithm field contains a numeric identifier in the range 128 to 255 that is associated with the FAD through configuration. The Metric-Type field contains one of IGP metric (0), Min Unidirectional Link Delay (1), or TE Default Metric (2). The Calc-Type field contains a value from 0 to 127, identifying the IGP algorithm type, such as shortest path (0). One or more sub-TLV fields may be present to specify "colors" that are used to include or exclude links during the Flex-Algorithm path computation. These are encoded using Exclude Admin Group, Include-any Admin Group, Include-all Admin-Group, and Exclude SRLG sub-TLVs.

The Sub-TLV field may also contain a Flags sub-TLV. In *draft-ietf-isr-flex-algo*, only the M-flag (Prefix Metric) is defined. The M-flag indicates that the Flex-Algorithm Prefix Metric (FAPM) sub-TLV must be advertised with the prefix. The FAPM is not a sub-TLV of the FAD, but rather a sub-TLV of the Extended IP Reachability TLV, and is intended to assist with inter-area and inter-domain Flex-Algorithm path calculations.

Any IGP shortest-path tree calculation is limited to a single area, and the same applies to Flex-Algorithm. To allow for inter-area or inter-domain Flex-Algorithm calculations, the FAPM sub-TLV can be attached to Extended IP Reachability TLVs that are advertised between areas or domains. The FAPM sub-TLV contains the metric equivalent to the metric of the redistributing router to reach the prefix. If the FAD Flags sub-TLV has the M-flag set, the FAPM must be used when calculating prefix reachability for inter-area and inter-domain prefixes.

Only a subset of the routers participating in each Flex-Algorithm need to advertise the definition of the Flex-Algorithm. However, every router that is part of the intended Flex-Algorithm topology must be configured to participate in the Flex-Algorithm. If a router is not configured to participate in a specific Flex-Algorithm, it ignores FAD sub-TLV advertisements for that Flex-Algorithm.

Application-specific link attributes

Advertisement of link attributes for the purpose of traffic engineering was initially introduced by RFC 5305, which included a number of sub-TLVs encoded within the Extended IS Reachability TLV, such as admin group, TE default metric, maximum link bandwidth, and unreserved bandwidth.

RFC 7308 updated RFC 5305 by increasing the size of the admin group sub-TLV, thereby allowing for advertisement of more than the standard 32 admin groups per link. RFC 5305 was again updated by RFC 8570, which proposed the use of metric extensions, adding additional sub-TLVs to the Enhanced

IS Reachability TLV, such as unidirectional link delay, unidirectional link loss, and unidirectional available bandwidth. These traffic-engineering extensions have been widely deployed for RSVP-TE purposes.

Other applications that also make use of traffic-engineering link attributes have been defined, such as SR, Loopfree Alternates (LFAs), and Flex-Algorithm. If these applications coexist, it may be advisable to unambiguously indicate which traffic-engineering attributes apply to which application. Their requirements may differ on a link-to-link basis, or two applications may not be fully congruent; for example, SR may not be fully deployed network-wide. For these reasons, Flex-Algorithm specifies the use of Application-Specific Link Attributes (ASLAs), from *draft-ietf-isis-te-app*, which defines two new code points for IS-IS ASLA advertisements:

- ASLA sub-TLV for Extended IS Reachability and Neighbor Link Attributes TLVs (TLVs 22, 23, 25, 141, 222, and 223)
- Application-Specific Shared Risk Link Group (SRLG) TLV

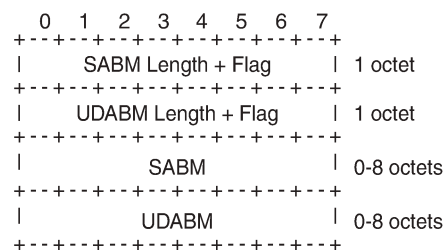
The ASLA sub-TLV contains Link Attribute sub-sub-TLVs, the format of which matches the existing formats defined in RFC 5305, RFC 7308, and RFC 8570. The Application-Specific SRLG TLV encodes link identifier sub-TLVs, such as IPv4/IPv6 Interface address, IPv4/IPv6 Neighbor address, and Link Local/Remote Identifiers. SR OS will advertise Application-Specific SRLG TLVs, but does not use SRLG TLVs for computing SRLG-diverse paths in Release 20.7.R1. Support for LFA (primary/backup) SRLG diversity for Flex-Algorithm is provided using locally configured LFA policies.

Each of the ASLA sub-TLV and Application-Specific SRLG TLV advertisements are coupled with an Application Identifier Bit Mask that identifies the applications associated with an advertisement. Two bit masks are available for use:

- the Standard Applications Bit Mask (SABM) is used for applications, where the definition of each bit is controlled by IANA.
- the User-Defined Applications Bit Mask (UDABM) allows for future non-standard extensibility.

The encoding shown in [Figure 150: Application Identifier Bit Mask](#) is used by both the ASLA sub-TLV and the Application-Specific SRLG TLV.

Figure 150: Application Identifier Bit Mask



36644

The SABM Length + Flag field contains a single L-flag, known as the "Legacy" flag. When the L-flag is set in the Application Identifier Bit Mask, all the applications specified in the bit mask must use the legacy traffic-engineering advertisements for the corresponding link. That is, link attributes should be carried as sub-TLVs of the Extended IS Reachability TLV rather than sub-sub-TLVs of the ASLA sub-TLV. This allows for a level of backward compatibility such that legacy advertisements may continue to be used if:

- Only RSVP-TE is deployed
- Only SR /LFA is deployed

- A combination of RSVP-TE and SR/LFA is deployed, but the set of links that each application uses are fully congruent.

The UDABM Length + Flag field contains a single R-flag, which is reserved for future use.

The SABM field defines four bits to identify applications:

- The R-bit (bit 0) specifies RSVP-TE
- The S-bit (bit 1) specifies SR
- The F-Bit (bit 2) specifies LFA
- The X-bit (bit 3) specifies Flex-Algorithm

Applicability of Flex-Algorithm to SR

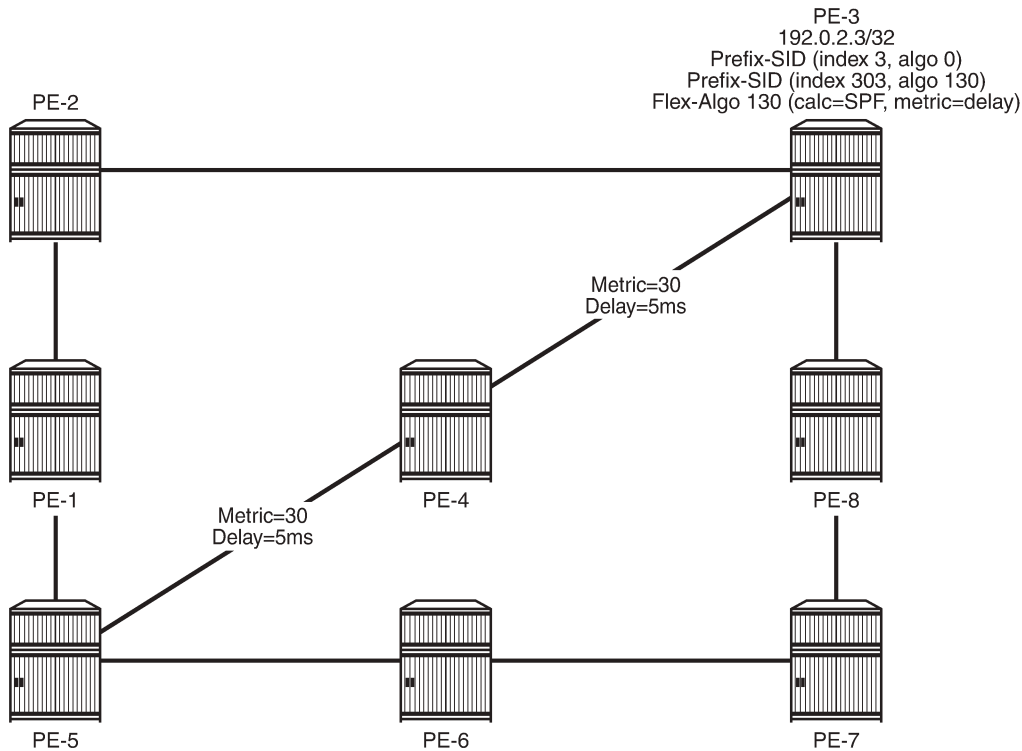
A router may use various algorithms when calculating reachability to other nodes or prefixes attached to those nodes. RFC 8667 — *IS-IS extensions for SR* — describes the use of the SR-Algorithm sub-TLV (carried as part of the Router Capabilities TLV) to advertise the algorithms that the router can support. By default, an SR router will signal support for algorithm 0 (metric-based SPF). To advertise participation for a specific Flex-Algorithm for SR, the Flex-Algorithm value must also be advertised in the SR-Algorithm sub-TLV.

When an SR router advertises a Prefix SID, it includes an SR-algorithm, so it is possible to associate a Prefix SID with a specific algorithm. For example, a router may advertise prefix P1 with Prefix SID {index=1, algorithm=0} and prefix P2 with Prefix SID {index=2, algorithm=128}. This indicates to other SR routers that to reach prefix P1, the default metric-based SPF should be used to calculate the best path, and to reach prefix P2, Flex-Algorithm 128 (and whatever that algorithm dictates) should be used.

Equally, in an SR-MPLS environment with an SR Global Block (SRGB) of {1000-1999}, a router may advertise prefix P1 with Prefix SID {index=1, algorithm=0}, and also Prefix SID {index=101, algorithm=129}. This indicates to other SR routers that when label 1001 is the active label to reach prefix P1, the default metric-based SPF should be used to calculate the best path, and when label 1101 is the active label, Flex-Algorithm 129 should be used.

[Figure 151: Flexible Algorithm example in an SR-MPLS domain](#) shows an SR-MPLS domain where all links have metric 10 except for links PE-5-PE-4 and PE-4-PE-3, which both have metric 30. All links have a unidirectional link latency of 10 ms, except for links PE-5-PE-4 and PE-4-PE-3, which both have a unidirectional latency of 5 ms. All routers use an SRGB of {1000-1999}.

Figure 151: Flexible Algorithm example in an SR-MPLS domain



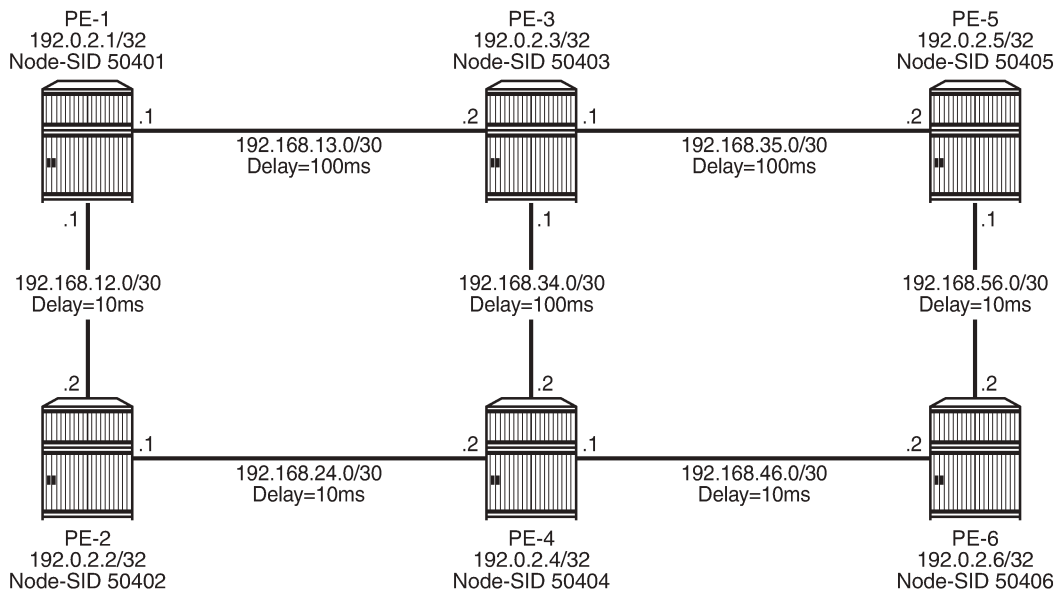
36645

In addition to the default algorithm 0 (metric-based SPF), all routers participate in Flex-Algorithm 130 with FAD {calc-type=SPF, metric=delay, constraints=none}. Router PE-3 advertises prefix 192.0.2.3/32 with Prefix SID {index=3, algorithm=0} and Prefix SID {index=303, algorithm=130}. Router PE-5 has an SR-TE LSP provisioned with a destination of PE-3 (192.0.2.3) and a top (active) label of 1003. As a result, it is associated with algorithm 0 and uses the shortest path IGP metric PE-5-PE-1-PE-2-PE-3 to reach its destination. Router PE-5 is also provisioned with a second SR-TE LSP, again with a destination of PE-3 (192.0.2.3), but this time with a top (active) label of 1303. This second LSP is associated with Flex-Algorithm 130 and uses the shortest path delay metric PE-5-PE-4-PE-3 to reach its destination.

Example topology

Figure 152: Example topology shows the example topology used in this chapter. All routers within the example topology form part of Autonomous System 64496 and belong to the same IS-IS Level-2 area. All IGP link metrics are 100 and are symmetric. Unidirectional link delay is also configured, and all links have a delay of 10 ms, with the exception of links PE-1-PE-3, PE-3-PE-5, and PE-3-PE-4, which have a delay of 100 ms. SR is enabled within the domain, and the associated Node-SIDs used as a baseline are shown (Adj-SIDs are not shown for the purpose of clarity). The SRGB in use is {50000-54999}.

Figure 152: Example topology



36646

An additional step is required if a Flex-Algorithm uses a metric-type of delay. Before the delay metric can be advertised, a value for that metric needs to be derived. There are various methods available to do this (including OAM probes and so on), but currently the only method that SR OS supports is static configuration. The following output provides an example of static configuration of delay metric at PE-1. The delay is entered as an if-attribute under each interface and is expressed in microseconds. As per [Figure 152: Example topology](#), the link PE-1-PE-2 has a delay metric of 10 ms, while the link PE-1-PE-3 has a delay metric of 100 ms.

```
# on PE-1:
configure {
  router "Base" {
    interface "int-PE-1-PE-2" {
      if-attribute {
        delay {
          static 10000
        }
      }
    }
    interface "int-PE-1-PE-3" {
      if-attribute {
        delay {
          static 100000
        }
      }
    }
  }
}
```

Configuration

The following steps are required to configure and enable the use of Flex-Algorithm:

- Enable the use of ASLAs

- Configure and advertise the FAD
- Configure Flex-Algorithm participation
- Configure a Flex-Algorithm Prefix Node-SID
- Configure traffic steering using Flex-Algorithm

These steps are described in the following subsections.

Enable the use of ASLAs

Flex-Algorithm specifies the use of ASLAs for advertisement of traffic-engineering information. If not already enabled, enable these under the IS-IS context for all routers in the domain, as follows:

```
# on all PEs:
configure {
  router "Base" {
    isis 0 {
      traffic-engineering-options {
        application-link-attributes {
        }
      }
    }
  }
}
```

For backward compatibility, the **application-link-attributes** command has an optional **legacy** argument, which allows link attributes to be encoded in the legacy manner as sub-TLVs of the Extended IS Reachability TLV, rather than being encoded as (sub-)sub-TLVs of the ASLA sub-TLV.

The following output shows how ASLAs are advertised as sub-TLVs of the Extended IS Reachability TLV. The output is taken at PE-1 and is truncated to include only the IS neighbor PE-3. Within the Extended IS Reachability TLV, there are three ASLA sub-TLVs:

- The first is non-legacy (L-bit is not set) and has an SABM field that has the R-bit and S-bit set, indicating that these attributes are specific to RSVP-TE and SR, respectively. The link attributes include Max Link Bandwidth and TE Metric.
- The second is non-legacy and has an SABM field with R-bit set, indicating that the intended application is RSVP-TE, and contains reservable and unreserved bandwidth link attributes.
- The third ASLA sub-TLV is non-legacy and has the X-bit set, indicating that this is specific only to Flex-Algorithm. This sub-TLV contains the Delay and TE Metric link attributes.

```
[ ]
A:admin@PE-1# show router isis database PE-1.00-00 detail

=====
Rtr Base ISIS Instance 0 Database (detail)
=====

Displaying Level 2 database
-----
LSP ID      : PE-1.00-00                Level      : L2
Sequence    : 0x114                    Checksum   : 0xe884   Lifetime   : 38293
Version     : 1                        Pkt Type  : 20       Pkt Ver    : 1
Attributes  : L1L2                     Max Area  : 3        Alloc Len  : 1492
SYS ID      : 1920.0000.2001           SysID Len : 6        Used Len   : 607
---snip---
TE IS Nbrs  :
  Nbr       : PE-3.00
  Default Metric : 100
```

```

Sub TLV Len      : 103
IF Addr   : 192.168.13.1
Nbr IP    : 192.168.13.2
TE APP LINK ATTR :
  SABML-flag:Non-Legacy SABM-flags:R S
    MaxLink BW: 10000000 kbps
    TE Metric : 100
TE APP LINK ATTR :
  SABML-flag:Non-Legacy SABM-flags:R
    Resvble BW: 10000000 kbps
    Unresvd BW:
      BW[0] : 10000000 kbps
      BW[1] : 10000000 kbps
      BW[2] : 10000000 kbps
      BW[3] : 10000000 kbps
      BW[4] : 10000000 kbps
      BW[5] : 10000000 kbps
      BW[6] : 10000000 kbps
      BW[7] : 10000000 kbps
TE APP LINK ATTR :
  SABML-flag:Non-Legacy SABM-flags: X      # when FAD is defined and applied
    Delay      : 100000
    TE Metric  : 100
Adj-SID: Flags:v4VLP Weight:0 Label:150013
Adj-SID: Flags:v6VL Weight:0 Label:524272
---snip---

```

Configure FAD and participation

To define the FAD, the following example uses a metric-type of delay with no other constraints. As previously described, not all participating routers need to advertise the FAD; only their participation in it. Therefore, in this example, PE-1 and PE-5 are used to advertise the FAD and the following configuration is applied to both routers.

First, the FAD is created with a name under the **flexible-algorithm-definitions** context. After the flex-algo context has been created, the metric-type (IGP, TE metric, delay) and any other constraints (include-all, include-any, exclude) can be configured within it. It is also possible to configure a priority value for the flex-algo in the range 0 to 255. If multiple FAD advertisements are received, the highest priority will be selected. If priorities are equal, the FAD advertised by the highest router ID is selected. The default priority is 100.

```

# on PE-1, PE-5:
configure {
  routing-options {
    flexible-algorithm-definitions {
      flex-algo "FlexAlgo-128" {
        admin-state enable
        description "FlexAlgo-128-Delay-Metric"
        metric-type delay
      }
    }
  }
}

```

After the FAD has been defined, it can be advertised into IS-IS. This is done under the flexible-algorithms context within the IS-IS instance. The Flex-Algorithm must initially be assigned a numeric identifier in the range 128 to 255, after which the **advertise** command is used to advertise the previously configured FAD "FlexAlgo-128". The **participate** command is used to configure participation for the specific Flex-Algorithm and must be enabled on all routers that are part of this Flex-Algorithm topology. Finally, LFA may be enabled. If it is, the LFA SPF will use the same Flex-Algorithm topology as that used to calculate the

primary path. Also, LFA settings (such as TI-LFA, Remote-LFA) within a Flex-Algorithm are inherited from the base IS-IS/LFA configuration.

PE-1 and PE-5 both advertise and participate in the Flex-Algorithm, as follows:

```
# on PE-1, PE-5:
configure {
  router "Base" {
    isis 0 {
      flexible-algorithms {
        admin-state enable
        flex-algo 128 {
          participate true
          advertise "FlexAlgo-128"
          loopfree-alternate { }
        }
      }
    }
  }
}
```

What remains is to configure all other routers in the domain to participate in the same Flex-Algorithm, as in the following output. This is essentially the same configuration as applied to PE-1 and PE-5, with the exception of the advertise statement.

```
# on PE-2, PE-3, PE-4, PE-6:
configure {
  router "Base" {
    isis 0 {
      flexible-algorithms
        admin-state enable
        flex-algo 128 {
          participate true
          loopfree-alternate { }
        }
      }
    }
  }
}
```

After the FAD with Flex-Algorithm identifier 128 has been advertised and all routers have signaled that they participate in this Flex-Algorithm, it can be validated with router outputs. The following output shows the relevant parts of the IS-IS LSP advertised by PE-1. Within the Router Capabilities TLV, there are two notable additions. The SR-Algorithm sub-TLV shows the default metric-based SPF (algorithm 0), but now includes algorithm 128, showing its participation in this algorithm. There is also a FAD sub-TLV containing the definition of Flex-Algorithm 128 with metric-type of delay. The FAD has a Flags sub-TLV with the M-flag set, indicating that the FAPM must be used when calculating prefix reachability for inter-area and inter-domain prefixes.

```
[ ]
A:admin@PE-1# show router isis database PE-1.00-00 detail

=====
Rtr Base ISIS Instance 0 Database (detail)
=====

Displaying Level 2 database
-----
LSP ID      : PE-1.00-00          Level      : L2
Sequence    : 0x114              Checksum   : 0xe884    Lifetime   : 49976
Version     : 1                  Pkt Type  : 20        Pkt Ver    : 1
Attributes  : L1L2              Max Area  : 3          Alloc Len  : 1492
SYS ID      : 1920.0000.2001     SysID Len : 6          Used Len   : 607
```

```

---snip---
Router Cap : 192.0.2.1, D:0, S:0
TE Node Cap : B E M P
SR Cap: IPv4 MPLS-IPv6
  SRGB Base:50000, Range:5000
SR Alg: metric based SPF, 128
Node MSD Cap: BMI : 12 ERLD : 15
FAD Sub-Tlv:
  Flex-Algorithm   : 128
  Metric-Type      : delay
  Calculation-Type : 0
  Priority          : 100
  Flags: M
---snip---

```

Configure a Flex-Algorithm Prefix Node-SID

At each egress node, a Prefix Node-SID must be assigned to each Flex-Algorithm in use. This will be advertised as a Prefix SID sub-TLV that will contain (among other things) the algorithm to be used to reach the associated Prefix Node-SID. The Node-SID is taken from the generic SRGB; no special or dedicated label space is required. In the following example, PE-5 is the egress router and the relevant configuration is shown in the following output. Under the IS-IS system interface context, the Node-SID label assigned to algorithm 0 is 50405 and is generic SR configuration. Within the same context, a sub-context is created for flex-algo 128 for which a Prefix SID label of 54405 is assigned.

```

# on PE-5:
configure {
  router "Base" {
    isis 0 {
      interface "system" {
        passive true
        level-capability 2
        ipv4-node-sid {
          label 50405
        }
        flex-algo 128 {
          ipv4-node-sid {
            label 54405
          }
        }
      }
    }
  }
}

```

When configured, the additional Prefix SID advertisement can be viewed in the PE-5 advertised IS-IS LSP.

```

[]
A:admin@PE-5# show router isis database PE-5.00-00 detail
=====
Rtr Base ISIS Instance 0 Database (detail)
=====

Displaying Level 2 database
-----
LSP ID   : PE-5.00-00           Level    : L2
Sequence : 0x101                Checksum : 0x9c1c   Lifetime : 53986
Version  : 1                    Pkt Type : 20      Pkt Ver  : 1
Attributes: L1L2                Max Area : 3        Alloc Len: 1492
SYS ID   : 1920.0000.2005       SysID Len: 6        Used Len  : 650
---snip---

```



```
TE IP Reach :
Default Metric : 0
Control Info: S, prefLen 32
Prefix : 192.0.2.5
Sub TLV :
  Prefix-SID Index:405, Algo:0, Flags:NnP
  Prefix-SID Index:4405, Algo:128, Flags:NnP
---snip---
```

The Flex-Algorithm Prefix SID can also be viewed using the **show router prefix-sids** command with the relevant flex-algo extension. The index is 4405 and with an SRGB start-label of 50000, this equates to the configured label value of 54405.

```
[ ]
A:admin@PE-1# show router isis prefix-sids algo 128

=====
Rtr Base ISIS Instance 0 Prefix/SID Table
=====
Prefix                               SID           Lvl/Typ      SRMS   AdvRtr
                               Algo          MT           Flags
-----
192.0.2.5/32                         4405         2/Int.      N      PE-5
                               128         0           NnP
-----
No. of Prefix/SIDs: (1 unique)
-----
SRMS : Y/N = prefix SID advertised by SR Mapping Server (Y) or not (N)
      S    = SRMS prefix SID is selected to be programmed
Flags: R    = Re-advertisement
      N    = Node-SID
      nP   = no penultimate hop POP
      E    = Explicit-Null
      V    = Prefix-SID carries a value
      L    = value/index has local significance
=====
```

After the Prefix Node-SID has been correctly advertised by PE-5 with algorithm 128, it is possible to use the tunnel table to verify the Flex-Algorithm path toward the destination prefix. The following output shows the tunnel table at PE-1 for PE-5 (192.0.2.5/32). In this output, there are two entries. The first entry (tunnel ID 524296) is the default SR-ISIS tunnel calculated using algorithm 0. This has a next-hop of PE-3 (192.168.13.2) and metric of 200 representing the IGP cost of the path PE-1-PE-3-PE-5. The second entry (tunnel ID 524198) is calculated using Flex-Algorithm 128. It has a next-hop of PE-2 (192.168.12.2) and a metric of 40000 representing the accumulative delay metric (40msec) for the path PE-1-PE-2-PE-4-PE-6-PE-5.

```
[ ]
A:admin@PE-1# show router tunnel-table 192.0.2.5/32 protocol isis

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.5/32 [L]  isis (0)  MPLS  524296   11   192.168.13.2  200
192.0.2.5/32 [L]  isis (0)  MPLS  524298   11   192.168.12.2  40000
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
```

E = Inactive best-external BGP route
k = RIB-API or Forwarding Policy backup hop
=====

Traffic steering using Flex-Algorithm

To statically steer traffic into a Flex-Algorithm LSP, the **static-route-entry** allows for indirect next-hops to configure a Flex-Algorithm identifier in addition to a resolution-filter specifying SR-ISIS. This uses the specified Flex-Algorithm to construct a tunnel toward the indirect next-hop. In the following example, a static-route for prefix 172.16.0.1/32 is configured at PE-1 toward PE-5 (192.0.2.5) using a resolution-filter of SR-ISIS and flex-algo 128. Note that if no tunnel exists in the tunnel table for the referenced Flex-Algorithm identifier that the static-route will not become active.

```
# on PE-1:
configure {
  router "Base" {
    static-routes {
      route 172.16.0.1/32 route-type unicast {
        indirect 192.0.2.5 {
          admin-state enable
          tunnel-next-hop {
            resolution filter
            flex-algo 128
            resolution-filter {
              sr-isis true
            }
          }
        }
      }
    }
  }
}
```

From the prefix in the route-table, the next-hop is PE-5 (192.0.2.5) and the next-hop is resolved to an SR-ISIS tunnel with tunnel ID 524298, which is the tunnel ID of the Flex-Algorithm LSP.

```
[ ]
A:admin@PE-1# show router route-table 172.16.0.1/32

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type   Proto   Age           Pref
  Next Hop[Interface Name]                       Metric
-----
172.16.0.1/32                                     Remote Static 01h29m00s 5
  192.0.2.5 (tunneled:SR-ISIS:524298)             1
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

Flex-Algorithm LSPs can also be used for BGP next-hop resolution and/or service next-hop resolution wherever an SR-TE or SR Policy path contains one or more Prefix Node-SIDs and the auto-bind-tunnel resolution-filter is configured appropriately. The following output provides an example of an SR-TE LSP configured at PE-1 toward PE-5. The LSP references a primary path named "FlexAlgo-128", where that

path has a single hop containing the label 54405. This is the label value that was previously allocated to Flex-Algorithm 128 at PE-5.

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      path "FlexAlgo-128" {
        admin-state enable
        hop 1 {
          sid-label 54405
        }
      }
    }
    lsp "PE-1-PE-5-SR-TE-FlexAlgo128" {
      admin-state enable
      type p2p-sr-te
      to 192.0.2.5
      max-sr-labels {
        label-stack-size 1
        additional-frr-labels 2
      }
      primary "FlexAlgo-128" {
      }
    }
  }
}
```

First, verification is obtained that the SR-TE LSP is administratively and operationally up.

```
[ ]
A:admin@PE-1# show router mpls sr-te-lsp "PE-1-PE-5-SR-TE-FlexAlgo128"

=====
MPLS SR-TE LSPs (Originating)
=====
LSP Name          Tun   Protect   Adm  Opr
  To              Id      Path
-----
PE-1-PE-5-SR-TE-FlexAlgo128  1     N/A      Up   Up
  192.0.2.5
-----
LSPs : 1
=====
```

The same SR-TE LSP is also present in the tunnel table with tunnel ID 655362.

```
[ ]
A:admin@PE-1# show router tunnel-table 192.0.2.5/32 protocol sr-te

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner    Encap TunnelId Pref  Nexthop      Metric
  Color
-----
192.0.2.5/32     sr-te   MPLS  655362   8    192.0.2.5    16777215
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
```

A VPRN is configured with PE-1 and PE-5 as members. At PE-1, the **auto-bind-tunnel** context has the **resolution-filter** set to SR-TE such that it can use the SR-TE LSP containing the Flex-Algorithm Prefix Node-SID for next-hop resolution.

```
# on PE-1:
configure {
  service {
    vprn "VPRN 1" {
      admin-state enable
      service-id 1
      customer "1"
      route-distinguisher "64496:1"
      vrf-target {
        community "target:64496:1"
      }
      auto-bind-tunnel {
        resolution filter
        resolution-filter {
          sr-te true
        }
      }
    }
  }
}
```

VPN-IPv4 prefix 172.31.5.0/24 is advertised by PE-5 with the relevant Route-Target value such that it is imported at PE-1. The following output shows that prefix 172.31.5.0/24 is imported to the VPRN and uses an SR-TE LSP with tunnel ID 655362 in order to resolve the next-hop.

```
[ ]
A:admin@PE-1# show router 1 route-table 172.31.5.0/24

=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]                Metric
-----
172.31.5.0/24                    Remote BGP VPN  00h22m08s  170
  192.0.2.5 (tunneled:SR-TE:655362)      16777215
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```

Flex-Algorithm with admin group constraint

The configuration example used so far in this chapter employed a metric-type of delay. For completeness, the following section contains a second example using admin groups as a constraint.

When admin groups are used as a constraint, the first step is to apply the required admin groups to the relevant links. For the purpose of this example, the link PE-1-PE-3 is associated with admin group "blue". Initially, the admin group is configured as an if-attribute in the base router context and assigned an integer value in the range 0 to 31. The configuration of the admin group "blue" is as follows:

```
# on all PEs:
configure {
```

```

routing-options {
  if-attribute {
    admin-group "blue" {
      value 10
    }
  }
}

```

The admin group is then assigned to each required interface as an if-attribute in the same way that delay was previously configured. The following output is taken from PE-1 with a similar configuration applied at PE-3.

```

# on PE-1:
configure {
  router "Base"{
    interface "int-PE-1-PE-3" {
      if-attribute {
        admin-group ["blue"]
        delay {
          static 100000
        }
      }
    }
  }
}

```



Note:

draft-ietf-isis-te-app permits the use of admin groups and Extended Admin Groups (EAGs). Admin groups (RFC 5305) contain a 4-octet bit mask, where each set bit corresponds to a single admin group, allowing for support of 32 admin groups. EAGs (RFC 7308) have no fixed limit. SR OS only supports advertisement of admin groups, not EAGs. For backward compatibility, if EAGs are used by another vendor they must use only the first 32 colors in the EAG.

The next step is to configure the FAD and participation. First, the FAD is configured to exclude the admin group "blue"; the metric-type remains the default IGP metric. The following configuration is applied at PE-1 and PE-5.

```

# on PE-1, PE-5:
configure {
  routing-options {
    flexible-algorithm-definitions {
      flex-algo "FlexAlgo-129" {
        admin-state enable
        description "FlexAlgo-129-AG-Blue"
        exclude {
          admin-group "blue" { }
        }
      }
    }
  }
}

```

In addition to the exclude admin-group constraint, there are options for include-any and include-all admin-groups:

- Include-any means that any link not configured with any of the specified admin-groups will be pruned.
- Include-all means that any link not configured with all of the specified admin-groups will be pruned.

The following step is to advertise the FAD and indicate the participation in the Flex-Algorithm. Again, the following configuration is applied at PE-1 and PE-5, who both participate and advertise in Flex-Algorithm 129. Similar configuration is applied to the other routers in the example topology, but without the **advertise**

command because they only have a requirement to participate in the Flex-Algorithm and not advertise its definition.

```
# on PE-1,PE-5:
configure {
  router "Base" {
    isis 0 {
      flexible-algorithms {
        flex-algo 129 {
          participate true
          advertise "FlexAlgo-129"
          loopfree-alternate { }
        }
      }
    }
  }
}
```

Finally, a Prefix Node-SID is assigned to the Flex-Algorithm at the egress nodes. In the following example, PE-5 is the egress router and label 54415 is assigned to Flex-Algorithm 129.

```
# on PE-5:
configure {
  router "Base" {
    isis 0 {
      interface "system" {
        passive true
        level-capability 2
        ipv4-node-sid {
          label 50405
        }
      }
      flex-algo 129 {
        ipv4-node-sid {
          label 54415
        }
      }
    }
  }
}
```

The Prefix SID and associated Flex-Algorithm advertised by PE-5 is learned at PE-1. As before, the SID 4415 is advertised as an index and, with an SRGB of {50000-54999}, represents label 54415.

```
[]
A:admin@PE-1# show router isis prefix-sids algo 129

=====
Rtr Base ISIS Instance 0 Prefix/SID Table
=====
Prefix                               SID           Lvl/Typ      SRMS  AdvRtr
                               SID           Algo         MT    Flags
-----
192.0.2.5/32                         4415         2/Int.      N     PE-5
                               129         0           0     NnP
-----
No. of Prefix/SIDs: 1 (1 unique)
-----
SRMS : Y/N = prefix SID advertised by SR Mapping Server (Y) or not (N)
      S   = SRMS prefix SID is selected to be programmed
Flags: R   = Re-advertisement
      N   = Node-SID
      nP  = no penultimate hop POP
      E   = Explicit-Null
      V   = Prefix-SID carries a value
      L   = value/index has local significance
=====
```

The tunnel table is also used to verify the Flex-Algorithm path toward the destination prefix. The following output shows the tunnel table at PE-1 for PE-5 (192.0.2.5/32). In this output, there are two entries. The first entry (tunnel ID 524296) is the default SR-ISIS tunnel calculated using algorithm 0. This has a next-hop of PE-3 (192.168.13.2) and metric of 200 representing the IGP cost of the path PE-1-PE-3-PE-5. The second entry (tunnel ID 524299) is calculated using Flex-Algorithm 129. It has a next-hop of PE-2 (192.168.12.2), avoiding the PE-1-PE-3 link, and a metric of 400 representing the IGP metric for the path PE-1-PE-2-PE-4-PE-6-PE-5.

```
[ ]
A:admin@PE-1# show router tunnel-table 192.0.2.5/32 protocol isis

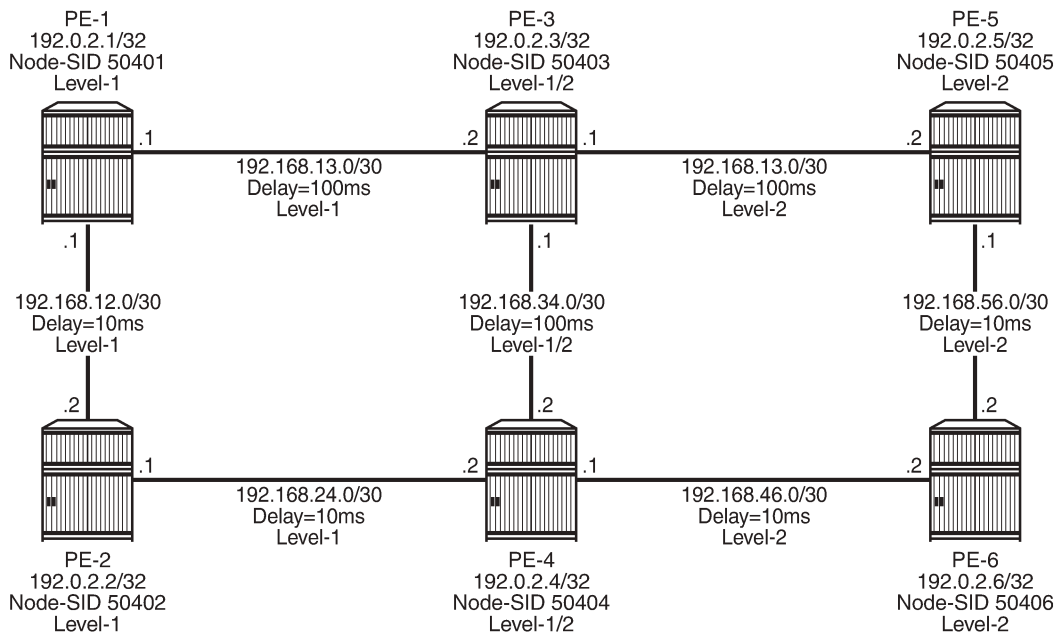
=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner      Encap TunnelId Pref  Nexthop      Metric
  Color
-----
192.0.2.5/32 [L]      isis (0)  MPLS  524296   11    192.168.13.2  200
192.0.2.5/32         isis (0)  MPLS  524299   11    192.168.12.2  400
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====
```

The Prefix Node-SID for Flex-Algorithm 129 is now available for carrying traffic. Methods for traffic steering into Flex-Algorithm LSPs have previously been described in this chapter and are therefore not repeated here.

Inter-area Flex-Algorithm

To validate the use of Flex-Algorithm in an inter-area environment, the example topology in [Figure 153: Example topology with modified IS-IS Level-1/2 capabilities](#) is modified such that PE-1 and PE-2 are IS-IS Level-1 routers, while PE-3 and PE-4 are IS-IS Level-1/2 routers. PE-5 and PE-6 remain Level-2 only routers.

Figure 153: Example topology with modified IS-IS Level-1/2 capabilities



36647

The previously configured Flex-Algorithm 128 (metric-type of delay) and Flex-Algorithm 129 (exclude admin-group blue) are again used, to show inter-area Flex-Algorithm path computations. Because PE-1 and PE-5 both advertise the FADs for these algorithms, this Level-1/2 inter-area scenario is affected because FAD sub-TLVs only have area scope; they are not redistributed between areas. In this scenario, PE-1 advertises the FAD within the Level-1 area, while PE-5 advertises the FAD within the Level-2 area.

As previously described, when a FAD includes the M-flag (Prefix Metric), an L1/L2 router (or ASBR) must include the FAPM sub-TLV when advertising a prefix within an Extended IP Reachability TLV between areas, levels, or domains. The advertised metric needs to be equal to the metric to reach the prefix for a Flex-Algorithm in the source area or domain. This allows a router in a different area/level/domain to include the FAPM when calculating prefix reachability for inter-area/domain prefixes and provides an optimal end-to-end path for a specific Flex-Algorithm.

In the example topology, both PE-5 and PE-6 are assigned Node-SID labels for Flex-Algorithms 128 and 129. PE-5 is assigned label 54405 for algorithm 128 and 54415 for algorithm 129, while PE-6 is assigned label 54406 for algorithm 128 and 54416 for algorithm 129.

The following output shows the PE-3 IS-IS Level-1 LSP as received by PE-1, truncated to show only the Extended IP Reachability TLV for PE-5 (192.0.2.5) and PE-6 (192.0.2.6). Each of these two prefixes has a Prefix SID sub-TLV for algorithm 0, algorithm 128, and algorithm 129. Flex-Algorithms 128 and 129 also have a FAPM sub-TLV containing the relevant metric for PE-3 to reach the destination prefix.

```
[ ]
A:admin@PE-1# show router isis database PE-3.00-00 detail
```

```
=====
Rtr Base ISIS Instance 0 Database (detail)
=====
```

```
Displaying Level 1 database
-----
```



```

LSP ID   : PE-3.00-00                               Level   : L1
Sequence : 0x58                                       Checksum : 0x4ec6   Lifetime : 54000
Version  : 1                                           Pkt Type : 18       Pkt Ver  : 1
Attributes: L1L2                                       Max Area : 3        Alloc Len : 482
SYS ID   : 1920.0000.2003                             SysID Len : 6        Used Len  : 482

TLVs :
---snip---
TE IP Reach :
  Default Metric : 100
  Control Info: D S, prefLen 32
  Prefix : 192.0.2.5
  Sub TLV :
    Prefix-SID Index:405, Algo:0, Flags:RNnP
    Prefix-SID Index:4405, Algo:128, Flags:RNnP
    Prefix-Metric-FlexAlg Algo:128, Metric:100000
    Prefix-SID Index:4415, Algo:129, Flags:RNnP
    Prefix-Metric-FlexAlg Algo:129, Metric:100
  Default Metric : 200
  Control Info: D S, prefLen 32
  Prefix : 192.0.2.6
  Sub TLV :
    Prefix-SID Index:406, Algo:0, Flags:RNnP
    Prefix-SID Index:4406, Algo:128, Flags:RNnP
    Prefix-Metric-FlexAlg Algo:128, Metric:110000
    Prefix-SID Index:4416, Algo:129, Flags:RNnP
    Prefix-Metric-FlexAlg Algo:129, Metric:200
---snip---

```

The information from the FAPM sub-TLV advertised by PE-3 and PE-4 (Level-1/2 routers) allows the routers in the Level-1 area to construct optimal end-to-end paths with accumulative metrics. The following output shows the tunnel table at PE-1 for PE-5 (192.0.2.5). The first entry is the SR-ISIS LSP calculated with algorithm 0 and showing an IGP metric of 200 for the path PE-1-PE-3-PE-5. The second entry is the SR-ISIS LSP calculated with Flex-Algorithm 129, which excludes the PE-1-PE-3 link, and therefore has an IGP metric of 400 for the path PE-1-PE-2-PE-4-PE-6-PE-5. The final entry is the SR-ISIS LSP calculated with Flex-Algorithm 128 using a metric-type of delay. This entry has a metric of 40000 representing the delay metric for the path PE-1-PE-2-PE-4-PE-6-PE-5.

```

[]
A:admin@PE-1# show router tunnel-table 192.0.2.5/32 protocol isis

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.5/32 [L]  isis (0)  MPLS  524414   11    192.168.13.2  200
192.0.2.5/32     isis (0)  MPLS  524418   11    192.168.12.2  400
192.0.2.5/32 [L]  isis (0)  MPLS  524416   11    192.168.12.2 40000
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====

```

Therefore, the optimal end-to-end paths can be calculated by redistributing prefixes with the FAPM sub-TLV and including that metric in the calculation for inter-area/domain prefixes.

Conclusion

With extensions to IS-IS, Flex-Algorithm provides a way to achieve a level of traffic engineering without the requirement for a centralized controller, and without the requirement to impose a deep label stack to represent the path; a single Node-SID is all that is required. Although the traffic engineering capabilities of Flex-Algorithm are limited compared to those available when using a centralized controller, it represents a reasonable trade-off between objective and complexity.

IS-IS Link Bundling

This chapter provides information about IS-IS link bundling.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

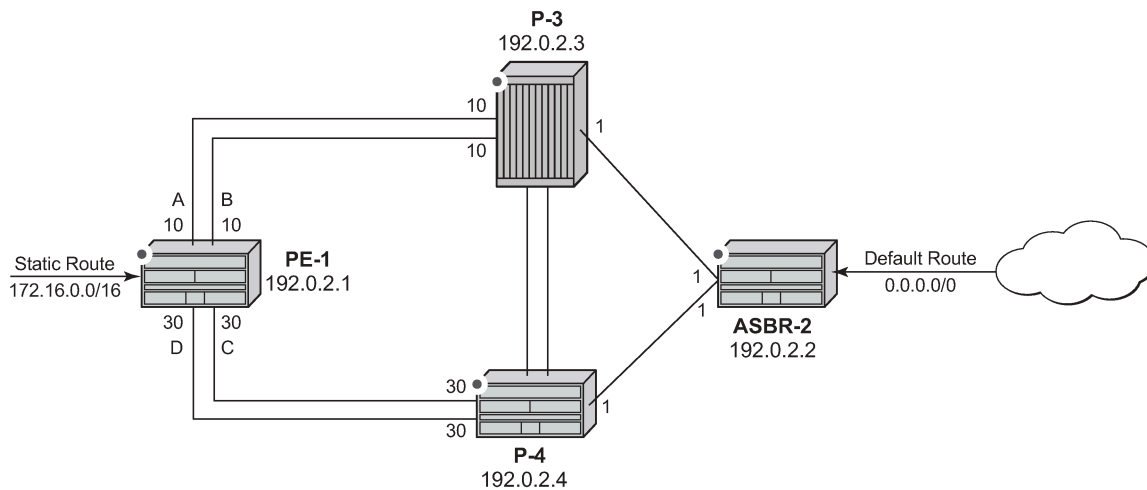
The chapter was initially written for SR OS Release 11.0.R6. The configuration in the current edition is based on MD-CLI in SR OS Release 20.7.R2.

Overview

Intermediate System to Intermediate System (IS-IS) Link Bundling allows for the grouping of a number of IS-IS interfaces into a single virtual link, called an IS-IS link group. It is used in conjunction with Equal Cost Multipath (ECMP) to dynamically change the metric of parallel IS-IS links if one or more links fail or suffer some sort of performance degradation.

Consider the network in [Figure 154: Link bundle schematic](#), where a Provider Edge router PE-1 connects to a core network comprised of two Provider (P) routers and a single Autonomous System Border Router (ASBR).

Figure 154: Link bundle schematic



al_0557

The links between PE-1 and P-3, and PE-1 and P-4 are 10 Gigabit Ethernet links. The links between ASBR-2 and P-3 and P-4 are both 100 Gigabit links. The link metrics are as shown in [Figure 154: Link bundle schematic](#).

In order to maximize the use of link bandwidth, ECMP is enabled on all routers and set to a value of 2, so that IP traffic flowing between PE1 and P-3, and PE-1 and P-4, is load balanced across the two links.

A default route is injected into the ASBR-2 router and redistributed via a policy statement into IS-IS, so that traffic flowing from PE-1 to the ASBR is resolved by this route. Traffic flows between PE-1 and ASBR-2, using the path with the lowest IS-IS metric, via P-3 with a metric of 11. The second path PE-1 to ASBR-2 via P-4 has the same bandwidth, but a higher IS-IS metric of 31.

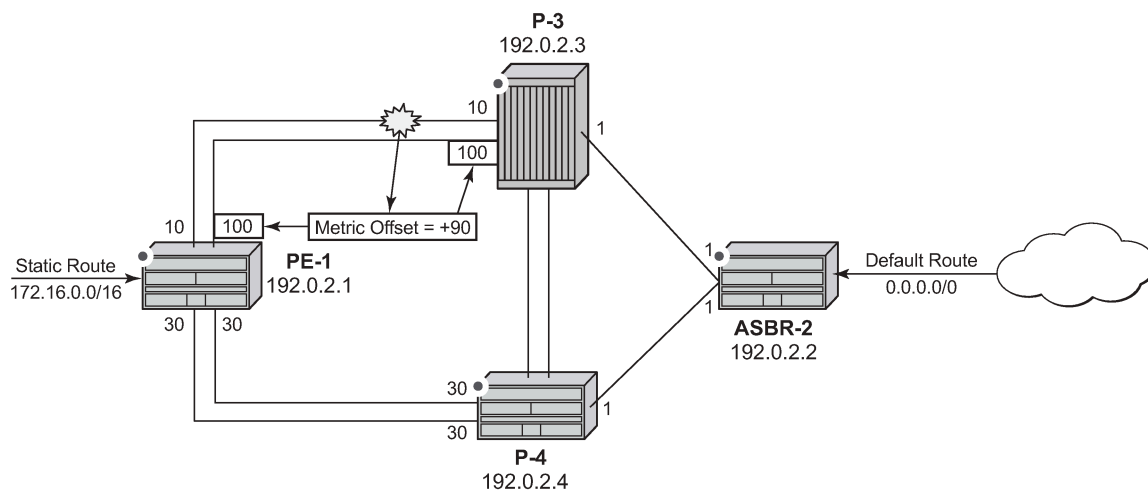
Traffic in the reverse direction flows toward a user subnet described by a static route configured on PE-1, which is redistributed into IS-IS using a policy statement. Once again, the shortest path between ASBR-2 and PE-1 is via P-3, so the bi-directional traffic flow is symmetric.

If one of the links between PE-1 and P-3 fails, traffic still flows via P-3, because the IS-IS metric is unchanged, but this now has less bandwidth than the second path via P-4. It is desirable to make use of the additional bandwidth of the second path, but this requires a change in metric. This can be achieved using IS-IS link bundling.

IS-IS link bundling allows for the creation of a group of IS-IS links, where the failure of a member link allows the metric of the remaining members of the link group to be increased by an offset value.

Using [Figure 155: Effect of single link failure on bundle group](#) as an example, the links between PE-1 and P-3 are included in a bundle group.

Figure 155: Effect of single link failure on bundle group



al_0558

To illustrate the change in metrics, a default static route is configured on ASBR-2 and redistributed into IS-IS, and the path to this route is monitored at PE-1. Similarly, a static route to subnet 172.16.0.0/16 is configured on PE-1 and redistributed into IS-IS and viewed on ASBR-2.

Should one of the links between PE-1 and P-3 fail, the metric of the remaining members can be increased by an offset, for example 90, so that the metric of the remaining link becomes $10 + 90 = 100$. The IS-IS metric between PE-1 and ASBR-2 via P-3 is now 101. The metric offset is applied to each remaining IS-IS interface individually and is advertised within the IS-IS database as the default cost in the TE-IS neighbors Type Length Variable (TLV).

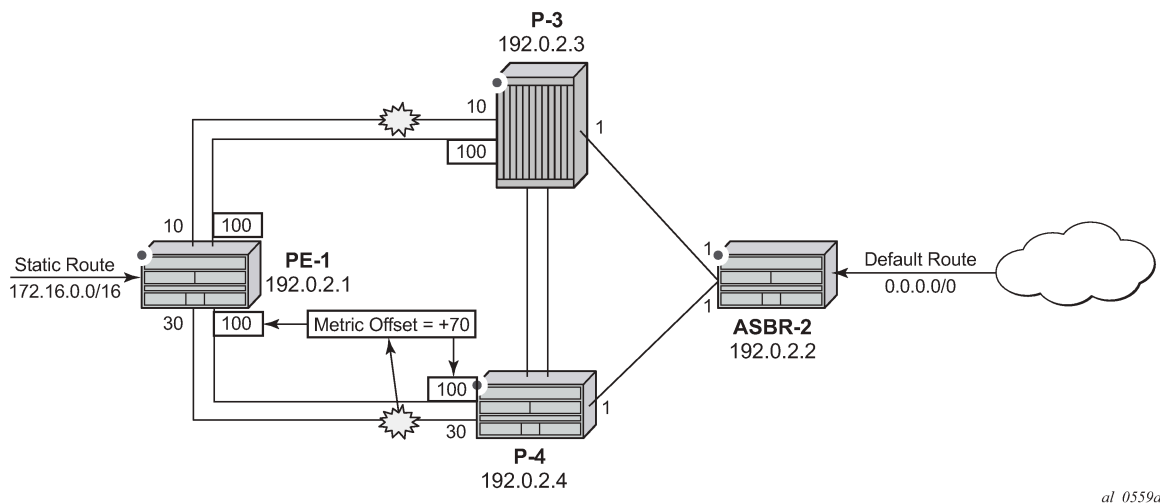
The path between PE-1 and ASBR-2 via P-4 now has the lowest IS-IS metric, and any affected routers within the IS-IS area will try and reroute the traffic based on the new metric.

The fundamentals of this feature are:

- The treatment of all member links in a link group bundle as a single virtual interface.
- The increase in metric by a specific offset value of each remaining individual link within the group when a failure of one or more links occurs.
 - The application of the offset occurs when the number of active links drops below a configured threshold.
 - The offset is removed when the number of active links within the link group bundle reaches the configured reversion threshold.
 - A link bundle is required on a router for the thresholds and offsets to apply.

Consider a second and subsequent failure where a link between PE-1 and P-4 also fails, so that there is only one active IS-IS interface between PE-1 and each of its neighboring P routers. This is shown in [Figure 156: Double link failure](#).

Figure 156: Double link failure



al_0559a

In this case, the metric for the remaining link between PE-1 and P-4 can be increased by an offset value of +70 so that the IS-IS metric PE-1 to P-4 becomes 100, the same as that between PE-1 and P-3 when a link has failed.

PE-1 now sees two equal cost paths to the default route—one via P-3 and one via P-4, so there are still two 10Gigabit Ethernet links across which the traffic can be load shared.

This can be summarized using the following table, where ABCD are the 4 links as per [Figure 154: Link bundle schematic](#) and link status is Up (U) or Down (D).

Table 3: Status of the links A, B, C, and D

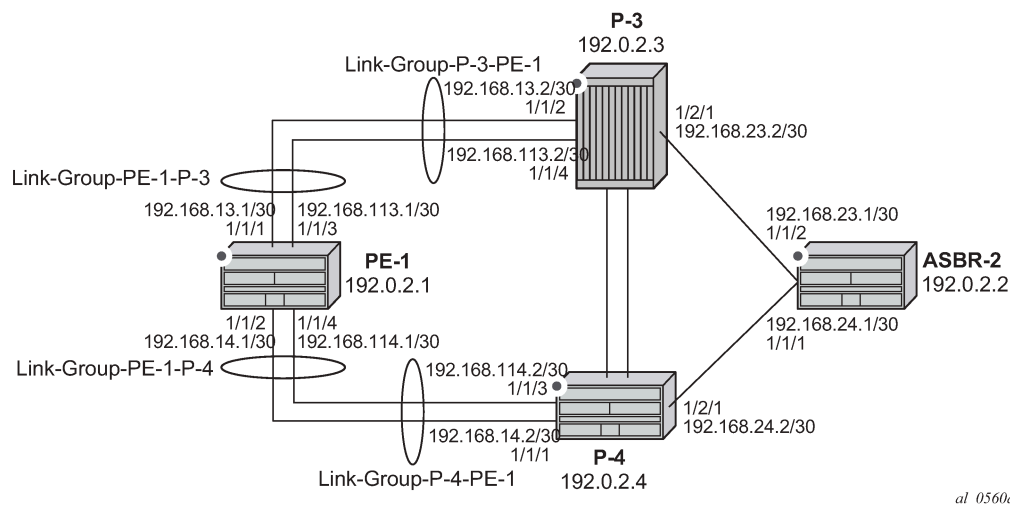
ABCD Status	A (metric,status)	B (metric,status)	C (metric,status)	D (metric,status)
UUUU	10 Transmit	10 Transmit	30 Idle	30 Idle

ABCD Status	A (metric,status)	B (metric,status)	C (metric,status)	D (metric,status)
UDUU	100 Idle	Down	30 Transmit	30 Transmit
UDUD	100 Transmit	Down	100 Transmit	Down
UUUD	10 Transmit	10 Transmit	100 Idle	Down

Configuration

The example topology is shown in [Figure 157: Example topology](#).

Figure 157: Example topology



On PE-1, ECMP is set to a value of 2 and the following router interfaces are configured:

```
# on PE-1:
configure {
  router "Base" {
    ecmp 2
    interface "int-PE-1-P-3-1" {
      port 1/1/1
      ipv4 {
        primary {
          address 192.168.13.1
          prefix-length 30
        }
      }
    }
    interface "int-PE-1-P-3-2" {
      port 1/1/3
      ipv4 {
        primary {
          address 192.168.113.1
          prefix-length 30
        }
      }
    }
  }
}
```

```

    }
  }
}
interface "int-PE-1-P-4-1" {
  port 1/1/2
  ipv4 {
    primary {
      address 192.168.14.1
      prefix-length 30
    }
  }
}
interface "int-PE-1-P-4-2" {
  port 1/1/4
  ipv4 {
    primary {
      address 192.168.114.1
      prefix-length 30
    }
  }
}
interface "system" {
  ipv4 {
    primary {
      address 192.0.2.1
      prefix-length 32
    }
  }
}
}

```

The IP router configuration for the remaining routers can be derived from [Figure 157: Example topology](#).

The IS-IS network is a level 1 network.

The IS-IS configuration for PE-1, including the interface metrics is as follows:

```

# on PE-1:
configure {
  router "Base" {
    isis 0 {
      admin-state enable
      advertise-passive-only true
      level-capability 1
      area-address [49.0001]
      interface "int-PE-1-P-3-1" {
        interface-type point-to-point
        level 1 {
          metric 10
        }
      }
      interface "int-PE-1-P-3-2" {
        interface-type point-to-point
        level 1 {
          metric 10
        }
      }
      interface "int-PE-1-P-4-1" {
        interface-type point-to-point
        level 1 {
          metric 30
        }
      }
      interface "int-PE-1-P-4-2" {
        interface-type point-to-point

```

```

        level 1 {
            metric 30
        }
    }
    interface "system" {
        passive true
    }
    level 1 {
        wide-metrics-only true
    }
}

```

The IS-IS configuration for the remaining routers can be derived from [Figure 157: Example topology](#).

The following configuration is for the static route and export policy on ASBR-2. The configuration of the static route on PE-1 is similar.

```

# on ASBR-2:
configure {
    router "Base" {
        static-routes {
            route 0.0.0.0/0 route-type unicast {
                blackhole {
                    admin-state enable
                }
            }
        }
    }
}

```

```

# on PE-1, ASBR-2:
configure {
    policy-options {
        policy-statement "STATIC-ISIS" {
            entry 10 {
                from {
                    protocol {
                        name [static]
                    }
                }
                to {
                    level 1
                }
                action {
                    action-type accept
                    metric {
                        set "@igp@"
                    }
                }
            }
        }
    }
    router "Base" {
        isis 0 {
            export-policy ["STATIC-ISIS"]
        }
    }
}

```

Link group configuration

PE-1 contains 2 link groups. The first link group contains the IS-IS interfaces toward P-3. The second contains the interfaces toward P-4.

Each link-group is configured using a unique name, which is unique per router, and the IS-IS interface names are configured within the group as group members.

The metric offset value is the amount by which the IS-IS metric of active member links is increased when the number of links drops below a configured threshold.

The IS-IS link group configuration for PE-1 for the interfaces toward P-3 is as follows:

```
# on PE-1:
configure {
  router "Base" {
    isis 0 {
      link-group "Link-Group-PE-1-P-3" {
        level 1 {
          ipv4-unicast-metric-offset 90
          oper-members 2
          revert-members 2
          member "int-PE-1-P-3-1" { }
          member "int-PE-1-P-3-2" { }
        }
      }
    }
  }
}
```

Similarly, the IS-IS link group for PE-1 for the interfaces toward P-4 is:

```
link-group "Link-Group-PE-1-P-4" {
  level 1 {
    ipv4-unicast-metric-offset 70
    oper-members 2
    revert-members 2
    member "int-PE-1-P-4-1" { }
    member "int-PE-1-P-4-2" { }
  }
}
```

Within the link-group, two thresholds are configured:

- oper-members threshold
- revert-members threshold

If the number of operational links in the link-group drops below the **oper-members** value, then all interfaces associated with that IS-IS link group have their interface metric increased by the configured offset value. As a result, IS-IS then tries to reroute traffic over lower cost paths.

If the number of operational links in the link-group equals the **revert-members** threshold value, then all interfaces associated with that IS-IS link group have their interface metric decreased by the configured offset value.

In this configuration, there is a requirement to increase the metric of each interface within a link group when a single interface fails. This means that the oper-members value is set to 2. In normal working circumstances, when both interfaces are active, the metric used is the configured interface metric. This means that the revert-members value must also be set to 2.

It is not possible to set the oper-members threshold to a value higher than that of the revert-members.

For completeness, the IS-IS configuration the P-routers is as follows.

```
# on P-3:
configure {
  router "Base" {
    isis 0 {
      admin-state enable
    }
  }
}
```

```
advertise-passive-only true
level-capability 1
area-address [49.0001]
interface "int-P-3-ASBR-2" {
    interface-type point-to-point
    level 1 {
        metric 1
    }
}
interface "int-P-3-PE-1-1" {
    interface-type point-to-point
    level 1 {
        metric 10
    }
}
interface "int-P-3-PE-1-2" {
    interface-type point-to-point
    level 1 {
        metric 10
    }
}
interface "system" {
    passive true
}
level 1 {
    wide-metrics-only true
}
link-group "Link-Group-P-3-PE-1" {
    level 1 {
        ipv4-unicast-metric-offset 90
        oper-members 2
        revert-members 2
        member "int-P-3-PE-1-1" { }
        member "int-P-3-PE-1-2" { }
    }
}
```

```
# on PE-4:
configure {
    router "Base" {
        isis 0 {
            admin-state enable
            advertise-passive-only true
            level-capability 1
            area-address [49.0001]
            interface "int-P-4-ASBR-2" {
                interface-type point-to-point
                level 1 {
                    metric 1
                }
            }
            interface "int-P-4-PE-1-1" {
                interface-type point-to-point
                level 1 {
                    metric 30
                }
            }
            interface "int-P-4-PE-1-2" {
                interface-type point-to-point
                level 1 {
                    metric 30
                }
            }
        }
    }
}
```

```

interface "system" {
    passive true
}
level 1 {
    wide-metrics-only true
}
link-group "Link-Group-P-4-PE-1" {
    level 1 {
        ipv4-unicast-metric-offset 70
        oper-members 2
        revert-members 2
        member "int-P-4-PE-1-1" { }
        member "int-P-4-PE-1-2" { }
    }
}
}

```

An overview of all link groups can be shown using the following commands, in this case on node PE-1. The link group status on PE-1 is as follows:

```

[]
A:admin@PE-1# show router isis link-group-status
=====
Rtr Base ISIS Instance 0 Link-Group Status
=====
Link-group           Mbrs   Oper   Revert Active Level   State
                   Mbr    Mbr    Mbr   Mbr
-----
Link-Group-PE-1-P-3  2      2      2     2     L1    normal
Link-Group-PE-1-P-4  2      2      2     2     L1    normal
=====

```

The output for the individual link group members on PE-1 is as follows:

For "Link-Group-PE-1-P-3" at PE-1:

```

[]
A:admin@PE-1# show router isis link-group-member-status "Link-Group-PE-1-P-3"
=====
Rtr Base ISIS Instance 0 Link-Group Member
=====
Link-group           I/F name                Level   State
-----
Link-Group-PE-1-P-3  int-PE-1-P-3-1          L1     Up
Link-Group-PE-1-P-3  int-PE-1-P-3-2          L1     Up
-----
Legend: BER = bitErrorRate
=====

```

For "Link-Group-PE-1-P-4" at PE-1:

```

[]
A:admin@PE-1# show router isis link-group-member-status "Link-Group-PE-1-P-4"
=====
Rtr Base ISIS Instance 0 Link-Group Member
=====
Link-group           I/F name                Level   State
-----
Link-Group-PE-1-P-4  int-PE-1-P-4-1          L1     Up
-----

```

```
Link-Group-PE-1-P-4  int-PE-1-P-4-2          L1          Up
-----
Legend: BER = bitErrorRate
=====
```

For P-3, the link group status is as follows:

```
[ ]
A:admin@P-3# show router isis link-group-status

=====
Rtr Base ISIS Instance 0 Link-Group Status
=====
Link-group           Mbrs   Oper   Revert Active Level   State
                   Mbr    Mbr    Mbr    Mbr
-----
Link-Group-P-3-PE-1  2      2      2      2      L1    normal
=====
```

For P-3, the link group member status is as follows:

```
[ ]
A:admin@P-3# show router isis link-group-member-status "Link-Group-P-3-PE-1"

=====
Rtr Base ISIS Instance 0 Link-Group Member
=====
Link-group           I/F name           Level   State
-----
Link-Group-P-3-PE-1  int-P-3-PE-1-1    L1     Up
Link-Group-P-3-PE-1  int-P-3-PE-1-2    L1     Up
-----
Legend: BER = bitErrorRate
=====
```

Routing table PE-1

In a normal working state, the routing table for PE-1 contains the default route for forwarding traffic toward ASBR-2. Because ECMP is set to a value of 2, two entries are available with next-hops pointing toward P-3, as follows. The metric for each path is 11.

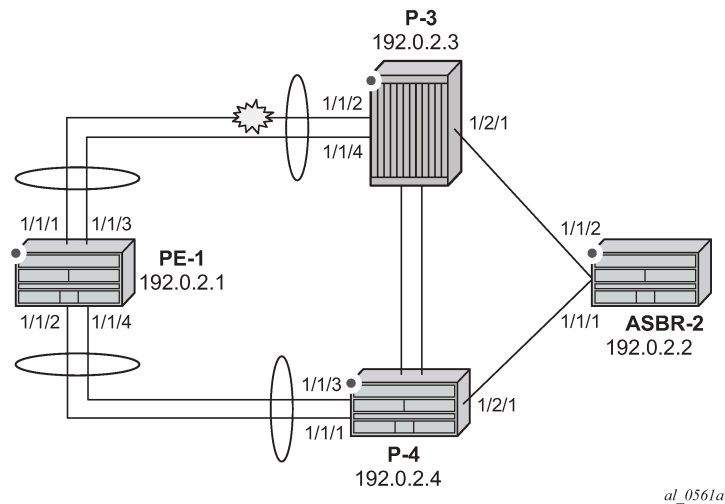
```
[ ]
A:admin@PE-1# show router route-table 0.0.0.0/0

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]           Type  Proto  Age           Pref
  Next Hop[Interface Name]           Metric
-----
0.0.0.0/0                    Remote ISIS  00h02m27s  15
      192.168.13.2              11
0.0.0.0/0                    Remote ISIS  00h02m27s  15
      192.168.113.2             11
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```

Failure of link member PE-1 to P-3

Figure 158: Link failure shows that one of the links in the active link group between PE-1 and P-3 fails.

Figure 158: Link failure



One of the links between PE-1 and P-3 is put into a failed state by disabling port 1/1/2 on P-3, as follows:

```
# on P-3:
configure {
  port 1/1/2 {
    admin-state disable
  }
}
```

The route-table on PE-1 shows that the metric for the default route prefix, 0.0.0.0/0, has increased from 11 to 31, and the next-hops are now interface addresses on P-4, as follows:

```
[ ]
A:admin@PE-1# show router route-table 0.0.0.0/0

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]                Metric
-----
0.0.0.0/0                          Remote  ISIS    00h01m14s    15
  192.168.14.2                        31
0.0.0.0/0                          Remote  ISIS    00h01m14s    15
  192.168.114.2                        31
-----

No. of Routes: 2
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```

The link-group status shows that the number of active members has fallen below the oper-members threshold and as a result, the metric offset has been applied.

```
[ ]
A:admin@PE-1# show router isis link-group-status

=====
Rtr Base ISIS Instance 0 Link-Group Status
=====
Link-group          Mbrs   Oper   Revert Active Level  State
                   Mbr    Mbr    Mbr    Mbr
-----
Link-Group-PE-1-P-3  2      2      2      1      L1      Offset-Applied
Link-Group-PE-1-P-4  2      2      2      2      L1      normal
=====
```

Finally, the status of an individual link group member is as follows:

```
[ ]
A:admin@PE-1# show router isis link-group-member-status "Link-Group-PE-1-P-3"

=====
Rtr Base ISIS Instance 0 Link-Group Member
=====
Link-group          I/F name          Level  State
-----
Link-Group-PE-1-P-3 int-PE-1-P-3-1    L1     If-Down
Link-Group-PE-1-P-3 int-PE-1-P-3-2    L1     Up
-----
Legend: BER = bitErrorRate
=====
```

The following IS-IS database on PE-1 shows that the link metric (TE-IS neighbor) toward P-3 has a metric of 100, comprised of the original metric of 10 plus the offset of 90.

```
[ ]
A:admin@PE-1# show router isis database PE-1 detail

=====
Rtr Base ISIS Instance 0 Database (detail)
=====

Displaying Level 1 database
-----
LSP ID   : PE-1.00-00          Level   : L1
Sequence : 0x7                Checksum : 0x3c96    Lifetime : 1099
Version  : 1                  Pkt Type  : 18       Pkt Ver  : 1
Attributes: L1                Max Area  : 3         Alloc Len : 1492
SYS ID   : 1920.0000.2001     SysID Len : 6         Used Len  : 163

TLVs :
  Area Addresses:
    Area Address : (3) 49.0001
  Supp Protocols:
    Protocols    : IPv4
  IS-Hostname   : PE-1
  Router ID    :
    Router ID    : 192.0.2.1
  I/F Addresses :
    I/F Address  : 192.0.2.1
    I/F Address  : 192.168.13.1
```

```

I/F Address : 192.168.14.1
I/F Address : 192.168.113.1
I/F Address : 192.168.114.1
TE IS Nbrs :
  Nbr : P-3.00
  Default Metric : 100
  Sub TLV Len : 12
  IF Addr : 192.168.113.1
  Nbr IP : 192.168.113.2
TE IS Nbrs :
  Nbr : P-4.00
  Default Metric : 30
  Sub TLV Len : 12
  IF Addr : 192.168.14.1
  Nbr IP : 192.168.14.2
TE IS Nbrs :
  Nbr : P-4.00
  Default Metric : 30
  Sub TLV Len : 12
  IF Addr : 192.168.114.1
  Nbr IP : 192.168.114.2
TE IP Reach :
  Default Metric : 1
  Control Info: , prefLen 16
  Prefix : 172.16.0.0
  Default Metric : 0
  Control Info: , prefLen 32
  Prefix : 192.0.2.1

Level (1) LSP Count : 1

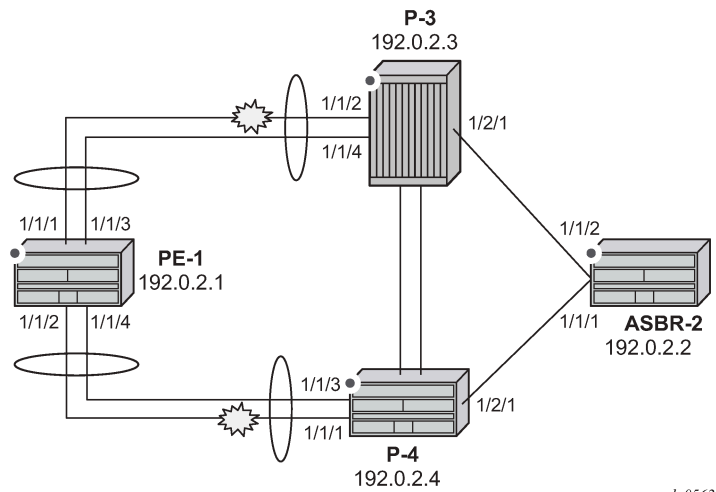
---snip---

```

Failure of link member PE-1 to P-4:

Figure 159: Second link failure shows that all link groups only have one active link instead of two.

Figure 159: Second link failure



If a link between PE-1 and P-4 now fails, simulated by disabling port 1/1/1 on P-4, then the metric offset is applied to the link groups on PE-1 and P-4 as the number of active links has dropped below the oper-members threshold for the link groups Link-Group-PE-1-P-4 on PE-1 and Link-Group-P-4-PE-1 on P-4.

```
# on P-4:
configure {
  port 1/1/1 {
    admin-state disable
  }
}
```

The routing table for PE-1 now shows that there are still two equal cost paths for the default route prefix advertised by ASBR-2, as follows:

```
[ ]
A:admin@PE-1# show router route-table 0.0.0.0/0

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type   Proto   Age           Pref
  Next Hop[Interface Name]                       Metric
-----
0.0.0.0/0                                         Remote  ISIS    00h01m16s    15
  192.168.113.2                                   101
0.0.0.0/0                                         Remote  ISIS    00h01m16s    15
  192.168.114.2                                   101
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

The metric for each routing table entry is 101, comprising of a cost of 100 for the PE-1 to P router link, where the link-group offset has been applied, and the cost of 1 for the P router to ASBR-2 router link.

By examining the IS-IS database on the PE-1 router, the updated metric for the link to neighbors P-3 and P-4 can be seen with the offset applied. These are seen in the "TE-IS Nbrs" TLV in the following output.

```
[ ]
A:admin@PE-1# show router isis database PE-1 detail

=====
Rtr Base ISIS Instance 0 Database
=====

Displaying Level 1 database
-----
LSP ID       : PE-1.00-00                               Level       : L1
Sequence     : 0x8                                       Checksum    : 0x3e0a   Lifetime    : 1129
Version      : 1                                       Pkt Type    : 18      Pkt Ver     : 1
Attributes   : L1                                       Max Area    : 3       Alloc Len   : 1492
SYS ID       : 1920.0000.2001                          SysID Len   : 6       Used Len    : 138

TLVs :
Area Addresses:
  Area Address : (3) 49.0001
Supp Protocols:
  Protocols    : IPv4
IS-Hostname    : PE-1
```



```
Router ID :
  Router ID : 192.0.2.1
I/F Addresses :
  I/F Address : 192.0.2.1
  I/F Address : 192.168.13.1
  I/F Address : 192.168.14.1
  I/F Address : 192.168.113.1
  I/F Address : 192.168.114.1
TE IS Nbrs :
  Nbr : P-3.00
  Default Metric : 100
  Sub TLV Len : 12
  IF Addr : 192.168.113.1
  Nbr IP : 192.168.113.2
TE IS Nbrs :
  Nbr : P-4.00
  Default Metric : 100
  Sub TLV Len : 12
  IF Addr : 192.168.114.1
  Nbr IP : 192.168.114.2
TE IP Reach :
  Default Metric : 1
  Control Info: , prefLen 16
  Prefix : 172.16.0.0
  Default Metric : 0
  Control Info: , prefLen 32
  Prefix : 192.0.2.1

Level (1) LSP Count : 1

---snip---
```

Conclusion

IS-IS link bundling allows service providers to configure multiple IS-IS interfaces as a single link group for ECMP purposes and allow link metric increases if an interface within the bundle group fails. This example provides the configuration for IS-IS link bundling, together with the associated commands and outputs which can be used for verifying and troubleshooting.

Next-Hop Resolution for Labeled BGP Routes

This chapter describes Next-Hop Resolution for Labeled BGP Routes.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 15.0.R7, but the CLI in the current edition is based on SR OS Release 22.10.R3.

Overview

BGP routes with the VPN-IPv4, VPN-IPv6, labeled IPv4, and labeled IPv6 address families are BGP routes whose Network Layer Reachability Information (NLRI) contains an MPLS label that is mapped to the route. BGP advertises labels that subsequently are used in the data plane for MPLS forwarding. BGP labeled routes are fundamental to IP VPN services, 6PE services, inter-AS connectivity, and seamless MPLS network segmentation. When a BGP speaker receives a BGP labeled route, it has the following options for resolving the next hop (NH) of the route:

- It can resolve the NH to an MPLS tunnel, such as an LDP or RSVP tunnel. In this case, the router pushes a transport label on top of the BGP label and allows the BGP labeled packet to be transported to the NH router over a set of intermediate routers that lack context for forwarding using the BGP label.
- It can resolve the NH to a local interface if the NH is an address on a local subnet. No additional labels need to be pushed onto the top of the label stack.
- It can resolve the NH using a static route and no additional label needs to be pushed. BGP NH resolution using a static route is useful in the following cases:
 - The static route has a blackhole NH in an intentional Remotely Triggered Blackhole (RTBH) scenario. Blackholed static routes are used for BGP NH resolution even when the configuration does not allow BGP NH resolution using static routes.
 - The static route has a NH address of a loopback interface of a directly connected peer. By default, this option is disabled.
- It can resolve the NH using the Longest Prefix Match (LPM) in the route table with static routes, OSPF, IS-IS, and RIP routes. This is applicable for route reflectors (RRs) that are not in the data path, so they do not need to have tunnels. By default, this option is disabled.

NH resolution of BGP routes using tunnels is the same for eBGP and iBGP routes, and for VPN IP routes and label-unicast routes. The common NH resolution logic uses the following routes in order of preference:

1. Local or direct routes
2. Non-default static routes

- Blackholed static routes
 - Non-blackholed non-default static routes, if allowed
3. Route Table Manager (RTM) routes (including static, OSPF, IS-IS, and RIP), if allowed—only for RRs
- When enabled, no routes are installed in the Forwarding Information Base (FIB) and no tunnels can be used.
4. Tunnels

NH resolution using a local (interface) or direct route

If possible, the BGP NH is resolved to a local interface route.

If the BGP NH is an IPv4-mapped IPv6 address in `::ffff:a.b.c.d` format, the system first tries to find a local route matching the IPv6 address. When no match is found, the system tries to find a local route matching the extracted IPv4 address `a.b.c.d`.

NH resolution using a non-default static route

If the BGP NH is an IPv4 address, the system looks for the non-default IPv4 static route that is the LPM of the address.

- If the LPM static route is blackholed, this static route is used, regardless of the **allow-static** command configuration.
- If the LPM static route is not blackholed, the static route is only used when the **allow-static true** command is configured.

If the BGP NH is an IPv4-mapped IPv6 address in the `::ffff:a.b.c.d` format, the system first tries to find the non-default static route that is the LPM of the full IPv6 address.

If no matching IPv6 static route is found, the system tries to find the non-default IPv4 route that is the LPM of the extracted IPv4 address `a.b.c.d`.

NH resolution using any type of route in the RTM—only on RR

This is only applicable for RRs that are not in the data path and configured with the **rr-use-route-table** and **route-table-install** commands. The considered routes in the RTM can be static, OSPF, IS-IS, or RIP.

If the BGP NH is an IPv4 address, the system searches the IPv4 RTM route that is the LPM of the address.

If the BGP NH is an IPv4-mapped IPv6 address in `::ffff:a.b.c.d` format, the system first searches for the IPv6 route that is the LPM of the full IPv6 address. If no match is found, the system searches for an RTM route matching the extracted IPv4 address `a.b.c.d`.

NH resolution using a tunnel in the Tunnel Table Manager (TTM)

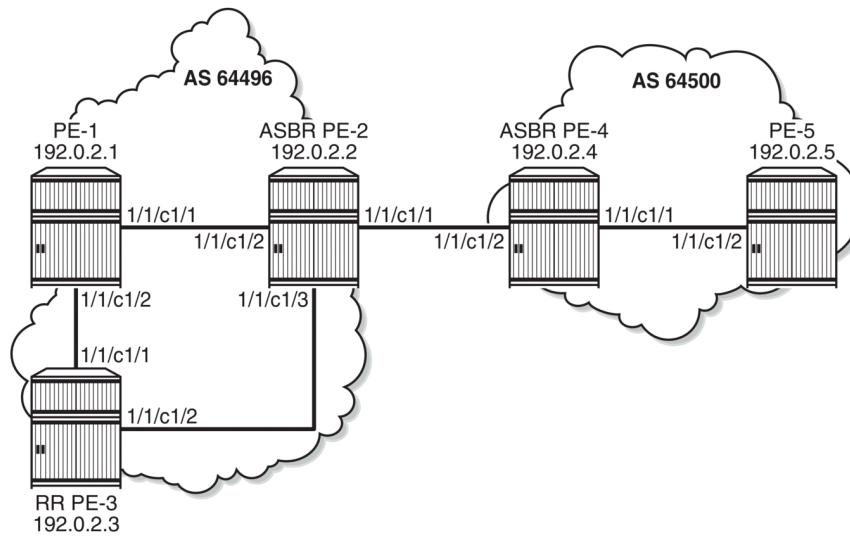
If the BGP NH is an IPv4 address, the TTM selects the tunnel table entry that matches the address prefix with the lowest preference and allowed by the applicable resolution filter. If the preference is the same, the tunnel table entry with the best metric is chosen, and so on.

If the BGP NH is an IPv4-mapped IPv6 address in `::ffff:a.b.c.d` format, the system searches the most preferred TTM tunnel matching the extracted IPv4 address `a.b.c.d` that is allowed by the applicable resolution filter.

Configuration

Figure 160: Example topology shows the example topology with three routers in AS 64496 and two routers in AS 64500.

Figure 160: Example topology



38424

The initial configuration includes the following:

- Cards, MDAs, ports
- Router interfaces between the PEs
- IS-IS as IGP between the PEs within an AS, not between ASBRs PE-2 and PE-4
- LDP between the PE-1 and PE-2 in AS 64496 (not to the RR PE-3) and between PE-4 and PE-5 in AS 64500

The following scenarios are configured in the following sections:

- [NH resolution for labeled IPv4 routes](#)
- [NH resolution for iBGP VPN-IPv4/v6 routes](#)
- [NH resolution for inter-AS VPRN model B](#)
- [NH resolution for inter-AS VPRN model C](#)

NH resolution for labeled IPv4 routes

In the [NH resolution for inter-AS VPRN model C](#) section, inter-AS VPRNs are configured, as described in the *VPRN Inter-AS VPRN Model C* chapter. Within each AS, the PEs advertise their system addresses (192.0.2.x) as labeled IPv4 routes. The configuration of the export policy is as follows:

```
# on all PEs:
configure {
  policy-options {
    prefix-list "PE-sys" {
      prefix 192.0.2.0/28 type range {
        start-length 32
        end-length 32
      }
    }
  }
  policy-statement "export-bgp" {
    entry 10 {
      from {
        prefix-list ["PE-sys"]
        protocol {
          name [direct]
        }
      }
      to {
        protocol {
          name [bgp-label]
        }
      }
      action {
        action-type accept
      }
    }
  }
}
}
```

Within each AS, BGP group "iBGP" is configured for the VPN-IPv4, VPN-IPv6, and label-IPv4 address families. In AS 64496, PE-3 is configured as RR. The initial BGP configuration on PE-3 is as follows:

```
# on PE-3:
configure {
  router "Base" {
    bgp {
      split-horizon true
      group "iBGP" {
        peer-as 64496
        advertise-inactive true
        cluster {
          cluster-id 192.0.2.3
        }
      }
      neighbor "192.0.2.1" {
        group "iBGP"
        family {
          vpn-ipv4 true
          vpn-ipv6 true
          label-ipv4 true
        }
      }
      neighbor "192.0.2.2" {
        group "iBGP"
        family {

```

```

        label-ipv4 true
    }
}
}
}
}

```

Between the Autonomous System Border Routers (ASBRs) PE-2 and PE-4, BGP is configured for the label-IPv4 address family only. The initial configuration for the eBGP peering uses the interface address of the remote ASBR (such as 192.168.24.2), which is the standard way for eBGP peering between ASBRs. However, for demonstration purposes, loopback addresses are configured later.

The BGP labeled routes for the system IP addresses are not used within an AS because IGP routes are preferred by the RTM, so they are inactive. However, BGP exports these inactive routes to the ASBR peer in the remote AS (**advertise-inactive true**) where these routes are used. The initial BGP configuration on PE-2 is as follows:

```

# on PE-2:
configure {
  router {
    bgp {
      split-horizon true
      group "iBGP" {
        peer-as 64496
        family {
          label-ipv4 true
        }
        advertise-inactive true
      }
      neighbor "192.0.2.3" {
        group "iBGP"
      }
      group "eBGP4_local" {
        family {
          label-ipv4 true
        }
        advertise-inactive true
      }
      neighbor "192.168.24.2" {
        group "eBGP4_local"
        peer-as 64500
      }
    }
  }
}

```

The default BGP NH resolution does not allow static routes and the only transport tunnel type that can be used for labeled IPv4 routes is LDP:

```

[/]
A:admin@PE-2# configure {
  router "Base" {
    bgp {
      next-hop-resolution
      labeled-routes
      info detail | match "allow-static"
      info detail | match "family label-ipv4" post-lines 17
    }
  }
}

[ex:/configure router "Base" bgp next-hop-resolution labeled-routes]
A:admin@PE-2# info detail | match "allow-static"
allow-static false

```

```
[ex:/configure router "Base" bgp next-hop-resolution labeled-routes]
A:admin@PE-2# info detail | match "family label-ipv4" post-lines 17
    family label-ipv4 {
    ---snip---
        resolution-filter {
            bgp false
            ldp true
        }
    ---snip---
```

Labeled IPv4 BGP NH resolved to local route

The route table on PE-2 shows that the route to 192.0.2.5 on PE-5 is a BGP labeled IPv4 route with NH 192.168.24.2:

```
[/]
A:admin@PE-2# show router route-table 192.0.2.5/32

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                                Type   Proto   Age      Pref
  Next Hop[Interface Name]                        Metric
-----
192.0.2.5/32                                     Remote BGP_LABEL 00h06m11s 170
  192.168.24.2                                     0
-----
No. of Routes: 1
---snip---
```

To verify that BGP NH resolution prefers local routes over static routes (if **allow-static true** is configured), the following is configured on the ASBRs. For the following static routes between PE-2 and PE-4, additional loopback addresses and a static route to the loopback address on the eBGP peer are configured. The configuration on ASBR PE-2 is as follows:

```
# on PE-2:
configure {
  router "Base" {
    interface "loopback" {
      loopback
      ipv4 {
        primary {
          address 10.0.0.2
          prefix-length 32
        }
      }
    }
    static-routes {
      route 10.0.0.4/32 route-type unicast {
        next-hop "192.168.24.2" {
          admin-state enable
        }
      }
    }
  }
}
```

On PE-2, the following additional eBGP group for the label IPv4 address family is configured and the BGP NH resolution for labeled routes is configured to allow static routes. The eBGP peer is only one hop away, so a **multihop** command is not required.

```
# on PE-2:
configure {
  router "Base" {
    bgp {
      next-hop-resolution {
        labeled-routes {
          allow-static true
        }
      }
      group "eBGP4_static" {
        admin-state enable
      }
      neighbor "10.0.0.4" {
        peer-as 64500
        group "eBGP4_static"
        family {
          label-ipv4 true
        }
        advertise-inactive true
        local-address 10.0.0.2
      }
    }
  }
}
```

Another static route is configured to the system IP address of the eBGP peer with preference 25 to ensure that this static route is not preferred over the preceding static route with default preference 5. LDP is enabled on the interface between the ASBRs, such as "int-PE-2-PE-4" on PE-2. This makes it possible to resolve the BGP NH to an LDP tunnel. Also, an additional BGP group is configured for the labeled IPv4 address family to the system IP address of the eBGP peer, such as 192.0.2.4. The configuration on PE-2 is as follows:

```
# on PE-2:
configure {
  router "Base" {
    bgp {
      group "eBGP4_tunnel" {
        admin-state enable
      }
      neighbor "192.0.2.4" {
        peer-as 64500
        group "eBGP4_tunnel"
        family {
          label-ipv4 true
        }
        advertise-inactive true
      }
    }
  }
  ldp {
    interface-parameters {
      interface "int-PE-2-PE-4" {
        ipv4 {
        }
      }
    }
  }
}
static-routes {
```



```

route 192.0.2.4/32 route-type unicast {
  next-hop "192.168.24.2" {
    admin-state enable
    preference 25
  }
}

```

This additional configuration does not result in a BGP NH resolution to an LDP tunnel, because the destination can also be reached via a static route, which is preferred. In the [Labeled IPv4 BGP NH resolved to tunneled route](#) section, the configuration is modified to exclude static routes from the NH resolution.

The following FIB on PE-2 shows that a labeled BGP route with resolved NH 192.168.24.2 is used for prefix 192.0.2.5/32. The BGP NH is not resolved to a tunnel.

```

[/]
A:admin@PE-2# show router fib 1 ip-prefix-prefix-length 192.0.2.5/32

=====
FIB Display
=====
Prefix [Flags]                                Protocol
NextHop
-----
192.0.2.5/32                                  BGP_LABEL
  192.168.24.2 (int-PE-2-PE-4)
-----
Total Entries : 1
=====

```

PE-2 has three labeled IPv4 BGP routes for prefix 192.0.2.5/32: the first route with local NH 192.168.24.2 (which is best and used), the second route with NH 10.0.0.4/32 (which can be reached via a static route), and the third route with NH 192.0.2.4 (which can be reached via a less preferred static route):

```

[/]
A:admin@PE-2# show router bgp routes 192.0.2.5/32 label-ipv4

=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
BGP LABEL-IPV4 Routes
=====
Flag  Network                                LocalPref  MED
      Nexthop (Router)                       Path-Id     IGP Cost
      As-Path                                Label
-----
u*>i  192.0.2.5/32                                None        None
      192.168.24.2                                None         0
      64500                                         524285
*i    192.0.2.5/32                                None        None
      10.0.0.4                                    None         1
      64500                                         524285

```

```
*i 192.0.2.5/32          None      None
   192.0.2.4           None      1
   64500                None      524285
-----
Routes : 3
=====
```

Table 4: Default preferences in route table shows the default preferences in a route table. These preferences are configurable, except for the direct attached routes, which always have preference 0.

Table 4: Default preferences in route table

Route type	Preference
Direct Attached	0
Static	5
OSPF Internal	10
IS-IS Level 1 Internal	15
IS-IS Level 2 Internal	18
RIP	100
OSPF External	150
IS-IS Level 1 External	160
IS-IS Level 2 External	165
BGP	170

The following shows the BGP NHs with the resolving prefix and the resolved NH. On PE-2, all three NHs of the labeled IPv4 routes for prefix 192.0.2.5/32 have resolved NH 192.168.24.2. NH 192.168.24.2 has owner local and preference 0; NH 10.0.0.4 has owner static and default preference 5; NH 192.0.2.4 has owner static and preference 25 by configuration.

```
[/]
[/]
A:admin@PE-2# show router bgp next-hop
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====

BGP Next Hop
=====
Next Hop                    Pref      Owner
Resolving Prefix           FibProg  Metric
Resolved Next Hop         Colored  Ref. Count
Admin-tag-policy           FlexAlgo Last Mod.
-----
10.0.0.4                    5         STATIC
10.0.0.4/32                 N         1
192.168.24.2                N         0
--                           --        00h04m53s
```

```

192.0.2.3          15      ISIS
  192.0.2.3/32    N       10
  192.168.23.2    N       0
  --             --      00h16m55s
192.0.2.4       25     STATIC
  192.0.2.4/32    N       1
  192.168.24.2  N       0
  --             --      00h04m15s
192.168.24.2   0       LOCAL
  192.168.24.0/30 N       0
  192.168.24.2 N       0
  --             --      00h16m55s
-----
Next Hops : 4
=====

```

Labeled IPv4 BGP NH resolved to non-default static route

When the BGP group "eBGP4_local" is disabled, the BGP NH can no longer be resolved to a local route. On the ASBRs PE-2 and PE-4, the following command disables the BGP group:

```

# on PE-2, PE-4:
configure {
  router "Base" {
    bgp {
      group "eBGP4_local" {
        admin-state disable
      }
    }
  }
}

```

The FIB on PE-2 shows that the route to prefix 192.0.2.5/32 is a labeled BGP route with resolved NH 192.168.24.2. This looks identical to the preceding output for the FIB when the BGP NH could be resolved to a local route, but in this case, the BGP NH is resolved to a non-default static route, as is shown later.

```

[/]
A:admin@PE-2# show router fib 1 ip-prefix-prefix-length 192.0.2.5/32
=====
FIB Display
=====
Prefix [Flags]                               Protocol
NextHop
-----
192.0.2.5/32                                 BGP_LABEL
  192.168.24.2 (int-PE-2-PE-4)
-----
Total Entries : 1
=====

```

PE-2 now has only two valid labeled IPv4 BGP routes instead of three: the best and used route has NH 10.0.0.4 and the less preferred route has NH 192.0.2.4:

```

[/]
A:admin@PE-2# show router bgp routes 192.0.2.5/32 label-ipv4
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496

```

```

=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====

BGP LABEL-IPV4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  192.0.2.5/32                            None       None
      10.0.0.4                               None       1
      64500                                             524285
*i    192.0.2.5/32                            None       None
      192.0.2.4                               None       1
      64500                                             524285
-----
Routes : 2
=====

```

On PE-2, NH 10.0.0.4 and NH 192.0.2.4 are both resolved to NH 192.168.24.2. NH 10.0.0.4 has preference 5, which is better than the configured preference 25 for NH 192.0.2.4.

```

[/]
A:admin@PE-2# show router bgp next-hop
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====

BGP Next Hop
=====
Next Hop                    Pref  Owner
Resolving Prefix           FibProg Metric
Resolved Next Hop         Colored Ref. Count
Admin-tag-policy          FlexAlgo Last Mod.
-----
10.0.0.4                    5    STATIC
  10.0.0.4/32                N    1
  192.168.24.2               N    0
  --                          --   00h17m17s
192.0.2.3                   15   ISIS
  192.0.2.3/32               N    10
  192.168.23.2               N    0
  --                          --   00h29m19s
192.0.2.4                   25   STATIC
  192.0.2.4/32               N    1
  192.168.24.2               N    0
  --                          --   00h16m39s
-----
Next Hops : 3
=====

```

When the preferred static route with NH 10.0.0.4 becomes unavailable, the other static route takes over. The following command disables the static route with NH 10.0.0.4 on PE-2.

```

# on PE-2:
configure {
  router "Base" {

```

```

static-routes {
  route 10.0.0.4/32 route-type unicast {
    next-hop "192.168.24.2" {
      admin-state disable
    }
  }
}

```

The FIB on PE-2 shows a labeled BGP route with resolved NH 192.168.24.2. Again, this FIB entry looks identical. The BGP NH is not resolved to a tunnel.

```

[/]
A:admin@PE-2# show router fib 1 ip-prefix-prefix-length 192.0.2.5/32
=====
FIB Display
=====
Prefix [Flags]                                Protocol
NextHop
-----
192.0.2.5/32                                  BGP_LABEL
  192.168.24.2 (int-PE-2-PE-4)
-----
Total Entries : 1
=====

```

On PE-2, the best and used labeled BGP route for prefix 192.0.2.5/32 has NH 192.0.2.4. The BGP route for prefix 192.0.2.5/32 with NH 10.0.0.4 is not valid, because the static route to 10.0.0.4/32 is disabled.

```

[/]
A:admin@PE-2# show router bgp routes 192.0.2.5/32 label-ipv4
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP LABEL-IPV4 Routes
=====
Flag Network                                LocalPref MED
NextHop (Router)                            Path-Id   IGP Cost
As-Path                                     Label
-----
u*>i 192.0.2.5/32                            None     None
      192.0.2.4                               None     1
      64500                                    524285
i     192.0.2.5/32                            None     None
      10.0.0.4                               None     0
      64500                                    524285
-----
Routes : 2
=====

```

On PE-2, NH 10.0.0.4 is not resolved, because the static route is disabled. NH 192.0.2.4 has resolved NH 192.168.24.2:

```
[/]
A:admin@PE-2# show router bgp next-hop
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====

BGP Next Hop
=====
Next Hop                               Pref      Owner
Resolving Prefix                       FibProg  Metric
Resolved Next Hop                       Colored  Ref. Count
Admin-tag-policy                         FlexAlgo Last Mod.
-----
10.0.0.4                               -         -
Unresolved                             -         -
--                                       -         -
--                                       00h09m04s
192.0.2.3                               15        ISIS
192.0.2.3/32                             N         10
192.168.23.2                             N         0
--                                       --        00h42m47s
192.0.2.4                               25      STATIC
192.0.2.4/32                             N         1
192.168.24.2                           N         0
--                                       --        00h30m07s
-----
Next Hops : 3
=====
```

The configuration on ASBR PE-2 is restored as follows and the BGP NH is resolved to the static route to 10.0.0.4 again:

```
# on PE-2:
configure {
  router "Base" {
    static-routes {
      route 10.0.0.4/32 route-type unicast {
        next-hop "192.168.24.2" {
          admin-state enable
        }
      }
    }
  }
}
```

Labeled IPv4 BGP NH resolved to tunneled route

When the system does not allow BGH NH resolution to static routes, the tunneled route is selected. The following command configures BGP NH resolution for labeled routes to its default setting:

```
# on PE-2:
configure {
  router "Base" {
    bgp {
      next-hop-resolution {
```

```

        labeled-routes {
            delete allow-static
        }
    }
}

```

On PE-2, the route table shows that the BGP labeled IPv4 route to 192.0.2.5/32 has NH 192.0.2.4, which is resolved to a tunnel:

```

[/]
A:admin@PE-2# show router route-table 192.0.2.5/32

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                                Type   Proto   Age           Pref
  Next Hop[Interface Name]                        Metric
-----
192.0.2.5/32                                     Remote BGP_LABEL 00h01m49s 170
  192.0.2.4 (tunneled)                            1
-----
No. of Routes: 1
---snip---
=====

```

On PE-2, the following FIB shows that the BGP labeled route uses an LDP tunnel to the NH 192.0.2.4:

```

[/]
A:admin@PE-2# show router fib 1 ip-prefix-prefix-length 192.0.2.5/32

=====
FIB Display
=====
Prefix [Flags]                                Protocol
  NextHop
-----
192.0.2.5/32                                  BGP_LABEL
  192.0.2.4 (Transport:LDP)
-----
Total Entries : 1
=====

```

PE-2 has two labeled BGP routes to prefix 192.0.2.5/32: the route with NH 10.0.0.4 is not valid because it requires a static route, which is not allowed for BGP NH resolution; the best and used route has NH 192.0.2.4 (which is the NH that is reached by an LDP tunnel):

```

[/]
A:admin@PE-2# show router bgp routes 192.0.2.5/32 label-ipv4

=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete
=====
BGP LABEL-IPV4 Routes

```

```

=====
Flag Network                               LocalPref MED
      Nexthop (Router)                    Path-Id   IGP Cost
      As-Path                               Label
-----
u*>i 192.0.2.5/32                          None     None
      192.0.2.4                            None     1
      64500                                 524285
i    192.0.2.5/32                          None     None
      10.0.0.4                             None     0
      64500                                 524285
-----
Routes : 2
=====

```

On PE-2, the following BGP NH list shows that NH 192.0.2.4 is resolved using a static route with NH 192.168.24.2:

```

[/]
A:admin@PE-2# show router bgp next-hop
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====

BGP Next Hop
=====
Next Hop                               Pref   Owner
Resolving Prefix                      FibProg Metric
Resolved Next Hop                     Colored Ref. Count
Admin-tag-policy                       FlexAlgo Last Mod.
-----
10.0.0.4                               5      STATIC
10.0.0.4/32                            N      1
192.168.24.2                           N      0
--                                     --    00h05m51s
192.0.2.3                               15     ISIS
192.0.2.3/32                           N      10
192.168.23.2                           N      0
--                                     --    00h49m53s
192.0.2.4                               25     STATIC
192.0.2.4/32                           N      1
192.168.24.2                           N      0
--                                     --    00h37m13s
-----
Next Hops : 3
=====

```

The configuration on the ASBRs is modified as follows and the BGP NH is resolved to the local route, to 192.168.24.2 again. Local routes prevail over tunneled routes.

```

# on PE-2, PE-4:
configure {
  router "Base" {
    bgp {
      group "eBGP4_local" {
        admin-state enable
      }
    }
  }
}

```


Labeled IPv4 BGP NH resolved to RTM route on RR

RR PE-3 is not in the data path and **next-hop-self** is disabled, which is the default setting. PE-3 does not have LDP tunnels to PE-1 and PE-2, so BGP NH resolution to RTM routes needs to be allowed, by configuring **rr-use-route-table true**. The following error is raised when attempting to configure **rr-use-route-table true** with **route-table-install true**:

```
[ex:/configure router "Base" bgp next-hop-resolution labeled-routes]
A:admin@PE-3# rr-use-route-table true
MINOR: BGP #12: configure router "Base" bgp route-table-install - Inconsistent Value error -
route-table-install and rr-use-route-table cannot both be set to true
```

The command **route-table-install** allows an RR to reflect routes without installing them in its FIB. This way, an RR can reflect more routes than it can install in its FIB.

The following configuration on RR PE-3 allows the use of the route table for labeled routes:

```
# on PE-3:
configure {
  router "Base" {
    bgp {
      route-table-install false
      split-horizon true
      next-hop-resolution {
        labeled-routes {
          rr-use-route-table true
        }
      }
    }
    group "iBGP" {
      peer-as 64496
      family {
        vpn-ipv4 true
        vpn-ipv6 true
        label-ipv4 true
      }
      advertise-inactive true
      cluster {
        cluster-id 192.0.2.3
      }
    }
    neighbor "192.0.2.1" {
      group "iBGP"
    }
    neighbor "192.0.2.2" {
      group "iBGP"
    }
  }
}
```

The following command on RR PE-3 shows that the labeled BGP route for 192.0.2.5/32 is not used. This is because the route is not installed in the FIB of the RR, which is allowed, because the RR is not in the data path and NHS is disabled.

```
[/]
A:admin@PE-3# show router bgp routes 192.0.2.5/32 label-ipv4
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
```

```
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
```

=====

BGP LABEL-IPV4 Routes

=====

Flag	Network Nexthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
*>i	192.0.2.5/32 192.0.2.2 64500	100 None	None 10 524284

Routes : 1

=====

The following labeled BGP route has NH 192.0.2.2, which is resolved to an IS-IS route:

```
[/]
A:admin@PE-3# show router bgp next-hop 192.0.2.2
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====

BGP Next Hop
=====
Next Hop                               Pref   Owner
Resolving Prefix                       FibProg Metric
Resolved Next Hop                       Colored Ref. Count
Admin-tag-policy                         FlexAlgo Last Mod.
-----
192.0.2.2                             15    ISIS
192.0.2.2/32                            N      10
192.168.23.1                            N      0
--                                         --    00h59m37s
-----
Next Hops : 1
=====
```

RR PE-3 advertises this labeled BGP route to PE-1, which installs the route in its FIB, so it is used:

```
[/]
A:admin@PE-1# show router bgp routes label-ipv4
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====

Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

BGP LABEL-IPV4 Routes
=====
Flag Network                               LocalPref MED
Nexthop (Router) Path-Id IGP Cost
As-Path Label
-----
u*>i 192.0.2.5/32                         100    None
```

```

192.0.2.2          None          10
64500             524284
-----
Routes : 1
=====

```

The tunnel table on PE-1 has a BGP tunnel to 192.0.2.5 with NH 192.0.2.2 and an LDP tunnel to 192.0.2.2 with NH 192.168.12.2:

```

[/]
A:admin@PE-1# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner    Encap TunnelId Pref  Nexthop      Metric
  Color
-----
192.0.2.2/32     ldp     MPLS  65537   9    192.168.12.2  10
192.0.2.5/32     bgp     MPLS  262146  12   192.0.2.2    1000
-----
---snip---
=====

```

On PE-1, the BGP NH for route 192.0.2.5/32 is resolved to an LDP tunnel to PE-2:

```

[/]
A:admin@PE-1# show router fp-tunnel-table 1

=====
IPv4 Tunnel Table Display

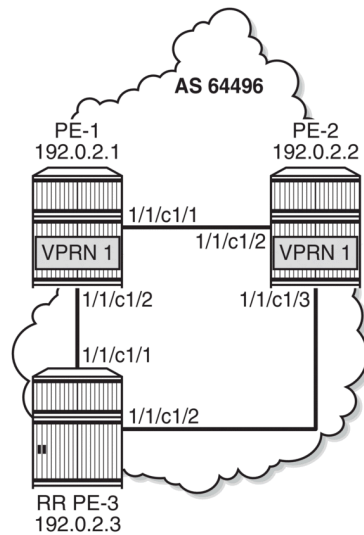
Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination      Protocol      Tunnel-ID
  Lbl/SID
  NextHop          Intf/Tunnel
  Lbl/SID (backup)
  NextHop  (backup)
-----
192.0.2.2/32     LDP           -
  524287
  192.168.12.2
192.0.2.5/32   BGP           -
524284
192.0.2.2     LDP
-----
Total Entries : 2
=====

```

NH resolution for iBGP VPN-IPv4/v6 routes

Figure 161: VPRN 1 in AS 64496 shows that VPRN 1 is configured on PE-1 and PE-2 in AS 64496.

Figure 161: VPRN 1 in AS 64496



38425

On both PE-1 and PE-2, the VPN-IPv4 and VPN-IPv6 address families are configured in group "iBGP":

```
# on PE-1, PE-2:
configure {
  router "Base" {
    bgp {
      split-horizon true
      group "iBGP" {
        peer-as 64496
        export {
          policy ["export-bgp"]
        }
      }
    }
    neighbor "192.0.2.3" {
      group "iBGP"
      family {
        vpn-ipv4 true
        vpn-ipv6 true
      }
    }
  }
}
```

On PE-1, VPRN 1 is configured as follows. The configuration on PE-2 is similar.

```
# on PE-1:
configure {
  service {
    vprn "VPRN 1" {
      admin-state enable
      service-id 1
      customer "1"
      bgp-ipvpn {
        mpls {
          admin-state enable
        }
      }
    }
  }
}
```



```
-----snip-----
```

PE-2 receives the following BGP VPN-IPv4 route with route distinguisher (RD) 64496:1 used in VPRN 1:

```
[/]
A:admin@PE-2# show router bgp routes vpn-ipv4 rd 64496:1
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  64496:1:1.1.1.1/32                     100        None
      192.0.2.1                             None        10
      No As-Path                             524282
-----
Routes : 1
=====
```

For iBGP VPN routes on a node that is not an RR, the NH can only be resolved using a tunnel in the TTM. If the BGP NH is an IPv4 address, the system uses the most preferred tunnel matching the address and allowed by the resolution filter. The resolution filter allows LDP and BGP, but within an AS, only LDP tunnels are used. The following FIB for VPRN 1 on PE-2 shows that the transport tunnel to NH 192.0.2.1 is an LDP tunnel:

```
[/]
A:admin@PE-2# show router 1 fib 1
=====
FIB Display
=====
Prefix [Flags]                               Protocol
NextHop
-----
1.1.1.1/32                                   BGP_VPN
  192.0.2.1 (VPRN Label:524282 Transport:LDP)
2.2.2.1/32                                   LOCAL
  2.2.2.1 (loopback1)
-----
Total Entries : 2
=====
```

The same is shown for BGP IPv6 routes:

```
[/]
A:admin@PE-2# show router bgp routes vpn-ipv6 rd 64496:1
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
```

```
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
```

=====

BGP VPN-IPv6 Routes

=====

Flag	Network Nextthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
u*>i	64496:1:2001:db8::1:1:1/128 ::ffff:192.0.2.1 No As-Path	100 None	None 10 524282

Routes : 1

=====

The following IPv6 FIB for VPRN 1 shows that a LDP tunnel is used to reach NH 192.0.2.1:

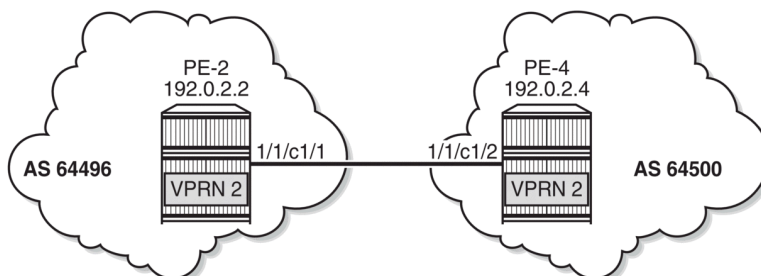
```
[/]
A:admin@PE-2# show router 1 fib 1 ipv6

=====
FIB Display
=====
Prefix [Flags]                                Protocol
NextHop
-----
2001:db8::1:1:1/128                            BGP_VPN
  192.0.2.1 (VPRN Label:524282 Transport:LDP)
2001:db8::2:2:2:1/128                          LOCAL
  2001:db8::2:2:2:1 (loopback1)
-----
Total Entries : 2
=====
```

NH resolution for inter-AS VPRN model B

Figure 162: VPRN 2 in AS 64496 and in AS 64500 shows that VPRN 2 is configured in AS 64496 and in AS 64500.

Figure 162: VPRN 2 in AS 64496 and in AS 64500



38426

On PE-2, VPRN 2 is configured as follows. The service configuration on PE-4 is similar.

```
# on PE-2:
configure {
  service {
    vprn "VPRN 2" {
      admin-state enable
      service-id 2
      customer "1"
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "2:2"
          vrf-target {
            community "target:2:2"
          }
          auto-bind-tunnel {
            resolution filter
            resolution-filter {
              ldp true
            }
          }
        }
      }
    }
  }
  interface "loopback2" {
    loopback true
    ipv4 {
      primary {
        address 2.2.2.2
        prefix-length 32
      }
    }
    ipv6 {
      address 2001:db8::2:2:2:2 {
        prefix-length 128
      }
    }
  }
}
}
```

BGP is configured for the VPN IP address families and BGP NH can be resolved to static routes. Multiple eBGP neighbors are defined, with NHs that can be resolved to a local, static, or tunneled route. The BGP configuration on PE-2 is as follows. The BGP configuration on PE-4 is similar.

```
# on PE-2:
configure {
  router "Base" {
    bgp {
      inter-as-vpn true
      split-horizon true
      rapid-update {
        vpn-ipv4 true
        vpn-ipv6 true
        label-ipv4 true
      }
    }
    next-hop-resolution {
      labeled-routes {
        allow-static true
      }
    }
  }
}
```



```

10.0.0.4          None      1
64500            524284
*i 2:2:4.4.4.2/32 None      None
192.0.2.4        None      1
64500            524284
-----
Routes : 3
=====

```

The IPv4 FIB on PE-2 shows prefix 4.4.4.2/32 with NH 192.168.24.2 on int-PE-2-PE-4. The NH is not resolved to a tunnel.

```

[/]
A:admin@PE-2# show router 2 fib 1
=====
FIB Display
=====
Prefix [Flags]          Protocol
NextHop
-----
2.2.2.2/32              LOCAL
 2.2.2.2 (loopback2)
4.4.4.2/32              BGP_VPN
 192.168.24.2 (int-PE-2-PE-4)
-----
Total Entries : 2
=====

```

In a similar way, the used VPN-IPv6 route on PE-2 has a NH resolved to a local route:

```

[/]
A:admin@PE-2# show router bgp routes 2001:db8::4:4:4:2/128 vpn-ipv6
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv6 Routes
=====
Flag Network          LocalPref  MED
Nexthop (Router)      Path-Id    IGP Cost
As-Path               Label
-----
u*>i 2:2:2001:db8::4:4:4:2/128  None      None
      ::ffff:192.168.24.2      None      0
      64500                    524284
*i 2:2:2001:db8::4:4:4:2/128  None      None
      ::ffff:10.0.0.4         None      1
      64500                    524284
*i 2:2:2001:db8::4:4:4:2/128  None      None
      ::ffff:192.0.2.4        None      1
      64500                    524284
-----
Routes : 3
=====

```

VPN IP NH resolved to static route

When the eBGP session using the interface addresses is disabled, the next preferred NH resolution is static, which is allowed by configuration:

```
# on PE-2:
configure {
  router "Base" {
    bgp {
      group "eBGP4_local" {
        admin-state disable
      }
    }
  }
}
```

On PE-2, the static route with the best preference is toward 10.0.0.4:

```
[/]
A:admin@PE-2# show router bgp routes 4.4.4.2/32 vpn-ipv4
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  2:2:4.4.4.2/32           None       None
      10.0.0.4              None       1
      64500                  None       524284
*>i   2:2:4.4.4.2/32           None       None
      192.0.2.4             None       1
      64500                  None       524284
-----
Routes : 2
=====
```

On PE-2, NH 10.0.0.4 is resolved to 192.168.24.2:

```
[/]
A:admin@PE-2# show router bgp next-hop
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
BGP Next Hop
=====
Next Hop                Pref  Owner
Resolving Prefix        FibProg Metric
Resolved Next Hop      Colored Ref. Count
Admin-tag-policy        FlexAlgo Last Mod.
```

```

-----
10.0.0.4          5      STATIC
  10.0.0.4/32    N      1
  192.168.24.2   N      0
  --            --    00h39m36s
192.0.2.4        25     STATIC
  192.0.2.4/32  N      1
  192.168.24.2  N      0
  --            --    01h10m58s
-----
Next Hops : 2
=====

```

This resolved NH 192.168.24.2 is the NH for prefix 4.4.4.2/32 in the FIB:

```

[/]
A:admin@PE-2# show router 2 fib 1

=====
FIB Display
=====
Prefix [Flags]          Protocol
NextHop
-----
2.2.2.2/32             LOCAL
  2.2.2.2 (loopback2)
4.4.4.2/32             BGP_VPN
  192.168.24.2 (int-PE-2-PE-4)
-----
Total Entries : 2
=====

```

For IPv6 routes on PE-2, the used route toward 2001:db8::4:4:4:2/128 has NH ::ffff:10.0.0.4:

```

[/]
A:admin@PE-2# show router bgp routes 2001:db8::4:4:4:2/128 vpn-ipv6

=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
BGP VPN-IPv6 Routes
=====
Flag Network          LocalPref  MED
NextHop (Router)      Path-Id    IGP Cost
As-Path               Label
-----
u*>i 2:2:2001:db8::4:4:4:2/128  None       None
      ::ffff:10.0.0.4         None       1
      64500                   524284
*>i  2:2:2001:db8::4:4:4:2/128  None       None
      ::ffff:192.0.2.4         None       1
      64500                   524284
-----
Routes : 2
=====

```

VPN IP NH resolved to tunneled route



Note:

This scenario is only for demonstration purposes. In an operational service provider network, no LDP sessions are established to an untrusted AS (inter-AS VPRN model B is used for untrusted connections).

When the BGP configuration is changed to the default setting that static routes are not allowed for the NH resolution, the used BGP route toward 4.4.4.2/32 uses a tunnel toward the system address of the eBGP peer. The BGP configuration is modified as follows:

```
# on PE-2:
configure {
  router "Base" {
    bgp {
      next-hop-resolution {
        labeled-routes {
          delete allow-static
        }
      }
    }
  }
}
```

On PE-2, the used VPN-IPv4 route toward 4.4.4.2/32 has NH 192.0.2.4:

```
[/]
A:admin@PE-2# show router bgp routes 4.4.4.2/32 vpn-ipv4
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  2:2:4.4.4.2/32                            None       None
      192.0.2.4                               None       1
      64500                                    None       524284
*i    2:2:4.4.4.2/32                            None       None
      10.0.0.4                                None       1
      64500                                    None       524284
-----
Routes : 2
=====
```

The tunnel table on PE-2 shows that an LDP tunnel is available toward 192.0.2.4/32:

```
[/]
A:admin@PE-2# show router tunnel-table 192.0.2.4/32
```

```

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.4/32     Ldp        MPLS  65538    9    192.168.24.2  1
---snip---
=====
    
```

The following FIB on PE-2 shows that an LDP tunnel is used toward NH 192.0.2.4 to reach prefix 4.4.4.2/32:

```

[/]
A:admin@PE-2# show router 2 fib 1

=====
FIB Display
=====
Prefix [Flags]                                Protocol
NextHop
-----
2.2.2.2/32                                     LOCAL
  2.2.2.2 (loopback2)
4.4.4.2/32                                   BGP_VPN
  192.0.2.4 (VPRN Label:524284 Transport:LDP)
-----
Total Entries : 2
=====
    
```

Similarly, the following IPv6 FIB on PE-2 shows that the same LDP tunnel is used toward NH 192.0.2.4 to reach prefix 2001:db8::4:4:4:2/128:

```

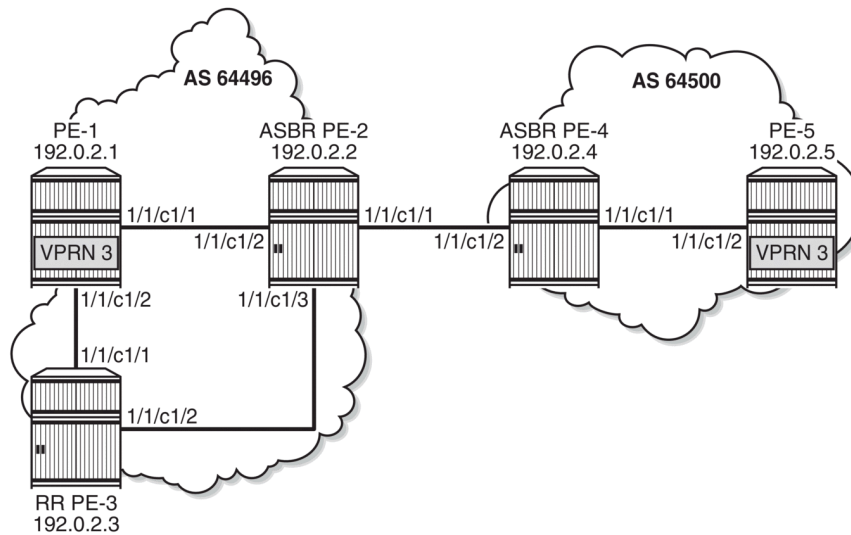
[/]
A:admin@PE-2# show router 2 fib 1 ipv6

=====
FIB Display
=====
Prefix [Flags]                                Protocol
NextHop
-----
2001:db8::2:2:2:2/128                          LOCAL
  2001:db8::2:2:2:2 (loopback2)
2001:db8::4:4:4:2/128                        BGP_VPN
  192.0.2.4 (VPRN Label:524284 Transport:LDP)
-----
Total Entries : 2
=====
    
```

NH resolution for inter-AS VPRN model C

Figure 163: VPRN 3 - inter-AS VPRN model C shows the example topology with RR PE-3 in AS 64496. VPRN 3 is configured on PE-1 and PE-5.

Figure 163: VPRN 3 - inter-AS VPRN model C



38427

A labeled IPv4 eBGP session is established between ASBRs PE-2 and PE-4, and a multi-hop eBGP session is established between PE-1 and PE-5 for the VPN-IPv4 and VPN-IPv6 address families. The following BGP configuration is configured on PE-1. The configuration on PE-5 is similar.

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      split-horizon true
      rapid-update {
        vpn-ipv4 true
        vpn-ipv6 true
        label-ipv4 true
      }
    }
    group "iBGP" {
      peer-as 64496
      export {
        policy ["export-bgp"]
      }
    }
    neighbor "192.0.2.3" {
      group "iBGP"
      family {
        vpn-ipv4 true
        vpn-ipv6 true
        label-ipv4 true
      }
    }
    group "eBGP_multihop" {
      peer-as 64500
    }
    neighbor "192.0.2.5" {
      group "eBGP_multihop"
      family {
        vpn-ipv4 true
        vpn-ipv6 true
      }
    }
  }
}
```

```

        local-address 192.0.2.1
        multihop 10
    }
}
}
}

```

The BGP configuration on RR PE-3 is as follows. The RR is configured with **route-table-install false**, so no routes are installed in the FIB; therefore, no eBGP multi-hop sessions can be established from the RR. The BGP NH is resolved using the RTM. Local routes would be preferred, but there are no candidates. BGP NH resolution to static routes is not allowed in this configuration.

```

# on PE-3:
configure {
  router "Base" {
    bgp {
      route-table-install false
      split-horizon true
      next-hop-resolution {
        labeled-routes {
          rr-use-route-table true
        }
      }
    }
    group "iBGP" {
      peer-as 64496
      advertise-inactive true
      cluster {
        cluster-id 192.0.2.3
      }
    }
    neighbor "192.0.2.1" {
      group "iBGP"
      family {
        vpn-ipv4 true
        vpn-ipv6 true
        label-ipv4 true
      }
    }
    neighbor "192.0.2.2" {
      group "iBGP"
      family {
        label-ipv4 true
      }
    }
  }
}
}

```

On the ASBRs, BGP is only configured for the labeled IPv4 address family. The BGP configuration on PE-2 is as follows. The configuration on PE-4 is similar.

```

# on PE-2:
configure {
  router "Base" {
    bgp {
      split-horizon true
      rapid-update {
        vpn-ipv4 true
        vpn-ipv6 true
        label-ipv4 true
      }
    }
    group "iBGP" {

```



```
}

```

With the preceding configuration, the resolution filter in VPRN 3 allows the use of LDP and BGP tunnels, which can be verified as follows. BGP tunnels are used for routes received from the peer AS.

```
[/]
A:admin@PE-1# configure {
  service {
    vprn "VPRN 3" {
      info detail | match "auto-bind-tunnel" post-lines 20

*[ex:/configure service vprn "VPRN 3"]
A:admin@PE-1# info detail | match "auto-bind-tunnel" post-lines 20
      auto-bind-tunnel {
        ---snip---
        resolution-filter {
          bgp true
          ---snip---
          ldp true
          ---snip---

```

On PE-1, the VPN-IPv4 route for prefix 5.5.5.3/32 has NH 192.0.2.5 in the peer AS. Prefix 5.5.5.3/32 is the IP address of a loopback interface in VPRN 3 on PE-5.

```
[/]
A:admin@PE-1# show router bgp routes vpn-ipv4 rd 3:3
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  3:3:5.5.5.3/32           None       None
      192.0.2.5             None       0
      64500                  524283
-----
Routes : 1
=====

```

On PE-1, the following tunnel table shows two tunnels: one LDP tunnel toward 192.0.2.2, and a BGP tunnel toward 192.0.2.5 in the remote AS.

```
[/]
A:admin@PE-1# show router tunnel-table
=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.2/32     ldp        MPLS  65537    9    192.168.12.2  10

```

```

192.0.2.5/32      bgp      MPLS  262145  12    192.0.2.2    1000
-----
---snip---
=====

```

The following FIB for VPRN 3 on PE-1 shows that the BGP tunnel is used for prefix 5.5.5.3/32 with NH 192.0.2.5:

```

[/]
A:admin@PE-1# show router 3 fib 1

=====
FIB Display
=====
Prefix [Flags]                                Protocol
NextHop
-----
1.1.1.3/32                                     LOCAL
  1.1.1.3 (loopback3)
5.5.5.3/32                                     BGP_VPN
  192.0.2.5 (VPRN Label:524283 Transport:BGP)
-----
Total Entries : 2
-----
=====

```

On RR PE-3, the following VPN IP routes with NH 192.0.2.1 are reflected, but they are not installed in the FIB, so these are not used locally:

```

[/]
A:admin@PE-3# show router bgp routes vpn-ipv4 rd 3:3

=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete

=====
BGP VPN-IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)    Path-Id    IGP Cost
      As-Path              Label
-----
*>i  3:3:1.1.1.3/32          100        None
      192.0.2.1           None        10
      No As-Path          524283
*>i  3:3:5.5.5.3/32          100        None
      192.0.2.5           None        0
      64500                524283
-----
Routes : 2
=====

```

On RR PE-3, NH 192.0.2.1 is resolved using the RTM:

```

[/]
A:admin@PE-3# show router bgp next-hop

=====

```

```

BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
=====
BGP Next Hop
=====
Next Hop                    Pref    Owner
  Resolving Prefix          FibProg Metric
  Resolved Next Hop        Colored Ref. Count
  Admin-tag-policy         FlexAlgo Last Mod.
-----
192.0.2.1                  15    ISIS
  192.0.2.1/32              N      10
  192.168.13.1            N      0
  --                        --     00h07m42s
192.0.2.2                  15    ISIS
  192.0.2.2/32              N      10
  192.168.23.1             N      0
  --                        --     00h07m42s
-----
Next Hops : 2
=====

```

Conclusion

The NH resolution of BGP routes using tunnels is consistent across different types of labeled route families (labeled IP and VPN-IP), both for eBGP and iBGP peering.

Policy Chaining and Logical Expressions

This chapter provides information about policy chaining and logical expressions.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 14.0.R4, but the MD-CLI in the current edition is based on SR OS Release 21.7.R1. In SR OS releases earlier than 14.0.R1, only policy chaining was supported. SR OS Release 14.0.R1 introduced support for route policy logical expressions using the logical operators AND, OR, and NOT, and parentheses.

Overview

Multiple policies can be chained together for sequential evaluation. For more complex evaluation logic, logical expressions can be used with operators: AND, OR, and NOT, and with parentheses. A logical expression can be included in a larger policy chain. Route policy logical expressions are supported in the following contexts:

- BGP export
- BGP import
- BGP leak-import (RIB leaking)
- VRF import
- VRF export
- GRT export (GRT leaking)

[Table 5: Policy chaining versus policy logical expressions](#) shows a comparison between examples of policy chaining and policy logical expressions.

Table 5: Policy chaining versus policy logical expressions

Policy chaining example	Policy logical expressions example
configure router bgp import policy ["A" "B" "C"]	configure router bgp import policy "[A]OR[B]"
For each route, policy A is evaluated first. <ul style="list-style-type: none"> • If policy A matches the route with action next- 	Several logical operators can be used. This example shows an OR relationship

Policy chaining example	Policy logical expressions example
<p>policy, then apply any route modifications and continue to evaluate policy B, and so on.</p> <ul style="list-style-type: none"> When the route is matched in a policy with action accept or reject, the evaluation is completed. 	<p>between policy A and policy B.</p> <p>For each route, policy A is evaluated first. A true/false result is determined for policy A:</p> <ul style="list-style-type: none"> If true, the logical expression with operator OR is true already, and the evaluation is completed. If false, then policy B is evaluated to determine the final true/false result. <p>The final result is mapped back to a policy action (accept, next-policy, and so on).</p>

To configure policy chaining that may or may not include a policy logical expression, the syntax is:

```
[ex:configure router "Base" bgp export]
A:admin@PE-1# policy

policy <value>
policy [<value>...] - 1..15 user-ordered values separated by spaces enclosed by
brackets

<value> - <string>
<string> - <1..255 characters>

Export policy name
```

When the **import/export policy** command has a single value, the value is either a policy or a logical expression. The **policy** may or may not be enclosed in double quotes. When enclosed in double quotes, all characters (including blanks) are considered part of the policy. The **logical expression** must be enclosed in double quotes. The logical expression accepts between 1 and 16 policies. In the logical expression:

- each policy must be enclosed in square brackets and must not be enclosed in double quotes.
- the operand may or may not be separated with spaces from the policies.
- the operand must be in uppercase (AND, OR, NOT).
- parentheses may be used to influence the logic. When used, nesting is limited to a maximum of 3 levels.

A policy is accepted with and without double quotes enclosing it.

```
# on PE-1:
configure {
  router "Base" {
```

```

    bgp {
      delete import      # to remove already existing import policies
      import {
        policy A
      }
    }
  (leads to: ["A"])

```

```

configure {
  router "Base" {
    bgp {
      delete import      # to remove already existing import policies
      import {
        policy "A"
      }
    }
  }
}
(leads to: ["A"])

```

In a policy that is enclosed in double quotes, spaces before and after the policy are not allowed, as they are considered part of the policy name. The following message is raised when spaces are present before and after a policy that is enclosed in double quotes.

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import      # to remove already existing import policies
      import {
        policy " A "
      }
    }
  }
}
MINOR: MGMT_CORE #224: configure router "Base" bgp import policy - Entry does not exist -
configure policy-options policy-statement " A "

```

A logical expression is accepted only when each policy is enclosed in square brackets and not in double quotes, while the logical expression itself is enclosed in double quotes.

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import      # to remove already existing import policies
      import {
        policy "[A]AND[B]"
      }
    }
  }
}
(leads to: ["[A]AND[B]"])

```

In a logical expression, spaces before and after the operand are allowed.

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import      # to remove already existing import policies
      import {
        policy "[A] AND [B]"
      }
    }
  }
}
(leads to: ["[A] AND [B]"])

```

In a logical expression, spaces before and after the square brackets that enclose a policy are allowed.

```

# on PE-1:

```

```
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy " [A] AND [B] "
      }
    }
  }
}
(leads to: [" [A] AND [B] "])
```

In a logical expression, spaces inside the square brackets that enclose a policy are allowed.

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy "[ A ]AND[ B ]"
      }
    }
  }
}
(leads to: ["[ A ]AND[ B ]"])
```

The following message is raised when the logical expression is not enclosed in double quotes.

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy [A]AND[B]
      }
    }
  }
}
A:admin@PE-1# import policy [A]AND[B]
MINOR: MGMT_CORE #2201: Unknown element - 'AND'
```

The following message is raised when there are double quotes inside the square brackets that enclose a policy in a logical expression that is not enclosed in double quotes.

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy ["A"]AND["B"]
      }
    }
  }
}
A:admin@PE-1# import policy ["A"]AND["B"]
MINOR: MGMT_CORE #2201: Unknown element - 'AND'
```

The following message is raised when a policy is enclosed in double quotes instead of square brackets in a logical expression that is not enclosed in double quotes.

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy "A"AND"B"
      }
    }
  }
}
```



```

}
A:admin@PE-1#         import policy "A"AND"B"
                        ^^^
MINOR: MGMT_CORE #2201: Unknown element - 'AND'

```

The following message is raised when there are double quotes inside the square brackets that enclose a policy in a logical expression.

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import    # to remove already existing import policies
      import {
        policy "["A"]AND["B"]"
      }
A:admin@PE-1#         import policy "["A"]AND["B"]"
                        ^
MINOR: MGMT_CORE #2201: Unknown element - 'A'

```

The following message is raised when a policy is enclosed in double quotes instead of square brackets in a logical expression.

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import    # to remove already existing import policies
      import {
        policy ""A"AND"B""
      }
A:admin@PE-1#         import policy ""A"AND"B""
                        ^
MINOR: MGMT_CORE #2301: Invalid element value - 'policy' expected string '<1..255 characters>'
or reference '<1..64 characters>' (configure policy-options policy-statement <name>)

```

The operand in a logical expression must be in uppercase.

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import    # to remove already existing import policies
      import {
        policy "[A]AND[B]"
      }
}
(leads to: ["[A]AND[B]"])

```

The following message is raised when an operand in a logical expression is not in uppercase.

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import    # to remove already existing import policies
      import {
        policy "[A]and[B]"
      }
A:admin@PE-1#         import policy "[A]and[B]"
                        ^^^^^^^^^^

```

```
MINOR: MGMT_CORE #2301: Invalid element value - 'policy' expected string '<1..255 characters>'
or reference '<1..64 characters>' (configure policy-options policy-statement <name>)
```

A logical expression may use parentheses (nesting up to a maximum of 3 levels) to influence the logic.

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy "(((P1]AND[P2])OR[P3])AND[P4])OR[P5]"
      }
    }
  }
}
(leads to: ["(((P1]AND[P2])OR[P3])AND[P4])OR[P5]"])
```

The following message is raised when the parentheses nesting exceeds 3 levels.

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy "((((P1]AND[P2])OR[P3])AND[P4])OR[P5])AND[P6]"
      }
    }
  }
}
A:admin@PE-1# import policy "((((P1]AND[P2])OR[P3])AND[P4])OR[P5])AND[P6]"
MINOR: MGMT_CORE #2301: Invalid element value - 'policy' expected string '<1..255 characters>'
or reference '<1..64 characters>' (configure policy-options policy-statement <name>)
```

The following message is raised when the parentheses are not balanced.

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy "(((P1]AND[P2])OR[P3])AND[P4]"
      }
    }
  }
}
A:admin@PE-1# import policy "(((P1]AND[P2])OR[P3])AND[P4]"
MINOR: MGMT_CORE #2301: Invalid element value - 'policy' expected string '<1..255 characters>'
or reference '<1..64 characters>' (configure policy-options policy-statement <name>)
```

A logical expression accepts a maximum of 16 policies.

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy
        "[P1]AND[P2]AND[P3]AND[P4]AND[P5]AND[P6]AND[P7]AND[P8]AND[P9]AND[P10]AND[P11]AND[P12]AND[P13]AND
[P14]AND[P15]AND[P16]"
      }
    }
  }
}
(leads to:
[" [P1]AND[P2]AND[P3]AND[P4]AND[P5]AND[P6]AND[P7]AND[P8]AND[P9]AND[P10]AND[P11]AND[P12]AND[P13]
AND[P14]AND[P15]AND[P16]" ])
```



```

        delete import    # to remove already existing import policies
        import {
            policy [A B]
        }
(leads to: ["A" "B"])

```

The following message is raised when the policy chain is not enclosed in square brackets.

```

# on PE-1:
configure {
    router "Base" {
        bgp {
            delete import    # to remove already existing import policies
            import {
                policy A B
            }
A:admin@PE-1#             import policy A B
                               ^
MINOR: MGMT_CORE #2201: Unknown element - 'B'

```

A policy chain is accepted with and without double quotes enclosing the policies.

```

# on PE-1:
configure {
    router "Base" {
        bgp {
            delete import    # to remove already existing import policies
            import {
                policy [A]
            }
(leads to: ["A"])

```

```

configure {
    router "Base" {
        bgp {
            delete import    # to remove already existing import policies
            import {
                policy ["A"]
            }
(leads to: ["A"])

```

```

configure {
    router "Base" {
        bgp {
            delete import    # to remove already existing import policies
            import {
                policy [A B]
            }
(leads to: ["A" "B"])

```

```

configure {
    router "Base" {
        bgp {
            delete import    # to remove already existing import policies
            import {
                policy ["A" B]
            }

```

```
(leads to: ["A" "B"])
```

```
configure {
  router "Base" {
    bgp {
      delete import    # to remove already existing import policies
      import {
        policy [A "B"]
      }
    }
  }
}
```

```
(leads to: ["A" "B"])
```

```
configure {
  router "Base" {
    bgp {
      delete import    # to remove already existing import policies
      import {
        policy ["A" "B"]
      }
    }
  }
}
```

```
(leads to: ["A" "B"])
```

A policy chain can contain policies and a logical expression.

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import    # to remove already existing import policies
      import {
        policy [A B "[C]OR[A]"]
      }
    }
  }
}
```

```
(leads to: ["A" "B" "[C]OR[A"])
```

In a policy chain, policies and the logical expression may be repeated. Only the first occurrence of duplicate policies and duplicate logical expressions is retained in the resulting policy chain.

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import    # to remove already existing import policies
      import {
        policy [A A "A"]
      }
    }
  }
}
```

```
(leads to: ["A"])
```

```
configure {
  router "Base" {
    bgp {
      delete import    # to remove already existing import policies
      import {
        policy [C A A C B A C B C A B C A]
      }
    }
  }
}
```

```
(leads to: ["C" "A" "B"])
```

```
configure {
  router "Base" {
    bgp {
      delete import    # to remove already existing import policies
      import {
```

```

        policy [C A "A" "C" B A C "B" "C" "A" B C A]
    }
    (leads to: ["C" "A" "B"])

```

```

configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy ["[A]AND[B]" "[A]AND[B]"]
      }
    }
  }
}
(leads to: ["[A]AND[B]"])

```

```

configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy ["[A]AND[B]" C "[A]AND[B]"]
      }
    }
  }
}
(leads to: ["[A]AND[B]" "C"])

```

A policy chain accepts a maximum of 15 policies.

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy [P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15]
      }
    }
  }
}
(leads to: ["P1" "P2" "P3" "P4" "P5" "P6" "P7" "P8" "P9" "P10" "P11" "P12" "P13" "P14" "P15"])

```

The following message is raised when there are too many policies in the policy chain.

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy [P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15 P16]
      }
    }
  }
}
MINOR: MGMT_CORE #253: configure router "Base" bgp import policy - Reached maximum number of
entries - number of entries must be between 1-15 but has 16

```

A policy chain has 1 logical expression at maximum. The following message is raised when the policy chain has more than 1 (different) logical expression.

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy ["[A]AND[B]" "[B]OR[C]"]
      }
    }
  }
}

```

MINOR: BGP #12: configure router "Base" bgp import policy - Inconsistent Value error - Policy chain exceeds maximum number of expressions - configure

```
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy ["[A]AND[B]" C "[B]OR[C]"]
      }
    }
  }
}
```

MINOR: BGP #12: configure router "Base" bgp import policy - Inconsistent Value error - Policy chain exceeds maximum number of expressions - configure

A logical expression with a length that does not exceed 64 characters can be anywhere in the policy chain (but the result can be different).

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy [A B "[C]AND[A]"]
      }
    }
  }
}
(leads to: ["A" "B" "[C]AND[A]"])
```

```
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy [A "[C]AND[A]" B]
      }
    }
  }
}
(leads to: ["A" "[C]AND[A]" "B"])
```

```
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy ["[C]AND[A]" A B]
      }
    }
  }
}
(leads to: ["[C]AND[A]" "A" "B"])
```

When the length of the logical expression exceeds 64 characters, the logical expression must be at the start of the policy chain.

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy ["[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]"
A B]
      }
    }
  }
}
(leads to: ["[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]" "A" "B"])
```

The following message is raised when a logical expression with a length that exceeds 64 characters is not at the start of a policy chain.

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy [A B
"[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]"
      }
}
MINOR: BGP #8: configure router "Base" bgp import policy - Wrong Length error - Policy entry 3
exceeds maximum length (64) - configure
```

The following message is raised when a logical expression with a length that exceeds 255 characters is at the start of a policy chain.

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import # to remove already existing import policies
      import {
        policy ["[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AN
D[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AN
D[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AN
D[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]"
A B]
      }
}
A:admin@PE-1# import policy ["[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]A
ND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]A
ND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]A
ND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]"
A B]
~~~~~
MINOR: MGMT_CORE #2301: Invalid element value - 'policy' expected string '<1..255 characters>'
or reference '<1..64 characters>' (configure policy-options policy-statement <name>)
```

Route policy logical expressions

Logical expressions are evaluated to be true or false. [Table 6: Boolean values for the policy actions](#) shows the mapping of policy actions to Boolean values.

Table 6: Boolean values for the policy actions

Policy action	Boolean value
Accept	True
Next-entry	True
Next-policy	True

Policy action	Boolean value
Reject	False

[Table 7: Actions for the logical operators](#) shows the evaluation actions for the logical operators NOT, OR, and AND.

Table 7: Actions for the logical operators

Logical operator	Action
NOT <expr>	Swaps the true/false result of the expression.
<expr1> OR <expr2>	If expr1 is true, the result is true and expr2 is not evaluated. If expr1 is false, expr2 must be evaluated. The final result is true if either expression is true; otherwise, it is false.
<expr1> AND <expr2>	If expr1 is false, the result is false and expr2 is not evaluated. If expr1 is true, expr2 must be evaluated. The final result is true only if both expressions are true.

[Table 8: Mapping the final result of an expression to a policy action](#) shows the mapping of the final result of an expression to a policy action. Routes are rejected when the entire expression is false.

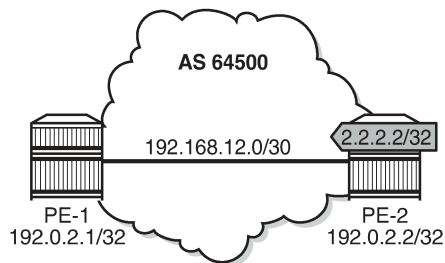
Table 8: Mapping the final result of an expression to a policy action

Final result	Action
True	accept , next-entry , or next-policy (depending on the last entry evaluated)
False	reject

Configuration

[Figure 164: Example topology](#) shows the example topology including the advertised route.

Figure 164: Example topology



26074

The initial configuration of the routers includes:

- Cards, MDAs, ports
- Router interfaces
- IS-IS
- LDP
- BGP
- Export policy "export-bgp" accepting routes for prefix 2.2.2.2/32 on PE-2.

It is possible to configure VPRNs and assign policies to BGP in the VPRN, although in this chapter, all the examples are for BGP in the base router.

Policy chaining and policy logical expressions

In this section, three route policies are configured that will add a community and set the **local-preference** (LP): only policy C does not set LP. Policy C has **action next-policy**, and policies A and B have **action accept**. The configuration is:

```
# on PE-1, PE-2:
configure {
  policy-options {
    community "A" {
      member "1:1" { }
    }
    community "B" {
      member "2:2" { }
    }
    community "C" {
      member "3:3" { }
    }
  }
  policy-statement "A" {
    entry 10 {
      action {
        action-type accept
        local-preference 110
        community {
          add ["A"]
        }
      }
    }
  }
  policy-statement "B" {
```

```

    entry 10 {
        action {
            action-type accept
            local-preference 220
            community {
                add ["B"]
            }
        }
    }
}
policy-statement "C" {
    entry 10 {
        action {
            action-type next-policy
            community {
                add ["C"]
            }
        }
    }
}
}

```

Initially, policy chaining is configured without a logical expression. Subsequently, policy chaining is configured with only one policy logical expression and no other policies in the chain, as described in the following sections.

Policy chaining without logical expression

Policy chaining may include one logical expression, except in this example, there is no policy logical expression in the chain.

Policy chaining is configured on PE-1:

```

# on PE-1:
configure {
    router "Base" {
        bgp {
            import {
                policy ["C" "A" "B"]
            }
        }
    }
}

```

PE-1 receives route 2.2.2.2/32 from PE-2. For each route, PE-1 evaluates policy C first. This policy adds community C (3:3) and has **action next-policy**, which implies that the next policy must also be evaluated. Policy A adds community A (1:1) and sets the LP to a value of 110 (by default, the **local-preference** equals 100). Policy A has **action accept** and, therefore, the evaluation is completed. The local-preference and the community are shown in the following output:

```

[/]
A:admin@PE-1# show router bgp routes hunt brief
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====

```

```

-----
RIB In Entries
-----
Network       : 2.2.2.2/32
Nextthop     : 192.0.2.2
Path Id      : None
From         : 192.0.2.2
Res. Protocol : ISIS                Res. Metric   : 10
Res. Nextthop : 192.168.12.2
Local Pref.  : 110                  Interface Name : int-PE-1-PE-2
Aggregator AS : None                Aggregator    : None
Atomic Aggr. : Not Atomic           MED           : None
AIGP Metric  : None                IGP Cost      : 10
Connector    : None
Community    : 1:1 3:3
---snip---

```

Policy logical expressions with two policies

In the following examples, the policy chain contains only a policy logical expression. When both policy A and policy B must be executed, the logical operator used is: AND. The sequence is important in this case, because both policies A and B set the LP and the last executed policy will set the final value for the LP. The following import policy expression is configured on PE-1:

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import
      import {
        policy ["[A]AND[B]"]
      }
    }
  }
}

```

Policy A is evaluated first and it adds community A (1:1) and sets LP 110. Then, policy B is evaluated, which adds community B (2:2) and sets LP 220.

```

[/]
A:admin@PE-1# show router bgp routes hunt brief | match "Local Pref."
Local Pref.      : 220                Interface Name : int-PE-1-PE-2

```

```

[/]
A:admin@PE-1# show router bgp routes hunt brief | match "Community"
Community        : 2:2 1:1

```

When the policy expression is [B]AND[A], the order is reversed. First, policy B sets LP 220, then policy A sets LP 110:

```

[/]
A:admin@PE-1# show router bgp routes hunt brief | match "Local Pref."
Local Pref.      : 110                Interface Name : int-PE-1-PE-2

```

```

[/]
A:admin@PE-1# show router bgp routes hunt brief | match "Community"
Community        : 1:1 2:2

```

When the policy expression contains operator OR instead of AND, the first true expression results in a completed evaluation. Because both policy A and policy B result in a true expression, whichever policy is evaluated first is executed and the second one is skipped. For example, when policy A is evaluated first and the result is true, policy B is skipped. Therefore, the community is A (1:1) and the LP 110:

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import
      import {
        policy ["[A]OR[B]"]
      }
    }
  }
}
```

```
[/]
A:admin@PE-1# show router bgp routes hunt brief | match "Local Pref."
Local Pref.      : 110                               Interface Name : int-PE-1-PE-2
```

```
[/]
A:admin@PE-1# show router bgp routes hunt brief | match "Community"
Community        : 1:1
```

Likewise, when policy B is evaluated first and the result is true, policy A is skipped. The added community is B (2:2) and the LP 220:

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import
      import {
        policy ["[B]OR[A]"]
      }
    }
  }
}
```

```
[/]
A:admin@PE-1# show router bgp routes hunt brief | match "Local Pref."
Local Pref.      : 220                               Interface Name : int-PE-1-PE-2
```

```
[/]
A:admin@PE-1# show router bgp routes hunt brief | match "Community"
Community        : 2:2
```

The logical operator NOT swaps the result from true to false, and vice versa. When policy A is evaluated as true, NOT[A] is false. A false expression in an AND relationship leads to a false result. The next policy in the logical expression is not evaluated. No communities are added and no LP is set (the default value for LP is 100). The route is rejected as invalid:

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import
      import {
        policy ["NOT[A]AND[B]"]
      }
    }
  }
}
```

```

}

[/]
A:admin@PE-1# show router bgp routes hunt brief
---snip---
-----
RIB In Entries
-----
Network      : 2.2.2.2/32
Nextthop    : 192.0.2.2
---snip---
Local Pref.  : 100                      Interface Name : int-PE-1-PE-2
---snip---
Community    : No Community Members
---snip---
Flags        : Invalid IGP Rejected
---snip---

```

However, a false NOT[A] expression in an OR relationship may still lead to the expression being evaluated to true:

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import
      import {
        policy ["NOT[A]OR[B]"]
      }
    }
  }
}

```

```

[/]
A:admin@PE-1# show router bgp routes hunt brief
---snip---
-----
RIB In Entries
-----
Network      : 2.2.2.2/32
Nextthop    : 192.0.2.2
---snip---
Local Pref.  : 220                      Interface Name : int-PE-1-PE-2
---snip---
Community    : 2:2 1:1
---snip---
Flags        : Used Valid Best IGP In-RTM
---snip---

```

Policy B is evaluated as true for the route and, therefore, the entire logical expression "NOT[A]OR[B]" is true, and the route is accepted. Every policy in the expression that was evaluated, before the entire logical expression was recognized to be true, is executed, including policy A. This implies that policy A adds community A (1:1) to the route and sets LP to a value of 110. Then, policy B adds community B (2:2) to the route and overwrites the LP to a value of 220.

The import policy "[B] OR NOT[A]" is true after the first policy is evaluated as true. Only policy B is executed, the assigned community is B (2:2) and the LP is 220:

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import

```

```

import {
    policy ["[B] OR NOT[A]"]
}

[/]
A:admin@PE-1# show router bgp routes hunt brief
---snip---
-----
RIB In Entries
-----
Network      : 2.2.2.2/32
Nexthop      : 192.0.2.2
---snip---
Local Pref.  : 220                      Interface Name : int-PE-1-PE-2
---snip---
Community    : 2:2
---snip---
Flags        : Used Valid Best IGP In-RTM
---snip---

```

Table 9: Assigned LP and communities for the import logical expressions summarizes the results for these different scenarios.

Table 9: Assigned LP and communities for the import logical expressions

Import logical expression	Assigned LP	Assigned community
import "[A]AND[B]"	220	2:2 1:1
import "[B]AND[A]"	110	1:1 2:2
import "[A]OR[B]"	110	1:1
import "[B]OR[A]"	220	2:2
import "NOT[A]AND[B]"	None	None (Route rejected)
import "NOT[A]OR[B]"	220	2:2 1:1
Import "[B] OR NOT[A]"	220	2:2

Policy logical expressions with three policies

In policy chaining, the next policy in the chain is evaluated when the action is **next-policy**. In policy logical expressions, the next policy is evaluated depending on the logical operator and the Boolean value for the previous policies in the expression.

Policy C has **action next-policy** instead of **accept** and adds community C (3:3), but does not set the LP.

Several logical expressions can be made with policies A, B, and C. The following import policy has all three policies in an AND relationship. The expression is evaluated as true and all policies are executed: three communities are added and the LP is set.

```

# on PE-1:
configure {
    router "Base" {

```

```

    bgp {
      delete import
      import {
        policy ["[C]AND[A]AND[B]"]
      }
    }

```

The first policy adds community C (3:3), the second policy adds community A (1:1) and sets LP 110, and the third policy adds community B (2:2) and sets LP 220:

```

[/]
A:admin@PE-1# show router bgp routes hunt brief
---snip---
-----
RIB In Entries
-----
Network       : 2.2.2.2/32
Nextthop      : 192.0.2.2
---snip---
Local Pref.   : 220                               Interface Name : int-PE-1-PE-2
---snip---
Community     : 2:2 1:1 3:3
---snip---
Flags         : Used Valid Best IGP In-RTM
---snip---

```

The import policy "[C]AND[A]OR[B]" results in the first two being executed. Policy C is evaluated as true, and the logical operation is AND. Therefore, the next policy must be evaluated too. Policy A is also evaluated as true and the next operation is OR. The final result is evaluated as true without evaluating policy B. The communities added are C and A (3:3 and later 1:1) and the LP is 110.

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import
      import {
        policy ["[C]AND[A]OR[B]"]
      }
    }
  }
}

```

```

[/]
A:admin@PE-1# show router bgp routes hunt brief
---snip---
-----
RIB In Entries
-----
Network       : 2.2.2.2/32
Nextthop      : 192.0.2.2
---snip---
Local Pref.   : 110                               Interface Name : int-PE-1-PE-2
---snip---
Community     : 1:1 3:3
---snip---
Flags         : Used Valid Best IGP In-RTM
---snip---

```

The import policy "[C]OR[A]OR[B]" is evaluated as true after the first policy is evaluated as true. Even though the action in policy C is **next-policy**, the next policy in this expression is not evaluated, because

the expression is already true. Only policy C is executed and it adds the community C (3:3), but does not configure the LP:

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import
      import {
        policy ["[C]OR[A]OR[B]"]
      }
    }
  }
}
```

```
[/]
A:admin@PE-1# show router bgp routes hunt brief
---snip---
-----
RIB In Entries
-----
Network      : 2.2.2.2/32
Nexthop      : 192.0.2.2
---snip---
Local Pref.  : 100                      Interface Name : int-PE-1-PE-2
---snip---
Community    : 3:3
---snip---
Flags        : Used Valid Best IGP
---snip---
```

However, if the policy chain contains not only a logical expression, but also single policies, the action next-policy ensures that a following policy in the chain is executed; for example, policy D in the following policy chain:

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import
      import {
        policy ["[C]OR[A]OR[B]" "D"]
      }
    }
  }
}
```

The expression "[C]OR[A]OR[B]" is true after policy C has been evaluated, but policy C has **action next-policy** and policy D is the next policy to be evaluated.

The import policy "[C]OR[A]AND[B]" expression evaluates policy C as true. Policy C has an OR relationship with policy A in the logical expression "[C]OR[A]", and therefore, policy A is not evaluated. There is an AND relationship with policy B and policy B is evaluated as true. Therefore, the entire logical expression "[C]OR[A]AND[B]" is true and the route is accepted. Both policy C and B are executed. First, policy C adds community C (3:3), then policy B adds community B (2:2) and sets LP 220:

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import
      import {
        policy ["[C]OR[A]AND[B]"]
      }
    }
  }
}
```

```

}

[/]
A:admin@PE-1# show router bgp routes hunt brief
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
-----
RIB In Entries
-----
Network       : 2.2.2.2/32
NextHop       : 192.0.2.2
---snip---
Local Pref.   : 220                Interface Name : int-PE-1-PE-2
---snip---
Community     : 2:2 3:3
---snip---
Flags         : Used Valid Best IGP In-RTM
---snip---

```

Table 10: Assigned LP and communities for the import logical expressions summarizes the results for these different scenarios.

Table 10: Assigned LP and communities for the import logical expressions

Import logical expression	Assigned LP	Assigned community
import "[C]AND[A]AND[B]"	220	2:2 1:1 3:3
import "[C]AND[A]OR[B]"	110	1:1 3:3
import "[C]OR[A]OR[B]"	None	3:3
import "[C]OR[A]AND[B]"	220	2:2 3:3

Combinations of policy logical operations using brackets

For this section, the following communities and policies are configured on PE-1. All these policies have a **from** condition that matches a community (D, E, F, G). Besides these policies, there are also export policies on PE-2 that add one or more communities (D, E, F, G) to the advertised routes. On PE-1, incoming route 2.2.2.2/32 will have one or more communities that may or may not match the **from** condition in the following route policies.

```

# on PE-1 and PE-2:
configure {
  policy-options {
    community "D" {
      member "4:4" { }
    }
  }
}

```

```
community "E" {
  member "5:5" { }
}
community "F" {
  member "6:6" { }
}
community "G" {
  member "7:7" { }
}
policy-statement "D" {
  entry 10 {
    from {
      community {
        name "D"
      }
    }
    action {
      action-type accept
      local-preference 4
    }
  }
  default-action {
    action-type reject
  }
}
policy-statement "E" {
  entry 10 {
    from {
      community {
        name "E"
      }
    }
    action {
      action-type accept
      local-preference 5
    }
  }
  default-action {
    action-type reject
  }
}
policy-statement "F" {
  entry 10 {
    from {
      community {
        name "F"
      }
    }
    action {
      action-type accept
      local-preference 6
    }
  }
  default-action {
    action-type reject
  }
}
policy-statement "G" {
  entry 10 {
    from {
      community {
        name "G"
      }
    }
  }
}
```

```

        action {
            action-type accept
            local-preference 7
        }
    }
    default-action {
        action-type reject
    }
}

```

The received routes have community E (5:5) present. The following import policy is configured on PE-1:

```

# on PE-1:
configure {
    router "Base" {
        bgp {
            delete import
            import {
                policy ["([D]AND[E])OR([F]AND[G])"]
            }
        }
    }
}

```

The first policy that is evaluated requires community D (4:4) to be present. This is not the case and the expression between brackets, ([D]AND[E]), is false. Policy E is not evaluated. The next policy to be evaluated is F and it requires community F (6:6), which is not present. The second expression between brackets, ([F]AND[G]), is therefore also false and policy G is not evaluated. The entire policy logical expression is false and the route is rejected.

The following commands show what policy evaluation caused the route to be rejected. For the entire logical expression "([D]AND[E])OR([F]AND[G])", the last policy that was evaluated, and that caused the route to be rejected, was policy F:

```

[/]
A:admin@PE-1# show router bgp policy-test plcy-or-long-expr "([D]AND[E])OR([F]AND[G])" family
  ipv4 prefix 0.0.0.0/0 longer display-rejects brief
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
      Network
-----
Rejected by Logical expression last policy F Default action
      2.2.2.2/32
-----
Total Routes : 1 Routes rejected : 1
=====

```

For the logical expression "[D]AND[E]", the last policy that was evaluated, and that led to the conclusion that the expression was false, was policy D:

```

[/]
A:admin@PE-1# show router bgp policy-test plcy-or-long-expr "[D]AND[E]" family ipv4 prefix
  0.0.0.0/0 longer display-rejects brief
=====

```

```

BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Network
-----
Rejected by Logical expression last policy D Default action
2.2.2.2/32
-----
Total Routes : 1 Routes rejected : 1
=====

```

For the logical expression "[F]AND[G]", the last policy that was evaluated, and that led to the conclusion that the expression was false, was policy F:

```

[/]
A:admin@PE-1# show router bgp policy-test plcy-or-long-expr "[F]AND[G]" family ipv4 prefix
0.0.0.0/0 longer display-rejects brief
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Network
-----
Rejected by Logical expression last policy F Default action
2.2.2.2/32
-----
Total Routes : 1 Routes rejected : 1
=====

```

The logical expression "([D]AND[E])OR([F]AND[G])" is false and, therefore, the route is rejected. No LP is set. Community E (5:5) was already present in the incoming route.

```

[/]
A:admin@PE-1# show router bgp routes hunt brief
---snip---
-----
RIB In Entries
-----
Network       : 2.2.2.2/32
Nextthop     : 192.0.2.2
---snip---
Local Pref.   : 100                Interface Name : int-PE-1-PE-2
---snip---
Community     : 5:5
---snip---
Flags         : Invalid IGP Rejected
---snip---

```

In the second example, the incoming route contains communities D (4:4) and E (5:5). The same policy logical expression " $([D]AND[E])OR([F]AND[G])$ " is evaluated as true because both policy D and policy E are true. There is an OR relationship with the rest of the expression and, therefore, the entire logical expression is true. Policy E is the last policy to be evaluated:

```
[/]
A:admin@PE-1# show router bgp policy-test plcy-or-long-expr "([D]AND[E])OR([F]AND[G])" family
  ipv4 prefix 0.0.0.0/0 longer display-rejects brief
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
      Network
-----
Accepted by Logical expression last policy E Entry 10
      2.2.2.2/32
-----
Routes : 1
=====
```

The route is accepted as valid and gets LP 5. The communities D (4:4) and E (5:5) were already present for the incoming route. The first policy that was executed, was policy D and it set the LP to a value of 4. Policy E was the second and last policy that was executed and it set the LP to a value of 5:

```
[/]
A:admin@PE-1# show router bgp routes hunt brief
---snip---
-----
RIB In Entries
-----
Network      : 2.2.2.2/32
Nexthop      : 192.0.2.2
---snip---
Local Pref.  : 5                      Interface Name : int-PE-1-PE-2
---snip---
Community    : 5:5 4:4
Cluster      : No Cluster Members
Originator Id : None                    Peer Router Id : 192.0.2.2
Fwd Class    : None                    Priority       : None
Flags        : Used Valid Best IGP In-RTM
---snip---
```

For the third example, the incoming route contains communities D (4:4) and E (5:5). The logical expression " $([D]OR[E])AND([F]OR[G])$ " is evaluated as false and the route is rejected:

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import
      import {
        policy ["([D]OR[E])AND([F]OR[G])"]
      }
    }
  }
}
```

First, policy D is evaluated as true because community D (4:4) is present. Policy D has an OR relationship with policy E, which is true without the need to evaluate policy E. The next policy to be evaluated is F. Policy F requires the community F (6:6) to be present, which is not the case. The logical expression "[F]OR[G]" can only be true if policy G is true. Policy G requires community G (7:7) to be present, which is false. The last policy that was evaluated before the route was rejected was policy G:

```
[/]
A:admin@PE-1# show router bgp policy-test plcy-or-long-expr "([D]OR[E])AND([F]OR[G])" family
  ipv4 prefix 0.0.0.0/0 longer display-rejects brief
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
      Network
-----
Rejected by Logical expression last policy G Default action
      2.2.2.2/32
-----
Total Routes : 1 Routes rejected : 1
=====
```

The route was rejected and, therefore, no policy was executed. The LP kept its default value of 100:

```
[/]
A:admin@PE-1# show router bgp routes hunt brief
---snip---
-----
RIB In Entries
-----
Network       : 2.2.2.2/32
Nexthop       : 192.0.2.2
---snip---
Local Pref.   : 100                Interface Name : int-PE-1-PE-2
---snip---
Community     : 5:5 4:4
---snip---
Flags         : Invalid IGP Rejected
---snip---
```

For the fourth example, the incoming route has communities E (5:5) and G (7:7). The logical expression "([D]OR[E])AND([F]OR[G])" is evaluated as true and the route is accepted. First, policy D is evaluated as false. Policy D has an OR relationship with policy E, which is evaluated as true. Consequently, the expression "[D]OR[E]" is true. This expression has an AND relationship with the expression "[F]OR[G]".

The next policy to be evaluated is F. Policy F requires the community F (6:6) to be present, which is false. The logical expression "[F]OR[G]" can only be true if policy G is true. Policy G requires community G (7:7) to be present, which is true. This makes [F]OR[G] true as well as the entire expression "([D]OR[E])AND([F]OR[G])".

The last policy that was evaluated before the route was accepted was policy G:

```
[/]
```

```
A:admin@PE-1# show router bgp policy-test plcy-or-long-expr "([D]OR[E])AND([F]OR[G])" family
  ipv4 prefix 0.0.0.0/0 longer display-rejects brief
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
      Network
-----
Accepted by Logical expression last policy G Entry 10
      2.2.2.2/32
-----
Routes : 1
=====
```

The route was accepted and has the changes of all policies that were evaluated: initially, policy D set the LP to 4. This value was overwritten by policy E to 5, by policy F to 6, and finally by policy G to a value of 7:

```
[/]
A:admin@PE-1# show router bgp routes hunt brief
---snip---
-----
RIB In Entries
-----
Network       : 2.2.2.2/32
Nexthop       : 192.0.2.2
---snip---
Local Pref.   : 7                      Interface Name : int-PE-1-PE-2
---snip---
Community     : 5:5 7:7
---snip---
Flags         : Used Valid Best IGP In-RTM
---snip---
```

[Table 11: Assigned LP for the import logical expressions](#) summarizes the results for these different scenarios.

Table 11: Assigned LP for the import logical expressions

Ingress community	Import logical expression	Assigned LP
5:5	import "([D]AND[E])OR([F]AND[G])"	Prefix rejected
5:5 4:4	import "([D]AND[E])OR([F]AND[G])"	5
5:5 4:4	import "([D]OR[E])AND([F]OR[G])"	Prefix rejected
5:5 7:7	import "([D]OR[E])AND([F]OR[G])"	7

Modification of attributes while processing

During the policy evaluation process, some prefix attributes can be modified while processing, and these modified attributes can be used as criteria for other policies in the logical expression.

In the following example, two route policies are configured:

- Policy X adds a new community Y (11:11) to the incoming route update.
- Policy Y uses community Y (11:11) as the only match criterion and removes communities X and Y. Policy Y also sets the LP to a value of 9, which is used here as an indication that policy Y was executed.

An export policy on PE-2 adds community X (10:10) to prefix 2.2.2.2/32 (not shown here).

Route policies X and Y are configured on PE-1:

```
# on PE-1:
configure {
  policy-options {
    community "X" {
      member "10:10" { }
    }
    community "Y" {
      member "11:11" { }
    }
  }
  policy-statement "X" {
    entry 10 {
      from {
        community {
          name "X"
        }
      }
      action {
        action-type accept
        community {
          add ["Y"]
        }
      }
    }
  }
  policy-statement "Y" {
    entry 10 {
      from {
        community {
          name "Y"
        }
      }
      action {
        action-type accept
        local-preference 9
        community {
          remove ["X" "Y"]
        }
      }
    }
  }
}
```

When no import policy is applied on PE-1, the received route 2.2.2.2/32 has community 10:10 and the default LP:

```
[/]
A:admin@PE-1# show router bgp routes hunt brief
```

```

---snip---
-----
RIB In Entries
-----
Network      : 2.2.2.2/32
Nextthop    : 192.0.2.2
---snip---
Local Pref.  : 100                      Interface Name : int-PE-1-PE-2
---snip---
Community    : 10:10
---snip---
Flags        : Used Valid Best IGP In-RTM
---snip---

```

The import policy "[X]AND[Y]" is configured on PE-1:

```

# on PE-1:
configure {
  router "Base" {
    bgp {
      delete import
      import {
        policy ["[X]AND[Y]"]
      }
    }
  }
}

```

The route update contains community X (10:10) and policy X is evaluated as true. Policy X adds community Y (11:11) to the route. Policy Y requires this community and is evaluated as true. Therefore, the entire logical expression "[X]AND[Y]" is true and the route is accepted. Policy Y removes communities X (10:10) and Y (11:11), and sets the LP to a value of 9:

```

[/]
A:admin@PE-1# show router bgp routes hunt brief
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
-----
RIB In Entries
-----
Network      : 2.2.2.2/32
Nextthop    : 192.0.2.2
---snip---
Local Pref.  : 9                      Interface Name : int-PE-1-PE-2
---snip---
Community    : No Community Members
---snip---
Flags        : Used Valid Best IGP In-RTM
---snip---

```

Conclusion

Route policy chaining and logical expressions allow complex route processing logic to be broken into smaller components. These policy components are reusable and facilitate the process of updating route control logic. Logical expressions offer more flexible combinations of policy statements.

Pop-Label for /32 Label-IPv4 BGP Routes

This chapter describes the pop-label for /32 label-IPv4 BGP routes.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

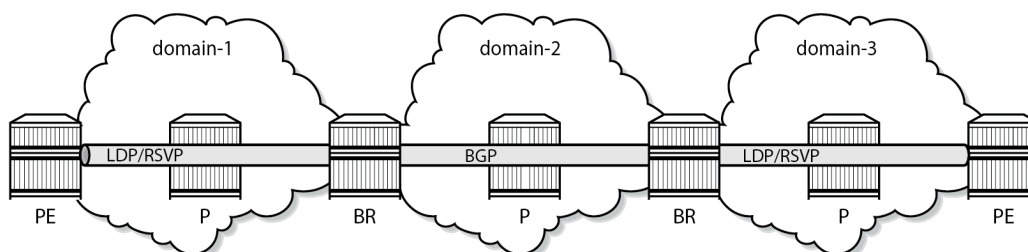
This chapter was initially written based on SR OS Release 15.0.R5, but the MD-CLI in the current edition is based on SR OS Release 23.7.R2.

Pop label for /32 label-IPv4 BGP routes is supported in SR OS Release 15.0.R1 and later.

Overview

Labeled IPv4 routes are used in seamless MPLS and in VPRN inter-AS model C scenarios. In these scenarios, transport tunnels run through multiple domains, where the area border routers (ABRs) or autonomous system border routers (ASBRs) effectively stitch LDP/RSVP tunnels to BGP tunnels. For inter-AS model C, the domain is an autonomous system (AS); for seamless MPLS, the domain is a part of an autonomous system. In either case, an end-to-end transport tunnel can be considered as a concatenation of multiple transport tunnels; as illustrated in [Figure 165: Stitching RSVP/LDP tunnels to BGP tunnels](#).

Figure 165: Stitching RSVP/LDP tunnels to BGP tunnels



27611b

Pop-label for /32 label-IPv4 routes allows operators to save on resources used in the network (less swap ingress label mapping entries in the data path) and can be implemented at the border routers (ABR or ASBR) for /32 label-IPv4 BGP routes that are originated by exporting static, OSPF, or IS-IS routes from the route table into BGP.

Pop label for /32 label-IPv4 BGP routes provides a tighter coupling between the LDP/RSVP-TE and the BGP tunnels stitched at the ABR or ASBR, as follows:

1. By implementing an **accept** policy action (without the **advertise-label pop** modifier) for the /32 addresses in a **route-table-import** policy. The router advertises a /32 label-IPv4 route with a label

that is swapped when an LDP/RSVP-TE is available, and withdrawn when the last LDP/RSVP-TE tunnel to that /32 prefix goes down. This applies to PEs with services, but should not be applied for route reflectors (RRs) when VPN addresses will be exchanged across EBGP sessions, because withdrawing labels for RRs would break the exchange of VPN routes. For the use of the **route-table-import** command, see the [Separate BGP RIBs for Labeled Routes](#) chapter.

2. By implementing the **accept** policy action with the **advertise-label pop** modifier for some system addresses in a **route-table-import** policy. The router advertises a /32 label-IPv4 route with a label that is popped rather than swapped, in case no LDP/RSVP-TE tunnel is available to that /32 prefix. This particularly applies to infrastructure nodes, for example off-data-path RRs, which do not participate in MPLS. RRs in different ASs, for example, still must be able to peer with each other through a multi-hop EBGP session, for the exchange of VPN routes belonging to the different services.

The **advertise-label pop** modifier can be used for the label-IPv4 redistribution of /32 prefixes of:

- OSPF and IS-IS routes
- Static routes:
 - Direct next-hop
 - Indirect next-hop
 - Blackhole

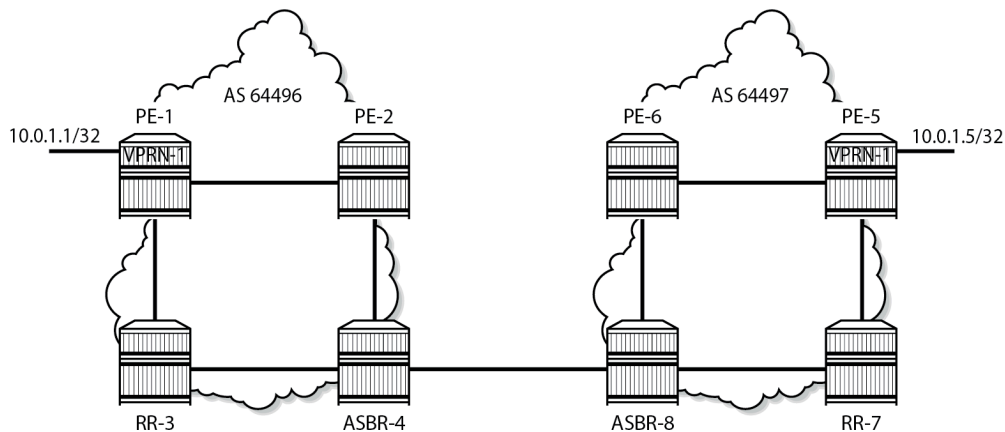
Redistributing /32 blackhole static routes does not require the **advertise-label pop** modifier; the label-IPv4 route is always advertised to the peer AS, and popped by the data plane.

The configuration in this chapter describes the redistribution of /32 prefixes for IS-IS routes. The redistribution of /32 routes for OSPF and the different static route types is similar.

Configuration

[Figure 166: Example topology](#) shows the example topology, depicting the inter-AS scenario also used in the "Inter-AS VPRN Model C" chapter. PE-1 and PE-5 host VPRN service "VPRN-1", with 10.0.1.1/32 and 10.0.1.5/32 being the loopback addresses for this service on PE-1 and PE-5, respectively. In AS 64496, RR-3 is the IPv4 VPN RR, and ASBR-4 is the label-IPv4 RR toward clients PE-1 and PE-2. In AS 64497, RR-7 is the IPv4 VPN RR, and ASBR-8 is the label-IPv4 RR toward clients PE-5 and PE-6. IS-IS is the IGP for AS 64496 and 64497, and ASBR-4 and ASBR-8 are their respective ASBRs. Additionally, and in support for model C, the RR-3 and RR-7 RRs require a multi-hop IPv4 VPN EBGP connection.

Figure 166: Example topology



27612b

The initial configuration includes:

1. Cards, MDAs, and ports.
2. Router interfaces.
3. IS-IS as IGP on all interfaces within AS 64496 and AS 64497 (alternatively, OSPF can be used).
4. LDP configured between PE-1, PE-2, and ASBR-4 in AS 64496, and between PE-5, PE-6, and ASBR-8 in AS 64497. The RR-3 and RR-7 RRs are off-data-path and do not have LDP enabled.

Base configuration

In this example topology, the PEs and the ASBRs generate labeled routes. The export policy configured on PE-1, PE-2, PE-5, and PE-6 advertises the system address 192.0.2.x/32. The export policy configured on the ASBRs advertises the system address of the RR. ASBR-4 and ASBR-8 advertise the system addresses of the PEs and the RRs to each other. The transport tunnels available in ASs 64496 and 64497 are LDP tunnels.

PE-1 and PE-2 peer with RR RR-3 for IPv4 VPN routes, and with RR ASBR-4 for label-IPv4 routes. This enables PE-1 and PE-2 to exchange service traffic with the PEs in the peer AS. Their internal BGP configuration is as follows:

```
# on PE-1, PE-2:
configure {
  policy-options {
    prefix-list "sys" {
      prefix 192.0.2.0/29 type range {
        start-length 32
        end-length 32
      }
    }
  }
  policy-statement "exp-sys" {
    entry 10 {
      from {
        prefix-list ["sys"]
        protocol {
          name [direct]
        }
      }
    }
  }
}
```

```

        }
        action {
            action-type accept
        }
    }
}
router "Base" {
    autonomous-system 64496
    bgp {
        loop-detect discard-route
        split-horizon true
        group "IBGP" {
            peer-as 64496
        }
        neighbor "192.0.2.3" {
            group "IBGP"
            family {
                vpn-ipv4 true
            }
        }
        neighbor "192.0.2.4" {
            group "IBGP"
            family {
                label-ipv4 true
            }
        }
        export {
            policy ["exp-sys"]
        }
    }
}

```

RR-3 is the IPv4 VPN RR for internal clients, using cluster ID 192.0.2.3, so it maintains IBGP sessions with PE-1 and PE-2. RR-3 also maintains a multi-hop EBGP session with RR-7, which is the RR for clients PE-5 and PE-6 in AS 64497. The **vpn-apply-import**, **vpn-apply-export**, and **import** and **export** commands can be used at **bgp**, **group**, or **neighbor** level for selectively exchanging dedicated VPN routes. The BGP configuration for RR-3 is as follows:

```

# on RR-3:
configure {
    policy-options {
        prefix-list "PE-pfxs" {
            prefix 192.0.2.1/32 type exact {
            }
            prefix 192.0.2.2/32 type exact {
            }
        }
        route-distinguisher-list "rd-123" {
            rd-entry "64497:1" { }
            rd-entry "64497:2" { }
            rd-entry "64497:3" { }
        }
        policy-statement "ebgp-exp-vpn" {
            entry 10 {
                from {
                    next-hop {
                        prefix-list "PE-pfxs"
                    }
                }
                action {
                    action-type accept
                }
            }
        }
    }
}

```

```

    }
  }
  policy-statement "ebgp-imp-vpn" {
    entry 10 {
      from {
        route-distinguisher-list "rd-123"
      }
      action {
        action-type accept
      }
    }
  }
}
router "Base" {
  autonomous-system 64496
  bgp {
    loop-detect discard-route
    route-table-install false
    split-horizon true
    group "EBGP-VPN" {
      peer-as 64497
      local-address 192.0.2.3
      export {
        policy ["ebgp-exp-vpn"]
      }
      import {
        policy ["ebgp-imp-vpn"]
      }
    }
    group "IBGP-VPN" {
      peer-as 64496
      cluster {
        cluster-id 192.0.2.3
      }
    }
    neighbor "192.0.2.1" {
      group "IBGP-VPN"
      family {
        vpn-ipv4 true
      }
    }
    neighbor "192.0.2.2" {
      group "IBGP-VPN"
      family {
        vpn-ipv4 true
      }
    }
    neighbor "192.0.2.7" {
      group "EBGP-VPN"
      multihop 10
      vpn-apply-export true
      vpn-apply-import true
      family {
        vpn-ipv4 true
      }
    }
  }
}

```

On ASBR-4, the *RR-pfxs* prefix list is the exact /32 address of RR-3. The *imp-pfxs* policy in ASBR-4 matches the *RR-pfxs* prefix list in entry 10 with action accept and the **advertise-label pop** modifier; entry 20 matches the **PE-pfxs** prefix list with action accept without modifier. The *exp-sys* policy is used to advertise the RR prefix and the received label-IPv4 routes for the PE prefixes to the peer AS. The system

prefixes of PE-1 and PE-2 are advertised by the PEs as label-IPv4 routes. The policy configuration on ASBR-4 is as follows:

```
# on ASBR-4:
configure {
  policy-options {
    prefix-list "PE-pfxs" {
      prefix 192.0.2.1/32 type exact {
      }
      prefix 192.0.2.2/32 type exact {
      }
    }
    prefix-list "RR-pfxs" {
      prefix 192.0.2.3/32 type exact {
      }
    }
  }
  policy-statement "exp-sys" {
    entry 10 {
      from {
        prefix-list ["RR-pfxs"]
      }
      action {
        action-type accept
      }
    }
    entry 20 {
      from {
        protocol {
          name [bgp-label]
        }
      }
      action {
        action-type accept
      }
    }
  }
  policy-statement "imp-sys" {
    entry 10 {
      from {
        prefix-list ["RR-pfxs"]
      }
      action {
        action-type accept
        advertise-label pop
      }
    }
    entry 20 {
      from {
        prefix-list ["PE-pfxs"]
      }
      action {
        action-type accept
      }
    }
  }
}
```

ASBR-4 is the label-IPv4 RR for internal clients, using cluster ID 192.0.2.4, so it maintains IBGP sessions with PE-1 and PE-2. ASBR-4 imposes **next-hop-self** on the IBGP advertised label-IPv4 routes. ASBR-4 also maintains an EBGP session with ASBR-8, and requires the **advertise-inactive true** command. The reason for the **advertise-inactive** command is that the system IP addresses for PEs are advertised in IGP and in BGP. Because the IGP has a lower preference value than BGP, the BGP routes are rendered

inactive. By default, inactive BGP routes are not advertised to the peer AS, and the **advertise-inactive true** command bypasses this issue. The BGP configuration for ASBR-4 is as follows:

```
# on ASBR-4:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      loop-detect discard-route
      split-horizon true
      rib-management {
        label-ipv4 {
          route-table-import {
            policy-name "imp-sys"
          }
        }
      }
    }
    group "EBGP-label" {
      advertise-inactive true
      ebgp-default-reject-policy {
        import false
      }
      export {
        policy ["exp-sys"]
      }
    }
    group "IBGP-label" {
      next-hop-self true
      peer-as 64496
      cluster {
        cluster-id 192.0.2.4
      }
    }
    neighbor "192.0.2.1" {
      group "IBGP-label"
      family {
        label-ipv4 true
      }
    }
    neighbor "192.0.2.2" {
      group "IBGP-label"
      family {
        label-ipv4 true
      }
    }
    neighbor "192.168.48.2" {
      group "EBGP-label"
      peer-as 64497
      family {
        label-ipv4 true
      }
    }
  }
}
```

Because RR-3 is deliberately placed off the data path, not participating in MPLS, an indirect static route is added to its configuration so that it can establish an EBGP session with RR-7, as follows:

```
# on RR-3:
configure {
  router "Base" {
    static-routes {
      route 192.0.2.7/32 route-type unicast {
```

```

        indirect 192.0.2.4 {
            admin-state enable
        }
    }
}

```

The configuration of the nodes in AS 64497 is similar to the nodes in AS 64496; see [Figure 166: Example topology](#) for the addresses required.

Redistributing IGP /32 routes to label-IPv4 routes

With the configuration as indicated in the previous section, ASBR-4 advertises the system addresses of PE-1, PE-2, and RR-3 in AS 64496 to ASBR-8 in the peer AS as label-IPv4 routes, as follows:

```

[/]
A:admin@ASBR-4# show router bgp neighbor 192.168.48.2 advertised-routes label-ipv4
=====
BGP Router ID:192.0.2.4      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP LABEL-IPV4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
i     192.0.2.1/32           n/a        20
      192.168.48.1         None       n/a
      64496                 None       524281
i     192.0.2.2/32           n/a        10
      192.168.48.1         None       n/a
      64496                 None       524282
i     192.0.2.3/32           n/a        10
      192.168.48.1         None       n/a
      64496                 None       524284
-----
Routes : 3
=====

```

The label-IPv4 routes are accepted and put in the routing table of ASBR-8. The next hop for all the label-IPv4 routes is 192.168.48.1, as follows:

```

[/]
A:admin@ASBR-8# show router route-table 192.0.2.0/24 longer
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age      Pref
Next Hop[Interface Name]   Metric
-----
192.0.2.1/32                Remote BGP_LABEL 00h01m15s 170
      192.168.48.1          0
192.0.2.2/32                Remote BGP_LABEL 00h01m15s 170
      192.168.48.1          0

```

```

192.0.2.3/32 Remote BGP_LABEL 00h01m15s 170
    192.168.48.1 0
192.0.2.5/32 Remote ISIS 00h35m37s 18
    192.168.68.1 20
192.0.2.6/32 Remote ISIS 00h35m37s 18
    192.168.68.1 10
192.0.2.7/32 Remote ISIS 00h35m37s 18
    192.168.78.1 10
192.0.2.8/32 Local Local 00h35m38s 0
    system 0
-----
No. of Routes: 7
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

Also, ASBR-8 is advertising label-IPv4 routes to ASBR-4, so that ASBR-4 ultimately has LDP and BGP tunnels available to destinations in its own and its peer AS, respectively, as follows:

```

[/]
A:admin@ASBR-4# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.1/32     ldp        MPLS  65538    9    192.168.24.1  20
192.0.2.2/32     ldp        MPLS  65537    9    192.168.24.1  10
192.0.2.5/32     bgp        MPLS  262147   12   192.168.48.2  1000
192.0.2.6/32     bgp        MPLS  262146   12   192.168.48.2  1000
192.0.2.7/32     bgp        MPLS  262145   12   192.168.48.2  1000
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====

```

The following shows the BGP inter-AS label mapping on ASBR-4:

```

[/]
A:admin@ASBR-4# show router bgp inter-as-label

=====
BGP Inter-AS labels
Flags: B - entry has backup, P - entry is promoted
=====
NextHop          Received      Advertised     Label
                  Label         Label          Origin
-----
0.0.0.0          0             524280         Edge
192.0.2.1        524284        524284         Internal
192.0.2.2        524284        524283         Internal
192.168.48.2     524279        524279         External
192.168.48.2     524281        524282         External
192.168.48.2     524284        524281         External
-----
Total Labels allocated: 6

```

The first entry in this table, with advertised label 524280, is used for tunnels for which ASBR-4 is the end-point, so that no label mapping is required. This is indicated by setting the next hop to 0.0.0.0, the received label to 0, and the label origin to Edge.

The second and third entries, with advertised labels 524284 and 524283, are used for tunnels to PE-1 and PE-2, respectively. Taking PE-2 as an example, label 524283 is swapped to label 524284.

The last three entries, with advertised labels 524279, 524282, and 524281, and received labels 524279, 524281, and 524284, respectively, are used for tunnels to the PEs and RR in the peer AS, which can be verified by displaying the label-IPv4 routes received by ASBR-4, as follows:

```
[/]
A:admin@ASBR-4# show router bgp neighbor 192.168.48.2 received-routes label-ipv4
=====
BGP Router ID:192.0.2.4      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP LABEL-IPV4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Path-Id    Label
-----
u*>i  192.0.2.5/32                             n/a        None
      192.168.48.2                             None        0
      64497                                       524284
u*>i  192.0.2.6/32                             n/a        None
      192.168.48.2                             None        0
      64497                                       524281
u*>i  192.0.2.7/32                             n/a        10
      192.168.48.2                             None        0
      64497                                       524279
-----
Routes : 3
=====
```

Verifying the content of the RIB provides an alternative to check whether tunnels are stitched. A check is performed for PE-1, which has service "VPRN-1" defined, and for RR-3, which does not have any services.

On ASBR-4, the label-IPv4 route for the 192.0.2.1/32 prefix in the RIB-In contains the received label 524284 with next hop 192.0.2.1 resolved to an LDP tunnel; in the RIB-Out, the advertised BGP label to next hop 192.168.48.2 is 524284, and the label type is swap, as follows. This is consistent with the output from the previous commands. The label-IPv4 BGP route in RIB-In is valid, but not used on ASBR-4, because an IS-IS route is preferred between PE-1 and ASBR-4 in AS 64496 (TieBreakReason : RtmPref).

```
[/]
A:admin@ASBR-4# show router bgp routes 192.0.2.1/32 label-ipv4 hunt
=====
BGP Router ID:192.0.2.4      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
```

```
=====
BGP LABEL-IPV4 Routes
=====
-----
RIB In Entries
-----
Network       : 192.0.2.1/32
Nexthop       : 192.0.2.1
Path Id       : None
From          : 192.0.2.1
Res. Nexthop  : 192.0.2.1 (LDP)
Local Pref.   : 100
Aggregator AS : None
Atomic Aggr.  : Not Atomic
AIGP Metric   : None
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None
Fwd Class     : None
IPv4 Label    : 524284
Flags         : Valid IGP
TieBreakReason : RtmPref
Route Source  : Internal
AS-Path       : No As-Path
Route Tag     : 0
Neighbor-AS   : n/a
DB Orig Val   : NotFound
Source Class  : 0
Add Paths Send : Default
RIB Priority  : Normal
Last Modified : 00h10m05s
Interface Name : NotAvailable
Aggregator    : None
MED           : None
IGP Cost      : 20
Peer Router Id : 192.0.2.1
Priority       : None
Final Orig Val : NotFound
Dest Class    : 0
-----
RIB Out Entries
-----
Network       : 192.0.2.1/32
Nexthop       : 192.168.48.1
Path Id       : None
To            : 192.168.48.2
Res. Nexthop  : n/a
Local Pref.   : n/a
Aggregator AS : None
Atomic Aggr.  : Not Atomic
AIGP Metric   : None
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None
IPv4 Label    : 524284
Lbl Allocation : NEXT-HOP
Origin        : IGP
AS-Path       : 64496
Route Tag     : 0
Neighbor-AS   : 64496
DB Orig Val   : NotFound
Source Class  : 0
Interface Name : NotAvailable
Aggregator    : None
MED           : None
IGP Cost      : 20
Peer Router Id : 192.0.2.8
Label Type    : SWAP
Final Orig Val : N/A
Dest Class    : 0
-----
Routes : 2
=====
```

Checking for the 192.0.2.3/32 prefix in the ASBR-4 RIB shows that label 524280 is advertised to 192.168.48.2, and the label type is pop, as follows:

```
[/]
A:admin@ASBR-4# show router bgp routes 192.0.2.3/32 label-ipv4 hunt
=====
BGP Router ID:192.0.2.4      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP LABEL-IPV4 Routes
=====
-----
RIB In Entries
-----
-----
RIB Out Entries
-----
-----
Network       : 192.0.2.3/32
Nexthop       : 192.168.48.1
Path Id       : None
To            : 192.168.48.2
Res. Nexthop  : n/a
Local Pref.   : n/a
Aggregator AS : None
Atomic Aggr.  : Not Atomic
AIGP Metric   : None
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None
IPv4 Label    : 524280
Lbl Allocation : NEXT-HOP
Origin        : IGP
AS-Path       : 64496
Route Tag     : 0
Neighbor-AS   : 64496
DB Orig Val   : N/A
Source Class  : 0
Interface Name : NotAvailable
Aggregator    : None
MED           : 10
IGP Cost      : n/a
Peer Router Id : 192.0.2.8
Label Type    : POP
Final Orig Val : N/A
Dest Class    : 0
-----
Routes : 1
=====
```

RR-3 and RR-7 have a multi-hop EBGP session established and are exchanging VPN routes, as follows:

```
[/]
A:admin@RR-3# show router bgp summary all
=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId      AS PktRcvd InQ Up/Down State|Rcv/Act/Sent (Addr Family)
                PktSent OutQ
```

```
-----
192.0.2.1
Def. Inst      64496      29      0 00h11m17s 2/0/4 (VpnIPv4)
                30      0
192.0.2.2
Def. Inst      64496      28      0 00h11m17s 1/0/5 (VpnIPv4)
                31      0
192.0.2.7
Def. Inst      64497      13      0 00h03m22s 3/0/3 (VpnIPv4)
                13      0
-----
```

Communication between VPRN-1 on PE-1 and on PE-5 is verified with a ping:

```
[/]
A:admin@PE-1# ping 10.0.1.5 router-instance "VPRN-1"
PING 10.0.1.5 56 data bytes
64 bytes from 10.0.1.5: icmp_seq=1 ttl=64 time=7.15ms.
64 bytes from 10.0.1.5: icmp_seq=2 ttl=64 time=6.57ms.
64 bytes from 10.0.1.5: icmp_seq=3 ttl=64 time=6.76ms.
64 bytes from 10.0.1.5: icmp_seq=4 ttl=64 time=6.62ms.
64 bytes from 10.0.1.5: icmp_seq=5 ttl=64 time=6.55ms.

---- 10.0.1.5 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 6.55ms, avg = 6.73ms, max = 7.15ms, stddev = 0.222ms
```

Disabling LDP on PE-1 results in ASBR-4 withdrawing the label-IPv4 route for prefix 192.0.2.1/32, as follows:

```
17 2023/09/27 08:23:26.963 UTC MINOR: DEBUG #2001 Base Peer 1: 192.168.48.2
"Peer 1: 192.168.48.2: UPDATE
Peer 1: 192.168.48.2 - Send BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 15
  Flag: 0x90 Type: 15 Len: 11 Multiprotocol Unreachable NLRI:
    Address Family LBL-IPV4
    192.0.2.1/32 Label 0
"
```

Conclusion

Implementing the **advertise-label pop** policy action in a **route-table-import** policy provides operators the means to save on resources used in the network.

Route Policy Action to Suppress BGP Route Installation

This chapter describes Route Policy Action to Suppress BGP Route Installation.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information and MD-CLI configuration in this chapter are based on SR OS Release 20.5.R1. The route policy action to suppress BGP and BGP Labeled Unicast (BGP-LU) route installation in the route table and tunnel table associated with the BGP instance is supported in SR OS Release 19.10.R1 and later.

Overview

In some deployments, a Route Reflector (RR) or PE router receives many BGP routes that must be re-advertised to other peers whereas these BGP routes do not need to be installed in the route table and Forwarding Information Base (FIB) of the RR or PE router. Network operators can suppress BGP route installation in the route table when they know that the router can forward the associated traffic anyway; for example, using a default or summary route. By suppressing BGP route installation, CPM memory is saved as well as FIB table space in the line cards.

The **route-table-install false** policy action only takes effect in BGP import policies and only for the IPv4, IPv6, label-IPv4, and label-IPv6 address families.

With this policy action in place, the following applies:

- when a BGP unlabeled IPv4 or IPv6 route is received from a base router or VPRN BGP peer, the route is:
 - not installed in the Route Table Manager (RTM)
 - not downloaded to the IOMs for installation in the FIB tables
 - not available for CPM routing (for example, for control plane traffic)
 - not available to resolve other routes
- when a BGP-LU IPv4 route is received from a base router or VPRN BGP peer, the route is:
 - not installed in the RTM and Tunnel Table Manager (TTM)
 - not downloaded to the IOMs for installation in the FIB tables
 - not available for CPM routing (for example, for control plane traffic)
 - not available as a tunnel to resolve other routes



Note:

If the BGP-LU IPv4 route is re-advertised with a new next-hop, the **route-table-install false** policy action does not prevent a new Incoming Label Map (ILM) label from being allocated for the route and programmed into the ILM tables of the line cards.

- when a BGP-LU IPv6 route is received from a base router BGP peer, the route is:
 - not installed in the RTM
 - not downloaded to the IOMs for installation in the FIB tables
 - not available for CPM routing (for example, for control plane traffic)
 - not available to resolve other routes

Usual BGP rules do not allow the advertising of inactive routes when **advertise-inactive** is not configured. However, routes marked by the **route-table-install false** policy action can be re-advertised, even if **advertise-inactive** is not configured toward the RIB-OUT peer and even if **next-hop-self true** is configured toward the RIB-OUT peer. Because of the latter, incorrect use of this feature can blackhole traffic.



Note:

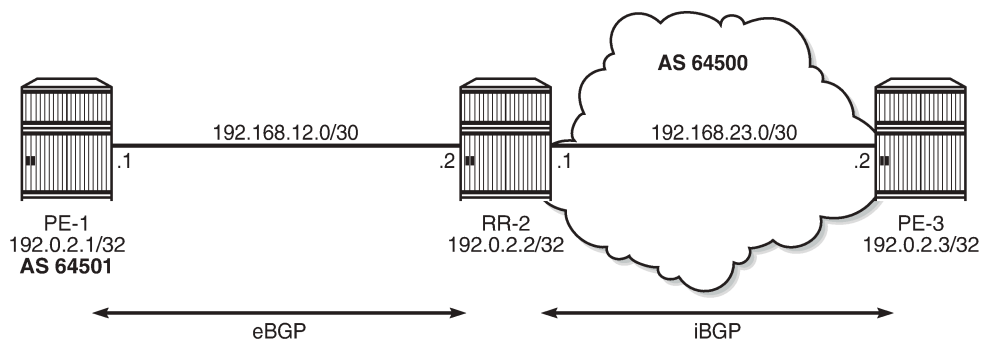
The **route-table-install false** command at the BGP instance level does not allow a route to be advertised under next-hop-self conditions.

The **route-table-install false** policy action overrides the effect of the **selective-label-ipv4-install true** command. Even if a /32 BGP-LU route should be installed in the route table and tunnel table because it has a dependent service, the **route-table-install false** policy action suppresses the installation.

Configuration

Figure 167: Example topology shows the example topology for this feature.

Figure 167: Example topology



36185

The initial configuration on the nodes includes:

- Cards, MDAs, ports
- Router interfaces
- SR-ISIS (on RR-2 and PE-3 in AS 64500)

An eBGP session is established between PE-1 in AS 64501 and RR-2 in AS 64500, and an iBGP session between RR-2 and PE-3 in AS 64500 with **next-hop-self true**. The BGP configuration on RR-2 is as follows:

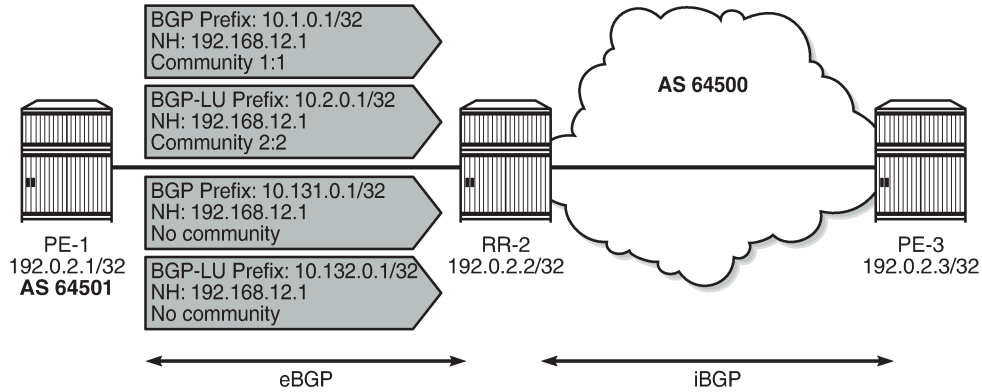
```
# on RR-2:
configure {
  router "Base" {
    bgp {
      split-horizon true
      ebgp-default-reject-policy {
        import true          # default
        export true         # default
      }
      next-hop-resolution {
        labeled-routes {
          transport-tunnel {
            family label-ipv4 {
              resolution-filter {
                ldp false
                sr-isis true
              }
            }
          }
        }
      }
    }
    group "eBGP" {
      peer-as 64501
      local-as {
        as-number 64500
      }
    }
    group "iBGP-IPv4" {
      peer-as 64500
      family {
        ipv4 true
        label-ipv4 true
      }
      cluster {
        cluster-id 192.0.2.2
      }
    }
    neighbor "192.0.2.3" {
      group "iBGP-IPv4"
      next-hop-self true
    }
    neighbor "192.168.12.1" {
      group "eBGP"
      next-hop-self true
      family {
        ipv4 true
        label-ipv4 true
      }
    }
  }
}
```

Figure 168: PE-1 exports BGP IPv4 and BGP-LU IPv4 routes to RR-2 shows that PE-1 advertises two BGP IPv4 routes and two BGP-LU IPv4 routes to RR-2:

- BGP route 10.1.0.1/32 with community 1:1
- BGP-LU route 10.2.0.1/32 with community 2:2
- BGP route 10.131.0.1/32 without community

- BGP-LU route 10.132.0.1/32 without community

Figure 168: PE-1 exports BGP IPv4 and BGP-LU IPv4 routes to RR-2



36186

On PE-1, the following export policies are applied for BGP neighbor 192.168.12.2:

```
# on PE-1:
configure {
  policy-options {
    community "1:1" {
      member "1:1" { }
    }
    community "2:2" {
      member "2:2" { }
    }
    prefix-list "10.1.0.0/16" {
      prefix 10.1.0.0/16 type longer {
      }
    }
    prefix-list "10.131.0.0/16" {
      prefix 10.131.0.0/16 type longer {
      }
    }
    prefix-list "10.132.0.0/16" {
      prefix 10.132.0.0/16 type longer {
      }
    }
    prefix-list "10.2.0.0/16" {
      prefix 10.2.0.0/16 type longer {
      }
    }
  }
  policy-statement "export-10.1" {
    entry 10 {
      from {
        prefix-list ["10.1.0.0/16"]
      }
      to {
        protocol {
          name [bgp]
        }
      }
      action {
        action-type accept
        community {
          add ["1:1"]
        }
      }
    }
  }
}
```

```
    }
  }
}
policy-statement "export-10.131" {
  entry 10 {
    from {
      prefix-list ["10.131.0.0/16"]
    }
    to {
      protocol {
        name [bgp]
      }
    }
    action {
      action-type accept
    }
  }
}
policy-statement "export-10.132" {
  entry 10 {
    from {
      prefix-list ["10.132.0.0/16"]
    }
    to {
      protocol {
        name [bgp-label]
      }
    }
    action {
      action-type accept
    }
  }
}
policy-statement "export-10.2" {
  entry 10 {
    from {
      prefix-list ["10.2.0.0/16"]
    }
    to {
      protocol {
        name [bgp-label]
      }
    }
    action {
      action-type accept
      community {
        add ["2:2"]
      }
    }
  }
}
}
router "Base" {
  autonomous-system 64501
  bgp {
    split-horizon true
    group "eBGP" {
      peer-as 64500
      local-as {
        as-number 64501
      }
    }
  }
  neighbor "192.168.12.2" {
```

```

    group "eBGP"
    next-hop-self true
    family {
        ipv4 true
        label-ipv4 true
    }
    export {
        policy ["export-10.1" "export-10.2" "export-10.131" "export-10.132"]
    }
}
}

```

Initially, RR-2 has no import policy matching any of these four routes. The following BGP routes are received on RR-2:

```

[]
A:admin@RR-2# show router bgp routes
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)        Path-Id    IGP Cost
      As-Path                 Label
-----
i     10.1.0.1/32              None       None
      192.168.12.1           None       0
      64501                   -
i     10.131.0.1/32          None       None
      192.168.12.1          None       0
      64501                   -
-----
Routes : 2
=====

```

By default, all eBGP routes are rejected because no import policy is configured (**ebgp-default-reject-policy import true**), so the routes get the flags "Invalid IGP Rejected":

```

[]
A:admin@RR-2# show router bgp routes hunt | match "Flags"
Flags      : Invalid IGP Rejected
Flags      : Invalid IGP Rejected

```

The following BGP-LU routes are received on RR-2:

```

[]
A:admin@RR-2# show router bgp routes label-ipv4
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

```

```

=====
BGP Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                   Path-Id    IGP Cost
      As-Path                               Label
-----
i     10.2.0.1/32                            None       None
      192.168.12.1                          None       0
      64501                                             524287
i     10.132.0.1/32                         None       None
      192.168.12.1                          None       0
      64501                                             524287
-----
Routes : 2
=====

```

These BGP-LU routes are also rejected, as follows:

```

[]
A:admin@RR-2# show router bgp routes label-ipv4 hunt | match "Flags"
Flags          : Invalid IGP Rejected
Flags          : Invalid IGP Rejected

```

None of these invalid routes is installed in the Routing Table Manager (RTM) and none of these routes will be re-advertised by RR-2 to PE-3.

route-table-install false policy action

On RR-2, an import policy is configured that only accepts and installs BGP routes with community "1:1" or "2:2"; all other routes match the policy **default-action accept route-table-install false**.

BGP IPv4 route 10.1.0.1/32 will be installed in the route table and the BGP-LU IPv4 route 10.131.0.1 will be installed in the route table and tunnel table. However, BGP IPv4 route 10.131.0.1/32 will not be installed in the route table and BGP-LU IPv4 route 10.132.0.1/32 will not be installed in the route table and tunnel table. Suppression of BGP route installation in the RTM and in the Tunnel Table Manager (TTM) can be done when the router has other ways of forwarding the associated traffic; in this example, via a static route 10.128.0.0/9.

```

# on RR-2:
configure {
  policy-options {
    community "1:1" {
      member "1:1" { }
    }
    community "2:2" {
      member "2:2" { }
    }
  }
  policy-statement "bgp-install-1:1-2:2" {
    entry 10 {
      from {
        community {
          name "1:1"
        }
      }
      action {
        action-type accept
      }
    }
  }
}

```

```

    }
    entry 20 {
        from {
            community {
                name "2:2"
            }
        }
        action {
            action-type accept
        }
    }
    default-action {
        action-type accept
        route-table-install false
    }
}
}
router "Base"
  bgp {
    group "eBGP" {
        peer-as 64501
        local-as {
            as-number 64500
        }
    }
    neighbor "192.168.12.1" {
        group "eBGP"
        next-hop-self true
        family {
            ipv4 true
            label-ipv4 true
        }
        import {
            policy ["bgp-install-1:1-2:2"]
        }
    }
  }
  static-routes {
    route 10.128.0.0/9 route-type unicast {
        next-hop "192.168.12.1" {
            admin-state enable
        }
    }
  }
}

```

With this import policy, BGP route 10.1.0.1/32 is active, but route 10.131.0.1/32 is inactive, as follows:

```

[]
A:admin@RR-2# show router bgp routes
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                       Path-Id    IGP Cost
      As-Path                                Label

```



```
-----
u*>i 10.1.0.1/32          None      None
      192.168.12.1       None      0
      64501                -         -
*>i 10.131.0.1/32       None      None
      192.168.12.1       None      0
      64501                -         -
-----
Routes : 2
=====
```

In a similar way, BGP-LU IPv4 route 10.2.0.1/32 is active, but route 10.132.0.1/32 is inactive:

```
[ ]
A:admin@RR-2# show router bgp routes label-ipv4
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
u*>i 10.2.0.1/32          None      None
      192.168.12.1       None      0
      64501                -         524287
*>i 10.132.0.1/32       None      None
      192.168.12.1       None      0
      64501                -         524287
-----
Routes : 2
=====
```

BGP route 10.131.0.1/32 and BGP-LU route 10.132.0.1/32 have the flag "Disable-RTM-Install" set, but both routes are advertised to the RIB-OUT peer PE-3, as follows:

```
[ ]
A:admin@RR-2# show router bgp routes hunt
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
RIB In Entries
-----
Network      : 10.1.0.1/32
Nexthop      : 192.168.12.1
---snip---
```

```
Community      : 1:1
---snip---
Flags          : Used Valid Best IGP
---snip---

Network        : 10.131.0.1/32
Nexthop        : 192.168.12.1
---snip---
Community      : No Community Members
---snip---
Flags          : Valid Best IGP Disable-RTM-Install
---snip---
-----
RIB Out Entries
-----
Network        : 10.1.0.1/32
Nexthop        : 192.0.2.2
---snip---
Community      : 1:1
---snip---

Network       : 10.131.0.1/32
Nexthop        : 192.0.2.2
---snip---
Community      : No Community Members
---snip---
```

```
[ ]
A:admin@RR-2# show router bgp routes label-ipv4 hunt
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP Routes
=====
-----
RIB In Entries
-----
Network        : 10.2.0.1/32
Nexthop        : 192.168.12.1
---snip---
Community      : 2:2
---snip---
Flags          : Used Valid Best IGP
---snip---

Network        : 10.132.0.1/32
Nexthop        : 192.168.12.1
---snip---
Community      : No Community Members
---snip---
Flags          : Valid Best IGP Disable-RTM-Install
---snip---
-----
RIB Out Entries
-----
```

```

Network      : 10.2.0.1/32
NextHop     : 192.0.2.2
---snip---
Community   : 2:2
---snip---
Network    : 10.132.0.1/32
NextHop     : 192.0.2.2
---snip---
Community   : No Community Members
---snip---

```

On RR-2, the route table only has one BGP route and one BGP-LU route, as follows:

```

[]
A:admin@RR-2# show router route-table protocol bgp

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
  Next Hop[Interface Name]                Metric
-----
10.1.0.1/32                        Remote BGP    00h13m48s 170
    192.168.12.1                      0
-----
No. of Routes: 1

```

```

[]
A:admin@RR-2# show router route-table protocol bgp-label

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
  Next Hop[Interface Name]                Metric
-----
10.2.0.1/32                        Remote BGP_LABEL 00h13m48s 170
    192.168.12.1                      0
-----
No. of Routes: 1

```

On RR-2, the FIB contains BGP route 10.1.0.1/32, BGP-LU route 10.2.0.1/32, and static route 10.128.0.0/9:

```

[]
A:admin@RR-2# show router fib 1 ip-prefix-prefix-length 10.0.0.0/8 longer

=====
FIB Display
=====
Prefix [Flags]                    Protocol
  NextHop
-----
10.1.0.1/32                        BGP
    192.168.12.1 (int-RR-2-PE-1)
10.2.0.1/32                        BGP_LABEL
    192.168.12.1 (int-RR-2-PE-1)
10.128.0.0/9                       STATIC
    192.168.12.1 (int-RR-2-PE-1)
-----
Total Entries : 3

```

On RR-2, the tunnel table contains a BGP tunnel toward destination 10.2.0.1/32, but no tunnel toward destination 10.132.0.1/32, as follows:

```
[ ]
A:admin@RR-2# show router tunnel-table protocol bgp

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
10.2.0.1/32          bgp        MPLS  262145   12   192.168.12.1  1000
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
```

RR-2 advertises both the active and the inactive/suppressed routes to RIB-OUT peer PE-3. The result is that, on PE-3, the route table contains both BGP routes and both BGP-LU routes:

```
[ ]
A:admin@PE-3# show router route-table protocol bgp

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age           Pref
  Next Hop[Interface Name]  Metric
-----
10.1.0.1/32                Remote BGP      00h11m38s  170
  192.168.23.1              10
10.131.0.1/32              Remote BGP      00h11m38s  170
  192.168.23.1              10
-----
No. of Routes: 2
```

```
[ ]
A:admin@PE-3# show router route-table protocol bgp-label

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age           Pref
  Next Hop[Interface Name]  Metric
-----
10.2.0.1/32                Remote BGP_LABEL 00h11m38s  170
  192.0.2.2 (tunneled:SR-ISIS:0)  10
10.132.0.1/32              Remote BGP_LABEL 00h11m38s  170
  192.0.2.2 (tunneled:SR-ISIS:0)  10
-----
No. of Routes: 2
```

route-table-install false command

The **route-table-install false** command in the BGP global context is mainly used for off-path route reflectors that do not participate in traffic forwarding.

This section describes the **route-table-install false** command in the general **bgp** context, in combination with the **route-table-install false** parameter, which is part of the policy framework (**action** or **default-action**).

The **route-table-install false** command in the general BGP context is configured as follows:

```
# on RR-2:
configure {
  router "Base" {
    bgp {
      route-table-install false
    }
  }
}
```

The rest of the BGP configuration (including import policy) remains unchanged.

This **route-table-install false** command applies to all received BGP routes, so none of the BGP and BGP-LU routes received from PE-1 will be installed in the RTM and TTM. Therefore, all BGP and BGP-LU routes are inactive (in this example, the second route was already inactive because of the import policy).

```
[ ]
A:admin@RR-2# show router bgp routes
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
*>i  10.1.0.1/32                None       None
      192.168.12.1           None       0
      64501                   -
*>i  10.131.0.1/32           None       None
      192.168.12.1           None       0
      64501                   -
-----
Routes : 2
=====
```

```
[ ]
A:admin@RR-2# show router bgp routes label-ipv4
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
```

```

=====
BGP Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)    Path-Id    IGP Cost
      As-Path
-----
*>i  10.2.0.1/32            None       None
      192.168.12.1        None       0
      64501                524287
*>i  10.132.0.1/32         None       None
      192.168.12.1        None       0
      64501                524287
-----
Routes : 2
=====

```

When a BGP route is suppressed because of a **route-table-install false** general BGP command match, no flag is added. The "Disable-RTM-Install" flag is only present for the route when the **route-table-install false** policy action is matched. The following output shows that the first route did not get an additional flag:

```

[]
A:admin@RR-2# show router bgp routes hunt | match Flags
Flags          : Valid Best IGP                #for BGP route 10.1.0.1/32
Flags          : Valid Best IGP Disable-RTM-Install #for BGP-LU route 10.131.0.1/32

```

```

[]
A:admin@RR-2# show router bgp routes label-ipv4 hunt | match Flags
Flags          : Valid Best IGP                #for BGP route 10.2.0.1/32
Flags          : Valid Best IGP Disable-RTM-Install #for BGP-LU route 10.132.0.1/32

```

When the **route-table-install false** command is configured and **next-hop-self true** is configured toward the RIB-OUT peer, no BGP routes can be advertised for routes that are not installed in the RTM. In this example, the RIB-OUT toward PE-3 remains empty, as follows (the total number of routes equals the number of routes in the RIB-IN):

```

[]
A:admin@RR-2# show router bgp routes hunt | match "RIB Out Entries"
                                                    pre-lines 2 post-lines 50
-----
RIB Out Entries
-----
Routes : 2
=====

```

```

[]
A:admin@RR-2# show router bgp routes label-ipv4 hunt | match "RIB Out Entries"
                                                    pre-lines 2 post-lines 50
-----
RIB Out Entries
-----
Routes : 2
=====

```

Conclusion

The **route-table-install false** policy action in a BGP import policy allows the marking of a route with a "Disable-RTM-Install" flag and still re-advertises this route to RIB-OUT peers, even when **next-hop-self true** is configured. Other routers in the network can install these routes in the route table and FIB.

Separate BGP RIBs for Labeled Routes

This chapter provides information about separate border gateway protocol (BGP) route information bases (RIBs) for labeled-unicast routes.

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 14.0.R4, but the MD-CLI in the current edition corresponds to SR OS Release 20.7.R2.

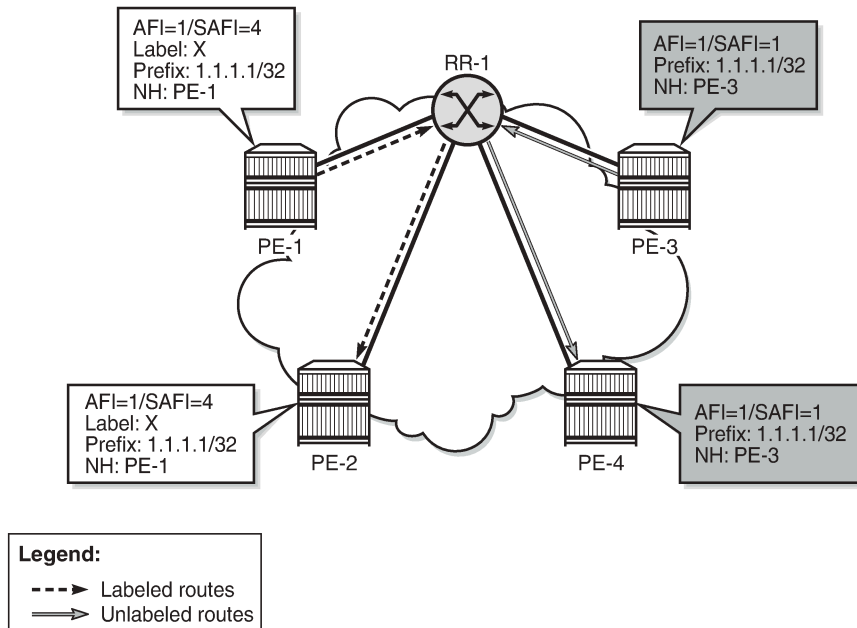
Release 14.0.R4 introduced separate BGP RIBs for labeled-unicast routes.

Overview

BGP separate labeled-IPv4 RIB implementation

[Figure 169: RR-1 with separate labeled-IPv4 RIB implementation](#) shows how RR-1 sends a labeled-IPv4 route to PE-2 with label X and next hop PE-1.

Figure 169: RR-1 with separate labeled-IPv4 RIB implementation

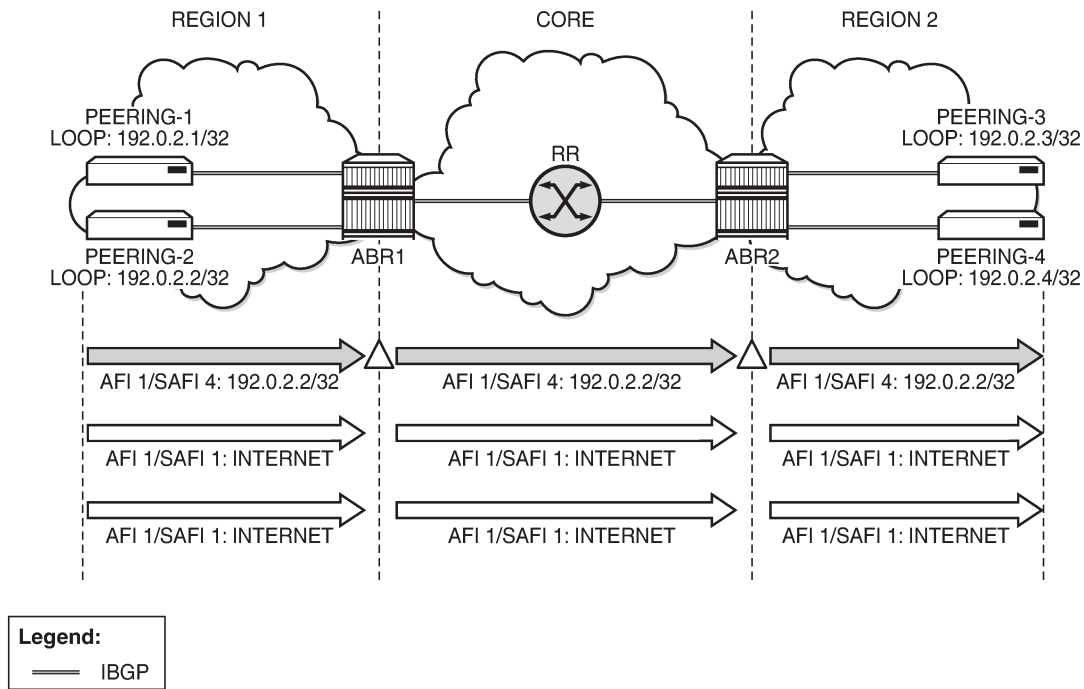


25972

In SR OS Release 14.0.R4, and later, a separate RIB is used for labeled-IPv4 routes. With this implementation, client PE-2 learns the best labeled-IPv4 route and client PE-4 learns the best unlabeled IPv4 route. RR-1 does not need to set next-hop-self and traffic can be sent directly from PE-2 to PE-1 and from PE-4 to PE-3. The RR is used only for control traffic, as intended.

Figure 170: Seamless MPLS - Separate labeled-IPv4 implementation shows a seamless MPLS use case, which is a good example of the coexistence of labeled (AFI 1/SAFI 4) and unlabeled (AFI 1/SAFI 1) BGP sessions.

Figure 170: Seamless MPLS - Separate labeled-IPv4 implementation

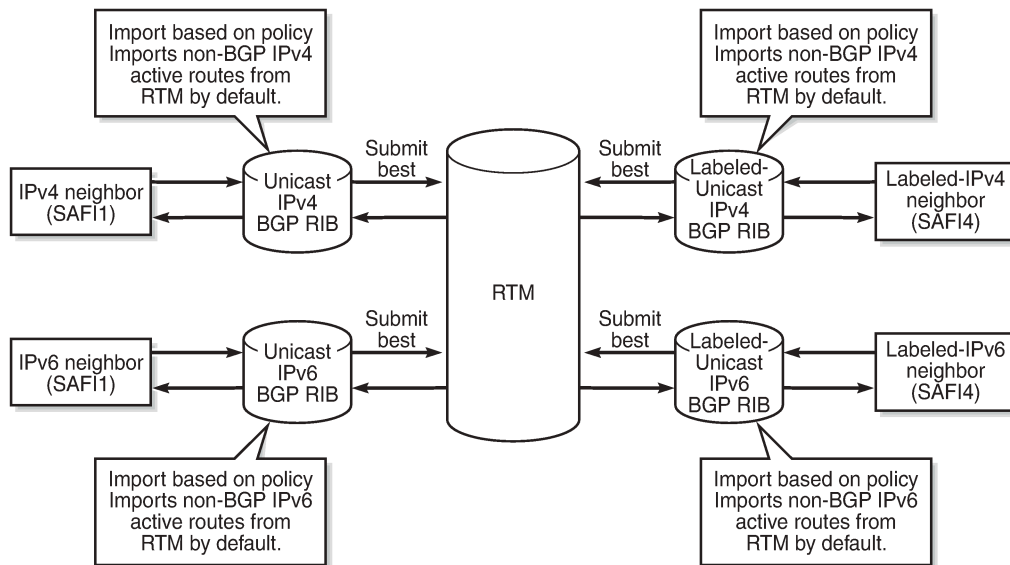


25973

RIB architecture

Figure 171: System architecture with separate RIBs for labeled-unicast and unlabeled routes shows the system architecture with four separate RIBs for IPv4 and IPv6 routes.

Figure 171: System architecture with separate RIBs for labeled-unicast and unlabeled routes



25974

Labeled-unicast routes from peers are stored in a labeled RIB and unlabeled routes from the same or different peers are stored in a non-labeled RIB. Both labeled and unlabeled routes can be sent and received to and from the same peer. Different sets of routes can be advertised to labeled/unlabeled peers. Labeled and unlabeled BGP sessions are using the common equal cost multipath (ECMP) and multipath limit.

More user control is provided over the RTM route import process. By default, a RIB imports all non-BGP active routes from RTM, but a user-defined route policy can be applied. Route policies can be used to reduce BGP memory usage.

The address families mapped to the RIBs are: **ipv4**, **label-ipv4**, **ipv6**, **label-ipv6**. In route policies, protocol types **bgp** and **bgp-label** can be used.

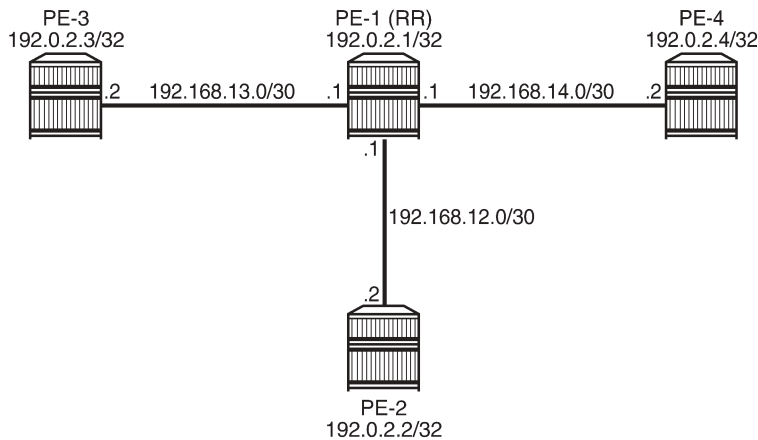
The default RTM preference for labeled IP routes is configurable (**label-preference**) in the **bgp** context of the base router or a VPRN. The default preference is 170.

Configuration

All the examples are based on labeled and unlabeled IPv4 addresses. For IPv6, the configuration is similar.

[Figure 172: Example IPv4 topology](#) shows the example topology using IPv4 addresses.

Figure 172: Example IPv4 topology



25975

The initial configuration includes:

- Cards, MDAs, ports
- Router interfaces
- IS-IS in AS 64500 (PE-1, PE-2, PE-4)
- LDP in AS 64500
- Loopback addresses 3.3.3.3/32 in PE-3 and 4.4.4.4/32 in PE-4
- Export policy "export-bgp" accepting routes from protocol direct on all nodes
- Import policy "import-bgp" accepting BGP routes on PE-1

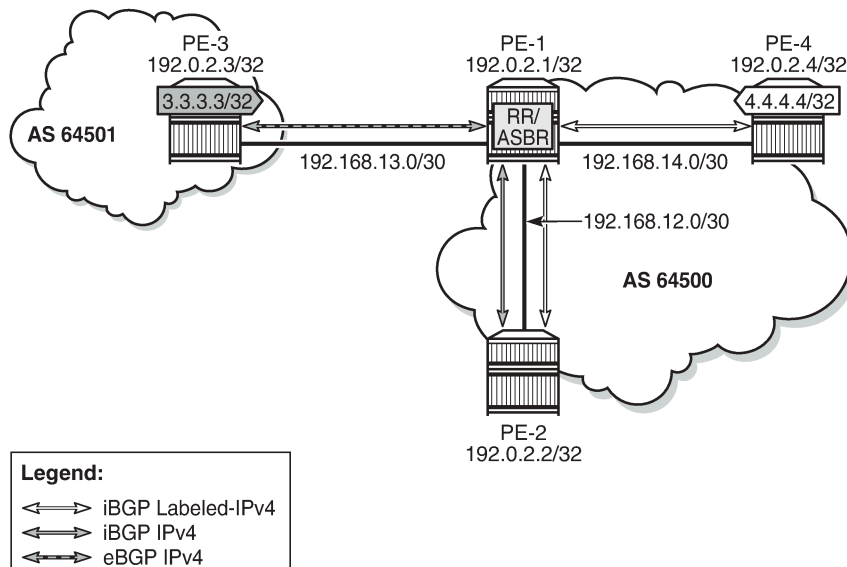
The following will be configured and verified:

1. Coexistence of labeled and unlabeled address families for BGP
2. Applying next-hop-self
3. Export policy to advertise route as labeled/unlabeled
4. Behavior of RR with a mix of labeled and unlabeled iBGP sessions

Coexistence of labeled and unlabeled address families for BGP

Figure 173: BGP sessions shows the eBGP and iBGP sessions that are established between the nodes and the routes advertised for the loopback addresses.

Figure 173: BGP sessions



25976

PE-1 acts as RR for PE-2 and PE-4, and it is an autonomous system border router (ASBR) toward PE-3. PE-1 has two single-family connections: unlabeled IPv4 to PE-3 and labeled IPv4 to PE-4. PE-1 also has one dual-family connection to PE-2. The BGP configuration on PE-1 is as follows:

```
# on PE-1:
configure {
  router "Base" {
    autonomous-system 64500
    bgp {
      split-horizon true
      group "eBGP" {
        peer-as 64501
        import {
          policy ["import-bgp"]
        }
      }
      group "iBGP" {
        peer-as 64500
        cluster {
          cluster-id 192.0.2.1
        }
        export {
          policy ["export-bgp"]
        }
      }
    }
    neighbor "192.0.2.2" {
      group "iBGP"
      family {
        ipv4 true
        label-ipv4 true
      }
    }
    neighbor "192.168.13.2" {
      group "eBGP"
      family {
        ipv4 true
      }
    }
  }
}
```

```
    }  
  }  
  neighbor "192.0.2.4" {  
    group "iBGP"  
    family {  
      label-ipv4 true  
    }  
  }  
}
```

The BGP configuration on PE-2 is as follows:

```
# on PE-2:  
configure {  
  router "Base" {  
    autonomous-system 64500  
    bgp {  
      split-horizon true  
      group "iBGP" {  
        peer-as 64500  
        export {  
          policy ["export-bgp"]  
        }  
      }  
      neighbor "192.0.2.1" {  
        group "iBGP"  
        family {  
          ipv4 true  
          label-ipv4 true  
        }  
      }  
    }  
  }  
}
```

The BGP configuration on PE-3 in AS 64501 is as follows:

```
# on PE-3:  
configure {  
  router "Base" {  
    autonomous-system 64501  
    bgp {  
      split-horizon true  
      export {  
        policy ["export-bgp"]  
      }  
      group "eBGP" {  
        peer-as 64500  
      }  
      neighbor "192.168.13.1" {  
        group "eBGP"  
        family {  
          ipv4 true  
        }  
      }  
    }  
  }  
}
```

The BGP configuration on PE-4 is as follows:

```
# on PE-4:  
configure {  
  router "Base" {  
    autonomous-system 64500  
    bgp {  
      split-horizon true  
      group "iBGP" {
```

```

peer-as 64500
export {
  policy ["export-bgp"]
}
neighbor "192.0.2.1" {
  group "iBGP"
  family {
    label-ipv4 true
  }
}

```

The BGP summary on PE-1 shows that there is a dual-family connection with PE-2: IPv4 and Lbl-IPv4. PE-1 has an Lbl-IPv4 connection with PE-4 and an IPv4 connection with PE-3.

```

[]
A:admin@PE-1# show router bgp summary all
=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId
          AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
          PktSent OutQ
-----
192.0.2.2
Def. Instance 64500      5   0 00h00m25s 2/0/6 (IPv4)
                8   0                2/0/5 (Lbl-IPv4)
192.0.2.4
Def. Instance 64500      5   0 00h00m32s 3/1/4 (Lbl-IPv4)
                7   0
192.168.13.2
Def. Instance 64501      5   0 00h00m43s 3/2/0 (IPv4)
                5   0
-----

```

The unlabeled IPv4 routes on PE-1 include unlabeled routes imported from PE-2 and PE-3, including the loopback address 3.3.3.3/32 advertised by PE-3, as follows:

```

[]
A:admin@PE-1# show router bgp routes ipv4
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                   Path-Id    IGP Cost
      As-Path                             Label
-----
u*>i 3.3.3.3/32                             None       None
      192.168.13.2                         None       0

```

```

64501
*i 192.0.2.2/32          100      None
   192.0.2.2           None      10
   No As-Path          -
u*>i 192.0.2.3/32      None      None
     192.168.13.2     None      0
     64501             -
*i 192.168.12.0/30    100      None
   192.0.2.2           None      10
   No As-Path          -
*i 192.168.13.0/30   None      None
   192.168.13.2       None      0
   64501               -
-----
Routes : 5
=====

```

The labeled-unicast IPv4 routes on PE-1 include labeled routes imported from PE-2 and PE-4, including the loopback address 4.4.4.4/32 advertised by PE-4, as follows:

```

[]
A:admin@PE-1# show router bgp routes label-ipv4
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP Routes
=====
Flag Network                LocalPref  MED
     Nexthop (Router)      Path-Id    IGP Cost
     As-Path                Label
-----
u*>i 4.4.4.4/32              100        None
     192.0.2.4              None        10
     No As-Path              524284
*i 192.0.2.2/32            100        None
   192.0.2.2                 None        10
   No As-Path                 524285
*i 192.0.2.4/32            100        None
   192.0.2.4                 None        10
   No As-Path                 524284
*i 192.168.12.0/30         100        None
   192.0.2.2                 None        10
   No As-Path                 524285
*i 192.168.14.0/30         100        None
   192.0.2.4                 None        10
   No As-Path                 524284
-----
Routes : 5
=====

```

PE-2 imports the prefix 3.3.3.3/32 in its unlabeled RIB, as follows:

```

[]
A:admin@PE-2# show router bgp routes 3.3.3.3/32
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====

```



```

=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====

BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
  Nexthop (Router)                         Path-Id   IGP Cost
  As-Path                                   Label
-----
u*>i 3.3.3.3/32                               100      None
      192.168.13.2                          None     20
      64501                                   -
-----
Routes : 1
=====

```

PE-2 imports the prefix 4.4.4.4/32 in its labeled RIB, as follows:

```

[]
A:admin@PE-2# show router bgp routes 4.4.4.4/32 label-ipv4
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====

BGP Routes
=====
Flag Network                               LocalPref MED
  Nexthop (Router)                         Path-Id   IGP Cost
  As-Path                                   Label
-----
u*>i 4.4.4.4/32                               100      None
      192.0.2.4                          None     20
      No As-Path                           524284
-----
Routes : 1
=====

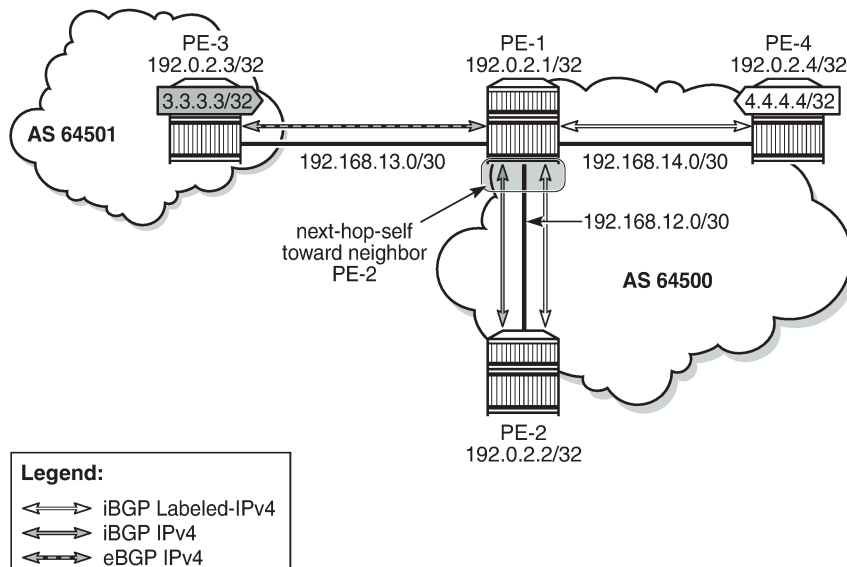
```

As expected, the prefixes from address family label-ipv4 are advertised independently from the prefixes from address family ipv4.

Applying next-hop-self

[Figure 174: PE-1 applies next-hop-self toward neighbor PE-2](#) shows that PE-1 applies next-hop-self for BGP updates toward PE-2.

Figure 174: PE-1 applies next-hop-self toward neighbor PE-2



25977

On PE-1, next-hop-self is enabled for neighbor PE-2 only, as follows:

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      neighbor 192.0.2.2 {
        next-hop-self true
      }
    }
  }
}
```

This applies to both address families. The next hop for unlabeled route 3.3.3.3/32 will be 192.0.2.1, as follows:

```
[ ]
A:admin@PE-2# show router bgp routes 3.3.3.3/32
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)        Path-Id    IGP Cost
      As-Path                  Label
-----
u*>i  3.3.3.3/32                 100        None
      192.0.2.1                None        10
      64501                     -
-----
Routes : 1
```

The labeled-unicast route 4.4.4.4/32 also has next hop 192.0.2.1, as follows:

```
[ ]
A:admin@PE-2# show router bgp routes 4.4.4.4/32 label-ipv4
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                       Path-Id    IGP Cost
      As-Path                                Label
-----
u*>i  4.4.4.4/32                               100        None
      192.0.2.1                               None       10
      No As-Path                               524283
-----
Routes : 1
=====
```

On PE-1, the next-hop-self configuration for neighbor PE-2 is removed as follows:

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      neighbor 192.0.2.2 {
        delete next-hop-self
      }
    }
  }
}
```

An export policy is configured to ensure that next-hop-self is only applied for address family ipv4. The route policy is configured as follows:

```
# on PE-1:
configure {
  policy-options {
    policy-statement "export-nhs" {
      entry 10 {
        from {
          protocol {
            name [bgp]
          }
        }
        action {
          action-type accept
          next-hop self
        }
      }
      entry 20 {
        from {
          protocol {
            name [bgp-label]
          }
        }
        action {
```

```

        action-type accept
    }
}
}

```

On PE-1, the export policy "export-nhs" is configured for neighbor PE-2:

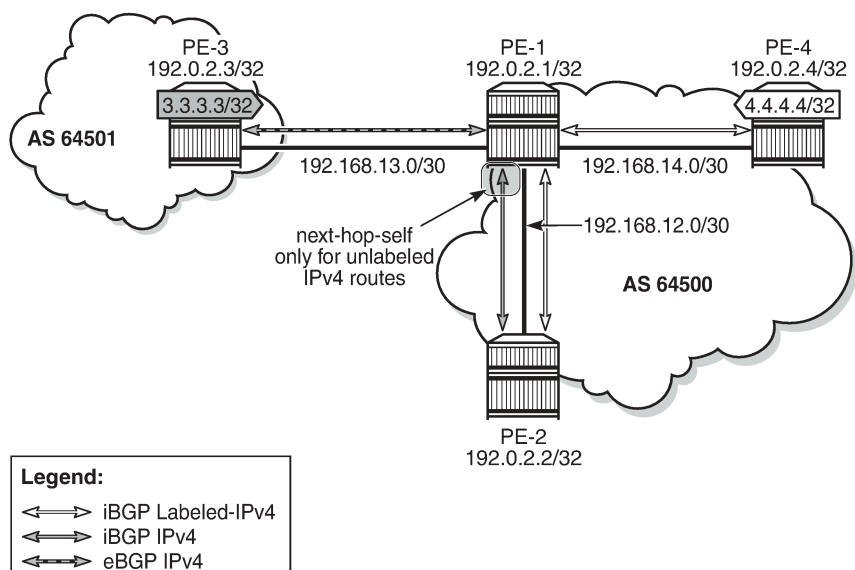
```

# on PE-1:
configure {
  router "Base" {
    bgp {
      neighbor "192.0.2.2" {
        export {
          policy ["export-nhs"]
        }
      }
    }
  }
}

```

Figure 175: Applying next-hop-self to unlabeled IP-4 routes to neighbor PE-2 shows that next-hop-self is applied to unlabeled IPv4 routes only.

Figure 175: Applying next-hop-self to unlabeled IP-4 routes to neighbor PE-2



25978

With this export policy, only the unlabeled route 3.3.3.3/32 will have next hop 192.0.2.1, while the labeled-unicast route 4.4.4.4/32 will have next hop 192.0.2.4, as follows:

```

[]
A:admin@PE-2# show router bgp routes 3.3.3.3/32
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes

```

```

=====
Flag Network                               LocalPref MED
      Nexthop (Router)                    Path-Id   IGP Cost
      As-Path                               Label
-----
u*>i 3.3.3.3/32                             100      None
      192.0.2.1                             None     10
      64501                                  -
-----
Routes : 1
=====

```

```

[]
A:admin@PE-2# show router bgp routes 4.4.4.4/32 label-ipv4
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP Routes
=====
Flag Network                               LocalPref MED
      Nexthop (Router)                    Path-Id   IGP Cost
      As-Path                               Label
-----
u*>i 4.4.4.4/32                             100      None
      192.0.2.4                             None     20
      No As-Path                            524284
-----
Routes : 1
=====

```

The export policy "export-nhs" toward neighbor PE-2 is removed as follows:

```

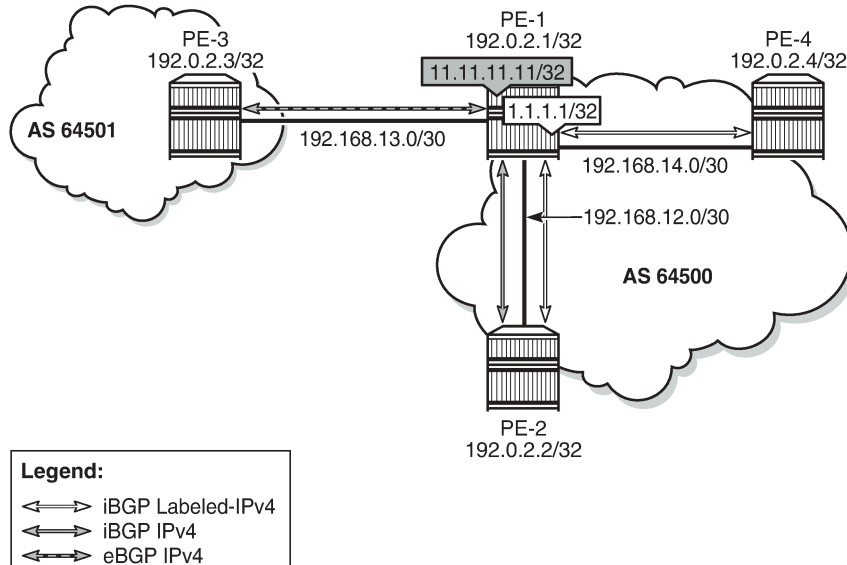
# on PE-1:
configure {
  router "Base" {
    bgp {
      neighbor "192.0.2.2" {
        delete export
      }
    }
  }
}

```

Export policy to advertise route as labeled/unlabeled

Figure 176: PE-1 advertises prefixes 1.1.1.1/32 and 11.11.11.11/32 shows that two loopback addresses are configured in PE-1 to be advertised: prefix 1.1.1.1/32 and 11.11.11.11/32. Initially, there is no route policy applied for a selective export as labeled or unlabeled route.

Figure 176: PE-1 advertises prefixes 1.1.1.1/32 and 11.11.11.11/32



25979

By default, these prefixes will be advertised as both labeled and unlabeled routes toward dual-family neighbor PE-2. On PE-2, the unlabeled IPv4 RIB contains prefixes 1.1.1.1/32 and 11.11.11.11/32, as follows:

```
[ ]
A:admin@PE-2# show router bgp routes
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                  l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  1.1.1.1/32                             100        None
      192.0.2.1                             None       10
      No As-Path                             -
---snip---
u*>i  11.11.11.11/32                         100        None
      192.0.2.1                             None       10
      No As-Path                             -
---snip---
```

The labeled-IPv4 RIB on PE-2 also contains prefixes 1.1.1.1/32 and 11.11.11.11/32, as follows:

```
[ ]
A:admin@PE-2# show router bgp routes label-ipv4
```

```

=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    Label
      As-Path
-----
*>i  1.1.1.1/32                             100        None
      192.0.2.1                             None       10
      No As-Path                                       524285
---snip---
*>i  11.11.11.11/32                         100        None
      192.0.2.1                             None       10
      No As-Path                                       524285
---snip---

```

In many cases, it is not required to advertise both a labeled route and an unlabeled route. The following policy is configured to advertise prefix 1.1.1.1/32 as a labeled-IPv4 route and prefix 11.11.11.11/32 as an unlabeled IPv4 route:

```

# on PE-1:
configure {
  policy-options {
    prefix-list "1.1.1.1/32" {
      prefix 1.1.1.1/32 type exact {
      }
    }
    prefix-list "11.11.11.11/32" {
      prefix 11.11.11.11/32 type exact {
      }
    }
  }
  policy-statement "export-bgp1" {
    entry 10 {
      from {
        prefix-list ["1.1.1.1/32"]
      }
      to {
        protocol {
          name [bgp-label]
        }
      }
      action {
        action-type accept
      }
    }
    entry 20 {
      from {
        prefix-list ["11.11.11.11/32"]
      }
      to {
        protocol {
          name [bgp]
        }
      }
      action {

```

```

        action-type accept
    }
}
default-action {
    action-type reject
}

```

This policy is applied on PE-1 as an export policy for neighbor PE-2, as follows:

```

# on PE-1:
configure {
    router "Base" {
        bgp {
            neighbor 192.0.2.2 {
                export {
                    policy ["export-bgp1"]
                }
            }
        }
    }
}

```

Prefix 11.11.11.11/32 is received as an unlabeled route on PE-2 and stored in the unlabeled RIB, but prefix 1.1.1.1/32 is not, as follows:

```

[]
A:admin@PE-2# show router bgp routes
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
      Nexthop (Router)                     Path-Id   IGP Cost
      As-Path                               Label
-----
u*>i 11.11.11.11/32                         100      None
      192.0.2.1                             None     10
      No As-Path                             -
-----
Routes : 1

```

On PE-2, prefix 1.1.1.1/32 is received as a labeled route and stored in the labeled-IPv4 RIB, but prefix 11.11.11.11/32 is not, as follows:

```

[]
A:admin@PE-2# show router bgp routes label-ipv4
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP Routes
=====

```

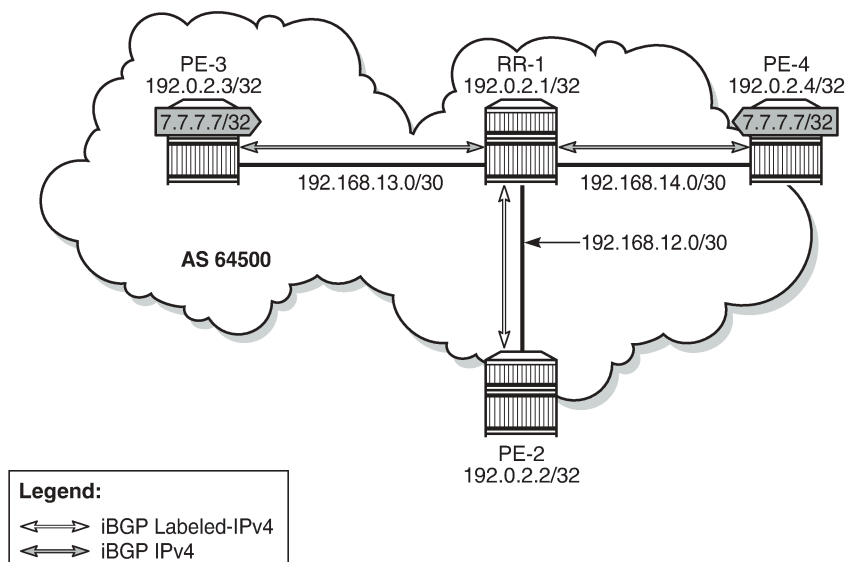

Flag	Network Nextthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
u*>i	1.1.1.1/32 192.0.2.1 No As-Path	100 None	None 10 524285
Routes : 1			

This selective route advertisement from PE-1 reduces the memory usage for the RIBs on PE-2.

RR behavior with a mix of labeled and unlabeled BGP sessions

[Figure 177: RR with labeled and unlabeled BGP sessions](#) shows a slightly different setup, with all PEs in the AS 64500 and RR-1 acting as the RR for all PEs. There are no dual-family connections. PE-3 and PE-4 have an unlabeled BGP session with RR-1 and PE-2 has a labeled BGP connection with RR-1. RR-1 has add-path=2 capability configured for neighbor PE-2. RR-1 receives the same prefix 7.7.7.7/32 from two neighbors: PE-3 and PE-4.

Figure 177: RR with labeled and unlabeled BGP sessions



25980

On RR-1, BGP is configured as follows:

```
# on RR-1:
configure {
  router "Base" {
    bgp {
      split-horizon true
      group "iBGP" {
        peer-as 64500
        cluster {
          cluster-id 192.0.2.1
        }
      }
      export {
        policy ["export-bgp"]
      }
    }
  }
}
```

```

    }
  }
  neighbor "192.0.2.2" {
    group "iBGP"
    family {
      label-ipv4 true
    }
    add-paths {
      label-ipv4 {
        send 2
        receive true
      }
    }
  }
  neighbor "192.0.2.3" {
    group "iBGP"
    family {
      ipv4 true
    }
  }
  neighbor "192.0.2.4" {
    group "iBGP"
    family {
      ipv4 true
    }
  }
}

```

RR-1 receives the prefix 7.7.7.7/32 from neighbors PE-3 and PE-4, as follows:

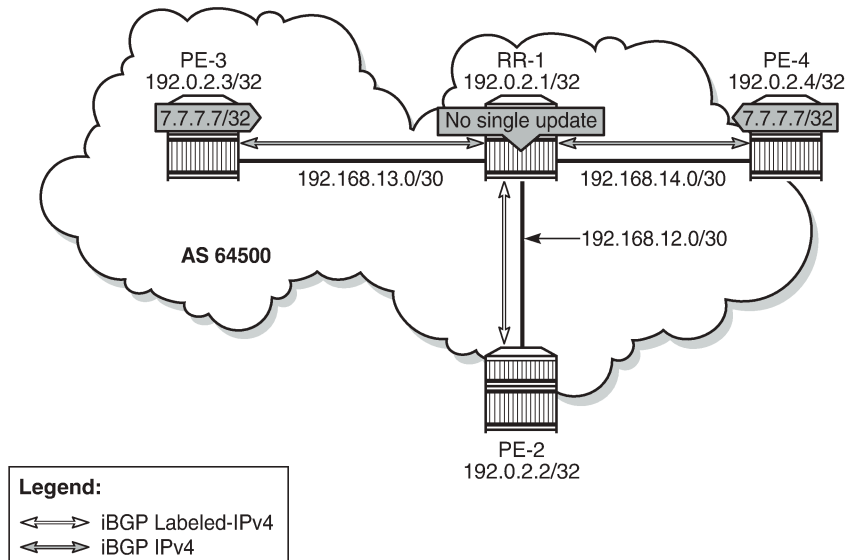
```

[]
A:admin@RR-1# show router bgp routes 7.7.7.7/32
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  7.7.7.7/32                100        None
      192.0.2.3              None        10
      No As-Path              -
*i    7.7.7.7/32                100        None
      192.0.2.4              None        10
      No As-Path              -
-----
Routes : 2

```

Both routes are unlabeled and BGP updates from unlabeled sessions are by default not exported to a labeled-IPv4 session, as shown in [Figure 178: Updates from unlabeled sessions not propagated to labeled sessions \(default\)](#).

Figure 178: Updates from unlabeled sessions not propagated to labeled sessions (default)



25981

PE-2 will not receive prefix 7.7.7.7/32, neither as unlabeled route, nor as labeled route, as follows:

```
[ ]
A:admin@PE-2# show router bgp routes 7.7.7.7/32 ipv4
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)        Path-Id    IGP Cost
      As-Path                  Label
-----
No Matching Entries Found
=====
```

```
[ ]
A:admin@PE-2# show router bgp routes 7.7.7.7/32 label-ipv4
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP Routes
=====
Flag  Network                LocalPref  MED
```

Nexthop (Router) As-Path	Path-Id	IGP Cost Label

No Matching Entries Found		
=====		

A route policy is created on RR-1 to accept both labeled and unlabeled routes, as follows:

```
# on RR-1:
configure {
  policy-options {
    policy-statement "import-all" {
      entry 10 {
        from {
          protocol {
            name [bgp]
          }
        }
        action {
          action-type accept
        }
      }
      entry 20 {
        from {
          protocol {
            name [bgp-label]
          }
        }
        action {
          action-type accept
        }
      }
    }
  }
}
```

This policy accepts all routes, labeled and unlabeled. For route 7.7.7.7/32 to be advertised to the labeled peer PE-2, it is sufficient to have a policy with only entry 10 that says from protocol bgp action accept. However, the preceding policy can also be used to import labeled routes to be advertised to unlabeled peers.

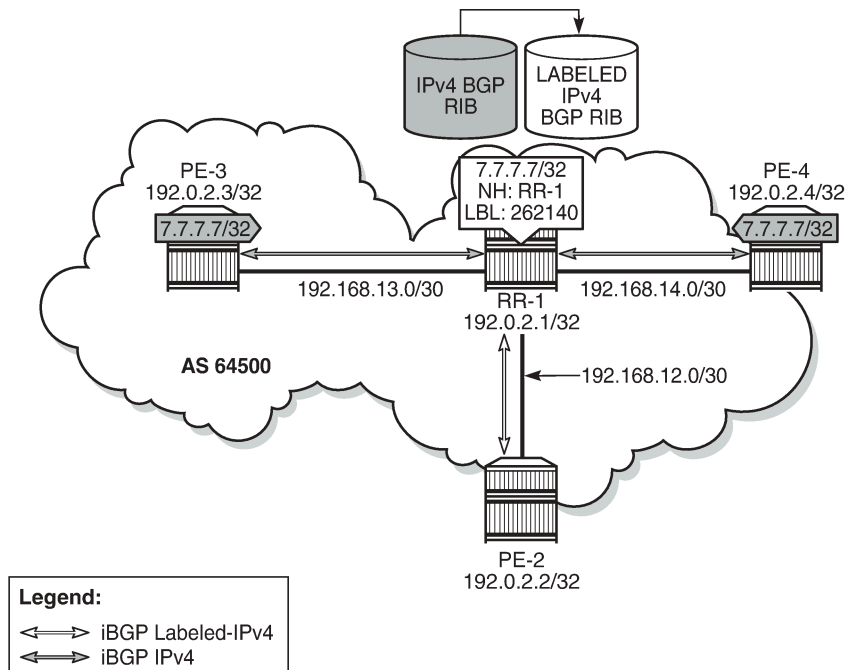
The following policy is applied as route-table-import policy in BGP RIB management, both for unlabeled IPv4 routes and labeled-IPv4 routes on RR-1:

```
# on RR-1:
configure {
  router "Base" {
    bgp {
      rib-management {
        ipv4 {
          route-table-import {
            policy-name "import-all"
          }
        }
        label-ipv4 {
          route-table-import {
            policy-name "import-all"
          }
        }
      }
    }
  }
}
```

For allowing unlabeled route 7.7.7.7/32 to be advertised on a labeled session, it is sufficient to have a route-table-import for labeled-IPv4 only. However, the configuration allows for RIB leaking in both ways: from unlabeled IPv4 BGP RIB to labeled-IPv4 BGP RIB and vice versa.

Figure 179: RIB leaking from IPv4 BGP RIB to labeled-IPv4 BGP RIB shows this RIB leaking process.

Figure 179: RIB leaking from IPv4 BGP RIB to labeled-IPv4 BGP RIB



25982

After applying this RIB leaking, RR-1 will advertise prefix 7.7.7.7/32 to PE-2. Therefore, RR-1 needs to add a label to the route and RR-1 needs to set next-hop-self. RR-1 advertises only one labeled route for prefix 7.7.7.7/32, with next hop 192.0.2.1, as follows:

```
[ ]
A:admin@RR-1# show router bgp neighbor 192.0.2.2 label-ipv4 advertised-routes
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
i     7.7.7.7/32             100        None
      192.0.2.1             None       n/a
      No As-Path              524282
---snip---
```

The BGP update message is as follows:

```
4 2020/01/15 13:41:23.925 UTC MINOR: DEBUG #2001 Base Peer 1: 192.0.2.2
"Peer 1: 192.0.2.2: UPDATE
```

```
Peer 1: 192.0.2.2 - Send BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 49
  Flag: 0x90 Type: 14 Len: 17 Multiprotocol Reachable NLRI:
    Address Family LBL-IPV4
    NextHop Len 4 NextHop 192.0.2.1
    7.7.7.7/32 Label 524282
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 0 AS Path:
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
  Flag: 0x80 Type: 9 Len: 4 Originator ID: 192.0.2.3
  Flag: 0x80 Type: 10 Len: 4 Cluster ID:
    192.0.2.1
"
```

On PE-2, the following labeled BGP route is imported:

```
[ ]
A:admin@PE-2# show router bgp routes 7.7.7.7/32 label-ipv4
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                       Path-Id    IGP Cost
      As-Path                                Label
-----
u*>i 7.7.7.7/32                               100        None
      192.0.2.1                               None       10
      No As-Path                               524282
-----
Routes : 1
```

Conclusion

The BGP RIB architecture with separate RIBs for unlabeled and labeled-unicast routes supports unlabeled sessions and labeled sessions in parallel. By default, labeled routes are not advertised to unlabeled sessions and vice versa. Route-table import policies for RIB management allow route leaking between separate RIBs: unlabeled BGP RIB and labeled-unicast BGP RIB.

MPLS

This section provides configuration information for the following topics:

- [Automatic Creation of RSVP-TE LSPs](#)
- [BFD for RSVP-TE and LDP LSPs](#)
- [BFD for RSVP-TE LSPs with Failure Action](#)
- [DiffServ Traffic Engineering](#)
- [IGP Shortcuts](#)
- [Inter-Area TE Point-to-Point LSPs](#)
- [LDP FEC to BGP Label Route Stitching](#)
- [LDP over RSVP Using OSPF as IGP](#)
- [LDP Point-to-Point LSPs](#)
- [LDP-IGP Synchronization](#)
- [LDP-SR Stitching for IPv4 Prefixes \(IS-IS\)](#)
- [MPLS LDP FRR using ISIS as IGP](#)
- [Multicast Label Distribution Protocol](#)
- [Path MTU Discovery](#)
- [Remote Loop-Free Alternate Node Protection](#)
- [RSVP Point-to-Point LSPs](#)
- [Seamless MPLS: Isolated IGP/LDP Domains and Labeled BGP](#)
- [Shared Risk Link Groups for RSVP-Based LSP](#)
- [Static Point-to-Point LSPs](#)
- [Topology-Independent Loop-Free Alternate for Link Protection](#)
- [Tunneling of ICMP Reply Packets over MPLS LSPs](#)
- [Unnumbered Interfaces in RSVP-TE and LDP](#)

Automatic Creation of RSVP-TE LSPs

This chapter provides information about automatic creation of Resource Reservation Protocol with Traffic Engineering (RSVP-TE) Label Switched Paths (LSPs).

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This feature is applicable to SR OS with no hardware constraints because this is a control-plane feature only.

This chapter was originally written for SR OS Release 11.0.R6, but the configuration in this edition corresponds to MD-CLI in SR OS Release 21.2.R1.

Overview

Automatic creation of RSVP-TE LSPs enables the automated creation of point-to-point RSVP-TE LSPs within a single Interior Gateway Protocol (IGP) Intermediate System to Intermediate System (IS-IS) level or Open Shortest Path First (OSPF) area that can subsequently be used by services and/or IGP shortcuts. The feature is divided into two components: creation of an RSVP-TE LSP mesh, and creation of single-hop RSVP-TE LSPs. Although both can be used simultaneously, it is likely that one or the other is used.

When creating an RSVP-TE LSP mesh, the mesh can be full or partial, the extent of which is governed by a prefix list containing the system addresses of all nodes that should form part of the mesh. When using single-hop RSVP-TE LSPs, point-to-point LSPs are established to all directly connected neighbors. The purpose of these single-hop LSPs is to allow for Equal Cost Multi-Path (ECMP) load balancing of traffic using LDP over RSVP, which is not possible using native RSVP LSPs.

The use of automatically created RSVP-TE LSPs avoids manual configuration of RSVP-TE LSP meshes. Even when provisioning tools—such as 5620 SAM—are used to automatically provision these LSPs, auto-mesh still provides a benefit by avoiding increased configuration file sizes.

The use of automatically created Targeted Label Distribution Protocol (T-LDP) sessions is also described when using the automatically created RSVP LSPs for Layer 2 services.

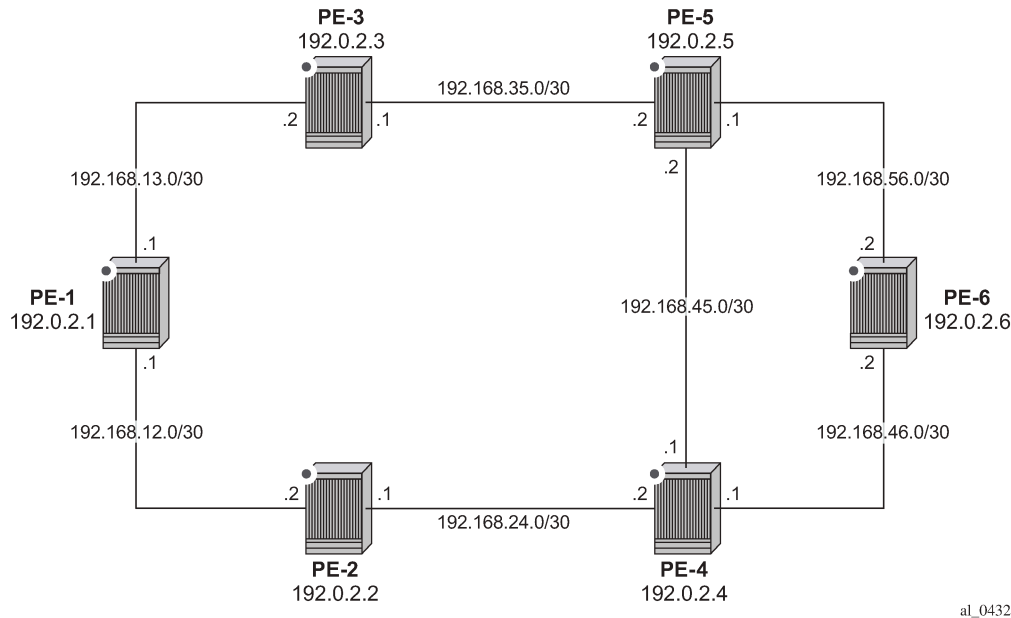
Configuration

Example Topology

The example topology is shown in [Figure 180: Example Topology](#). All routers participate in a single IS-IS Level 2 area that has traffic engineering enabled. Multi-Protocol Label Switching (MPLS) and RSVP are enabled on every interface, but no LSPs are initially provisioned. All routers are Border Gateway Protocol

(BGP) speakers and form part of Autonomous System (AS) 64496. PE-5 is a Route Reflector and the remaining routers are IBGP clients for the IPv4, VPN-IPv4, and L2-VPN address families. The objective of this example is to demonstrate how to automatically create transport LSPs using RSVP or LDP over RSVP, and then create services that utilize those LSPs. The exchange of BGP routes is needed for those services.

Figure 180: Example Topology



al_0432

Automatic Creation of an RSVP-TE LSP Mesh

To start the process of automatically creating an RSVP-TE LSP mesh, the user must create a route policy referencing a prefix-list. This prefix-list contains the system addresses of all nodes that are required to be in the mesh, and can be entered as a series of /32 addresses, or simply as a range as follows. This range encompasses all of the system addresses of the nodes in the example topology because the requirement is to make a full mesh.

```
configure {
  policy-options {
    prefix-list "System-Addresses" {
      prefix 192.0.2.0/24 type range {
        start-length 32
        end-length 32
      }
    }
  }
  policy-statement "Remote-PEs-policy" {
    entry 10 {
      from {
        prefix-list ["System-Addresses"]
      }
      action {
        action-type accept
      }
    }
  }
}
```

```
}}
```

After the route policy is created, the user must create an LSP template containing the common parameters which are used to establish all point-to-point LSPs within the mesh. For an RSVP-TE LSP mesh, the **lsp-template** must be configured with type **p2p-rsvp-mesh**. Upon creation of the template, CSPF is automatically enabled (and cannot be disabled), and the template must reference a **default-path** before it can be enabled. In the example contained in the following output, the template refers to a path named "loose-path" that has no strict or loose hops defined, meaning the system will dynamically calculate the path while considering other specified constraints. The LSP template in this output also stipulates frr-method facility bypass protection. The default behavior is no node-protect, so this configuration requests link protection only. FRR one-to-one protection is not supported for automatically created RSVP-TE LSPs; so facility bypass is the only form of protection supported.

Next, the user must associate the LSP template with the previously defined route policy, and this is accomplished using the **auto-lsp** command. In this example, the LSP template "Full-Mesh-template" is associated with the policy "Remote-PEs-policy" that in turn references a prefix-list containing all system addresses in the example topology. Up to five policies can be associated with an LSP template at the same time. If a policy associated with an LSP template is modified in order to add or remove prefixes, the system immediately re-evaluates the policy and the prefix-list to determine if one or more LSPs need to be established, or one or more LSPs need to be torn down.

```
configure {
  router "Base" {
    mpls {
      admin-state enable
      path "loose-path" {
        admin-state enable
      }
      lsp-template "Full-Mesh-template" {
        admin-state enable
        type p2p-rsvp-mesh
        default-path "loose-path"
        path-computation-method local-cspf
        fast-reroute {
          frr-method facility
        }
      }
      auto-lsp "Full-Mesh-template" {
        policy ["Remote-PEs-policy"]
      }
    }
  }
}
```

When the **auto-lsp** command is entered, the system commences the process of establishing the point-to-point LSPs. The prefixes defined in the prefix list are checked, and if a prefix corresponds to a router ID that is present in the Traffic Engineering Database (TED), the system instantiates a CSPF computed primary path to that prefix using the parameters specified in the LSP template. With the previously defined configuration applied on PE-6, the existence of point-to-point RSVP LSPs to every node in the example topology can be verified as shown in the following output. The LSP name is automatically constructed as TemplateName-DestIPv4Address-TunnelId. The LSP name signaled in the Session Attribute object concatenates the LSP name with the path name (for example Full-Mesh-template-192.0.2.1-61441::loose-path).

```
[/]
A:admin@PE-6# show router mpls lsp
```

```

=====
MPLS LSPs (Originating)
=====
LSP Name          Tun   Fastfail  Adm  Opr
To                Id     Config
-----
Full-Mesh-template-192.0.2.1-61441
192.0.2.1         61441  Yes       Up   Up
Full-Mesh-template-192.0.2.2-61442
192.0.2.2         61442  Yes       Up   Up
Full-Mesh-template-192.0.2.3-61443
192.0.2.3         61443  Yes       Up   Up
Full-Mesh-template-192.0.2.4-61444
192.0.2.4         61444  Yes       Up   Up
Full-Mesh-template-192.0.2.5-61445
192.0.2.5         61445  Yes       Up   Up
-----
LSPs : 5
=====

```

The following output on PE-6 shows that the auto-created LSP to PE-1 has primary path "loose-path" and is operationally up:

```

[/]
A:admin@PE-6# show router mpls lsp "Full-Mesh-template-192.0.2.1-61441" path detail | match
"LSP Name" post-lines 5
LSP Name      : Full-Mesh-template-192.0.2.1-61441
From          : 192.0.2.6
To            : 192.0.2.1
Admin State   : Up                               Oper State   : Up
Path Name     : loose-path
Path LSP ID   : 60928                             Path Type    : Primary

```

The LSP template requests FRR link protection. On PE-6, this protection can be verified by querying each primary LSP. In the following output, the primary LSP to PE-1 (Full-Mesh-template-192.0.2.1-61441) is signaled through PE-5 (192.0.2.5) and PE-3 (192.0.2.3), and the presence of the @ indicator after each hop denotes that link protection is available to the primary path.

```

[/]
A:admin@PE-6# show router mpls lsp "Full-Mesh-template-192.0.2.1-61441" path detail | match
"Actual Hops" post-lines 4
Actual Hops      :
  192.168.56.2(192.0.2.6) @           Record Label   : N/A
-> 192.168.56.1(192.0.2.5) @           Record Label   : 524270
-> 192.168.35.1(192.0.2.3) @           Record Label   : 524274
-> 192.168.13.1(192.0.2.1)             Record Label   : 524275

```

Finally, it can be verified that the signaled LSPs are placed in the tunnel table and made available to the tunnel table manager so they can be used by applications and services.

```

[/]
A:admin@PE-6# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner   Encap TunnelId  Pref  Nexthop      Metric
Color
-----
192.0.2.1/32 [B]  rsvp   MPLS  61441   7     192.168.56.1  30
192.0.2.2/32 [B]  rsvp   MPLS  61442   7     192.168.46.1  20

```

```

192.0.2.3/32 [B]      rsvp      MPLS    61443    7      192.168.56.1  20
192.0.2.4/32 [B]      rsvp      MPLS    61444    7      192.168.46.1  10
192.0.2.5/32 [B]      rsvp      MPLS    61445    7      192.168.56.1  10
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====

```

When the LSP template is in use and LSPs are instantiated, it is necessary to disable the template to change any parameters that cannot be handled as a Make-Before-Break (MBB). This essentially includes all LSP parameters with the exception of bandwidth and FRR without node-protection. Modification of any other parameters requires disabling the LSP template and a re-signal of the LSP once the LSP template is re-enabled. MBB is supported for timer-based and manual re-signaling of the automatically created LSPs.

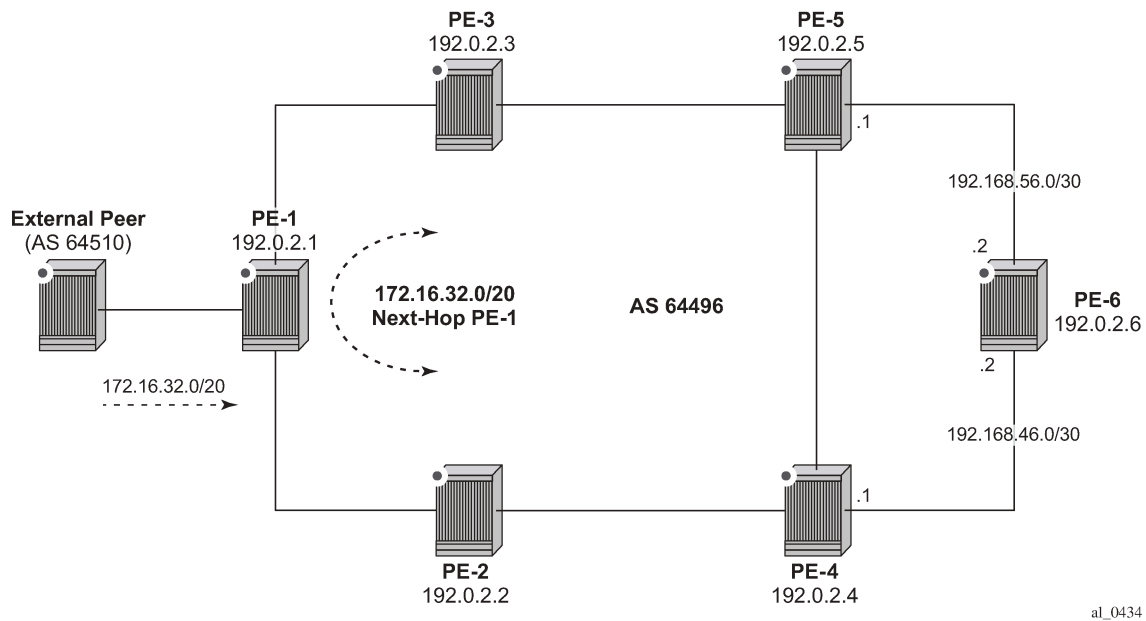
Service and Application Verification

With the RSVP-TE LSP mesh in place, it is now possible to create services and applications to utilize those LSPs. These applications and services include Layer 2 and Layer 3 VPNs, resolution of BGP labeled routes and resolution of BGP, IGP, and static routes. However, the automatically created LSPs are not available for explicit binding in a statically provisioned Service Distribution Point (SDP).

IGP Shortcuts

Figure 181: IGP Shortcuts with RSVP-TE Auto-Mesh demonstrates the use of IGP shortcuts. Prefix 172.16.32.0/20 is advertised to PE-1 from an external peer in AS 64510, which PE-1 subsequently advertises into IBGP, imposing Next-Hop-Self in the process. For more details on IGP shortcuts, see the IGP Shortcuts chapter.

Figure 181: IGP Shortcuts with RSVP-TE Auto-Mesh



al_0434

The objective is for PE-6 to use the automatically created LSP to PE-1 as an IGP shortcut (typically implemented in order to maintain a "BGP-free" core). IGP shortcuts for BGP are enabled under the main **bgp** context using the command **next-hop-resolution shortcut-tunnel** with options for **rsvp**, **ldp**, or **bgp**. Because the example topology only has (automatically created) RSVP-TE LSPs, this option is selected. Besides **rsvp**, **ldp**, and **bgp**, there are other options, but these are beyond the scope of this chapter.

```
[/]
A:admin@PE-6# configure {
  router "Base" {
    bgp {
      next-hop-resolution {
        shortcut-tunnel {
          family ipv4 {
            resolution-filter ?

resolution-filter

bgp          - Use BGP tunneling for next-hop resolution
ldp          - Use LDP tunneling for next-hop resolution
mpls-fwd-policy - Use MPLS forwarding policy for next-hop resolution
rib-api      - Use RIB API gRPC service for next-hop resolution
rsvp        - Use RSVP tunneling for next-hop resolution
sr-isis     - Use IS-IS SR tunneling for next-hop resolution
sr-ospf     - Use OSPF SR tunneling for next-hop resolution
sr-ospf3    - Use OSPFv3 SR tunneling for next-hop resolution
sr-policy   - Use SR policies for next-hop resolution
sr-te       - Use SR-TE
```

On PE-6, the following BGP next-hop resolution configuration defines that IGP shortcuts are resolved using RSVP:

```
[/]
A:admin@PE-6# configure {
  router "Base" {
    bgp {
      next-hop-resolution {
        shortcut-tunnel {
          family ipv4 {
            resolution filter
            resolution-filter {
              rsvp true
            }
          }
        }
      }
    }
  }
}
```

When the shortcuts are enabled, the route-table (and FIB) can be validated to ensure that the programmed next hop is the advertising BGP speaker (as opposed to the IGP next hop), and that traffic is tunneled to that next hop through an RSVP LSP. In this case, the RSVP LSP is the LSP with tunnel ID 61441, which is the LSP to PE-1.

```
[/]
A:admin@PE-6# show router route-table 172.16.32.0/20
```

```
=====
Route Table (Router: Base)
=====
```


Before a VPN-IPv4 prefix is considered valid, the receiving SR OS PE router must be able to resolve the BGP next hop to an LSP in the tunnel table (if not, the prefix is held in Routing Information Base RIB-IN and flagged as invalid). On PE-6, it is possible to verify that the VPN-IPv4 prefix 172.16.1.0/24 received from PE-1 is correctly resolved by looking at the VPRN-specific route table. In the following output, the VPN-IPv4 prefix 172.16.1.0/24 with a next hop of PE-1 (192.0.2.1) is correctly resolved to an RSVP LSP with a tunnel ID of 61441.

```
[/]
A:admin@PE-6# show router 1 route-table 172.16.1.0/24

=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]                               Type   Proto   Age      Pref
  Next Hop[Interface Name]                       Metric
-----
172.16.1.0/24                                     Remote BGP VPN 00h01m11s 170
  192.0.2.1 (tunneled:RSVP:61441)                 30
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

Layer 2 VPN

As previously described, automatically created RSVP LSPs cannot be referenced by statically provisioned SDPs. Without the ability for SDPs to explicitly reference automatically created RSVP LSPs, there is little value in manually defining SDPs within Layer 2 service constructs (there is little point in referring to an SDP that cannot bind to the underlying RSVP mesh). So, in order to deliver Layer 2 services, there is a requirement to adopt a model within the service construct that permits automatic creation of SDP bindings, and this is achieved using a pseudowire template dictating the characteristics of the SDP. The secondary effect of using pseudowire templates to dynamically create SDPs is that these automatically created SDPs can currently only use LDP or BGP as a transport tunnel, not RSVP. The solution is to enable LDP over RSVP.

This can be implemented using static provisioning of peers as shown in the next output, or it can be done using automatic creation of T-LDP sessions. Regardless of the method, a reciprocal configuration must exist at both peer endpoints. The static per-peer configuration is applied in the **targeted-session** context specifying the remote peer system IP address, and the keyword tunneling, which enables tunneling of LDP FECs over RSVP LSPs with a far-end address matching that of the T-LDP peer. At a global level, the **prefer-tunnel-in-tunnel false** command is shown, but is only required when a next hop router advertises a FEC over link-level LDP and T-LDP. In this case, by default, the system would prefer the link-level LDP tunnel, so the **prefer-tunnel-in-tunnel false** instructs the system to prefer an LDP over RSVP tunnel if it is available. Although link-layer LDP is not present in the example topology, the command is included because the presence of link-layer LDP is common.

```
[/]
A:admin@PE-6# configure {
  router "Base" {
    ldp {
      prefer-tunnel-in-tunnel false
      targeted-session {
```

```

        peer 192.0.2.1 {
            tunneling
        }
    }
}

```

The following output provides an example demonstrating the automatic creation of T-LDP sessions. No explicit reference is made to specific peers, but rather a **peer-template** is configured containing the parameters which apply to all T-LDP sessions spawned by this template. In this example, only the **tunneling** command is required. A **peer-template-map** is then used to create a mapping between the peer-template (TLDP-Mesh-template) and a **policy** defining the IP addresses of remote nodes to which T-LDP sessions should be established. In this example, the policy "Remote-PEs-policy" is the same policy previously used by the auto-created RSVP LSP mesh.

```

[/]
A:admin@PE-6# configure {
    router "Base" {
        ldp {
            prefer-tunnel-in-tunnel false
            targeted-session {
                peer-template "TLDP-Mesh-template" {
                    admin-state enable
                    tunneling true
                }
                peer-template-map "TLDP-Mesh-template" {
                    policy-map ["Remote-PEs-policy"]
                }
            }
        }
    }
}

```

Regardless of whether T-LDP sessions are explicitly provisioned, or dynamically created using a peer-template, the result is that a targeted LDP session is established which can be used for advertising address and service FECs, and which is capable of tunneling LDP over RSVP.

```

[/]
A:admin@PE-6# show router ldp targ-peer 192.0.2.1 detail
=====
LDP IPv4 Targeted Peers
=====
-----
192.0.2.1
-----
Admin State      : Up                Oper State       : Up
Last Oper Chg   : 0d 00:01:11
Hold Time       : 45                Hello Factor     : 3
Oper Hold Time  : 45
Hello Reduction : Disabled          Hello Reduction Fctr : 3
Keepalive Timeout : 40             Keepalive Factor  : 4
Active Adjacencies : 1             Last Modified    : Never
Auto Created    : Yes
Creator         : template          Template Name     : TLDP-Mesh-template
Tunneling       : Enabled
Lsp Name        : None
Mcast-Tunneling : Disabled
Lsp Name        : None
Local LSR       : None             32-BitLocalLsr   : Disabled

```



```

Local-LSR ID adv.  : Disabled
Community         :
BFD Status        : Disabled
=====
No. of IPv4 Targeted Peers: 1
=====

```

To create VPLS services using dynamically-created SDPs, BGP Auto-Discovery (BGP-AD) must be used together with LDP (or BGP) pseudowire signaling, for more details see the LDP VPLS Using BGP-Auto Discovery chapter.

In the following output, PE-6 uses BGP-AD and LDP signaling. The same configuration is applied on PE-1. The `vpls-id` is configured in the `bgp-ad` context. The `vpls-id` is a network-wide identifier assigned to all VPLS Switch Instances (VSIs) belonging to the same VPLS, and is carried in VPLS Network Layer Reachability Information (NLRI) as an extended community attribute. A second parameter used for BGP-AD and carried in the VPLS NLRI is the VSI-ID, which uniquely identifies each VSI. The VSI-ID is automatically derived from the global ASN, the VPLS service ID, and the system IP address. To automatically create SDPs, the `bgp` context of the VPLS service refers to a `pw-template` defining the parameters of the pseudowire. In this example, the use of the hash (entropy) label is enabled in the pseudowire template, and a `split-horizon-group`, SHG, is applied.

```

[/]
A:admin@PE-6# configure {
  service {
    pw-template "PW2-template" {
      pw-template-id 2
      hash-label {
      }
      split-horizon-group {
        name "SHG"
      }
    }
    vpls "VPLS-2" {
      admin-state enable
      service-id 2
      customer "1"
      bgp 1 {
        pw-template-binding "PW2-template" {
        }
      }
      bgp-ad {
        admin-state enable
        vpls-id "64496:2"
      }
      sap 1/1/4:2 {
      }
    }
  }
}

```

The following output shows the BGP operational parameters for vpls 2:

```

[/]
A:admin@PE-6# show service id 2 bgp
=====
BGP Information
=====
Bgp Instance      : 1
Vsi-Import        : None

```

```
Vsi-Export      : None
Route Dist     : None
Oper Route Dist : 64496:2
Oper RD Type   : derivedVpls
Rte-Target Import : None
Oper RT Imp Origin : derivedVpls
Oper RT Exp Origin : derivedVpls
Rte-Target Export: None
Oper RT Import  : 64496:2
Oper RT Export  : 64496:2

PW-Template Id : 2
Oper Group     : None
Mon Oper Group : None
BFD Template   : None
BFD-Enabled    : no
Import Rte-Tgt : None
PW-Template SHG : None
BFD-Encap      : ipv4
```

The BGP-AD operational parameters for VPLS 2 are as follows:

```
[/]
A:admin@PE-6# show service id 2 bgp-ad
```

BGP Auto-discovery Information

```
Admin State      : Up
Vpls Id         : 64496:2
Prefix          : 192.0.2.6
-----
```

The following shows that an SDP with ID 32767:4294967295 of type BgpAd has been automatically created. Both the SDP and the SAP are operationally up.

```
[/]
A:admin@PE-6# show service id 2 base | match "Service Access" post-lines 10
Service Access & Destination Points
```

```
-----
Identifier                Type           AdmMTU  OprMTU  Adm  Opr
-----
sap:1/1/4:2               q-tag        1518    1518    Up   Up
sdp:32767:4294967295 SB(192.0.2.1) BgpAd     0      1548    Up   Up
=====
```

* indicates that the corresponding row element may have been truncated.

To create Epipe services using dynamically created SDPs, two options exist. Either LDP FEC 129 signaling can be used, which in turn dictates the presence of pseudowire routing information, or BGP-VPWS based signaling can be used, for more details, see the "BGP Virtual Private Wire Services" chapter in the Layer 2 Services and EVPN volume of *7450 ESS, 7750 SR, and 7950 XRS Advanced Configuration Guide — Book II*. This example illustrates the use of BGP VPWS, but in either case, only single-segment pseudowires are supported. The following output shows the configuration requirements for a basic BGP-based Epipe service on PE-6. Once again, a **pw-template** is used to define the characteristics of the pseudowire, and this template is referenced in the **bgp** context of the Epipe service. The **bgp** context is also where the **route-distinguisher** and **route-target** values are configured, which are carried in the VPWS NLRI and extended communities respectively. The **ve-name**, **ve-id**, and **remote-ve-name** are all configured in the **bgp-vpws** context. The **ve-id** is carried in the VPWS NLRI, and when a PE router receives a VPWS NLRI to try to establish an Epipe service, the **ve-id** from the NLRI is validated against the **ve-id** configured in the **remote-ve-name**. These must match before the Epipe becomes operational.

```
[/]
```

```
A:admin@PE-6# configure {
  service {
    pw-template "PW3-template" {
      pw-template-id 3
      hash-label {
      }
    }
    epipe "Epipe-3" {
      admin-state enable
      service-id 3
      customer "1"
      bgp 1 {
        route-distinguisher "64496:3"
        route-target {
          export "target:64496:3"
          import "target:64496:3"
        }
        pw-template-binding "PW3-template" {
        }
      }
    }
    bgp-vpws {
      admin-state enable
      local-ve {
        name "PE-6"
        id 6
      }
      remote-ve "PE-1" {
        id 1
      }
    }
    sap 1/1/4:3 {
    }
  }
}
```

The basic service information is truncated to show only the relevant information in order to verify that the service is operational. SDP (32766:4294967294) has been automatically created and is of type **BgpVpws**. Both the SDP and the SAP are operationally up.

```
[/]
A:admin@PE-6# show service id 3 base | match "Service Access" post-lines 10
Service Access & Destination Points
-----
Identifier                                     Type           AdmMTU  OprMTU  Adm  Opr
-----
sap:1/1/4:3                                  q-tag          1518    1518    Up   Up
sdp:32766:4294967294 SB(192.0.2.1)      BgpVpws        0        1548    Up   Up
=====
```

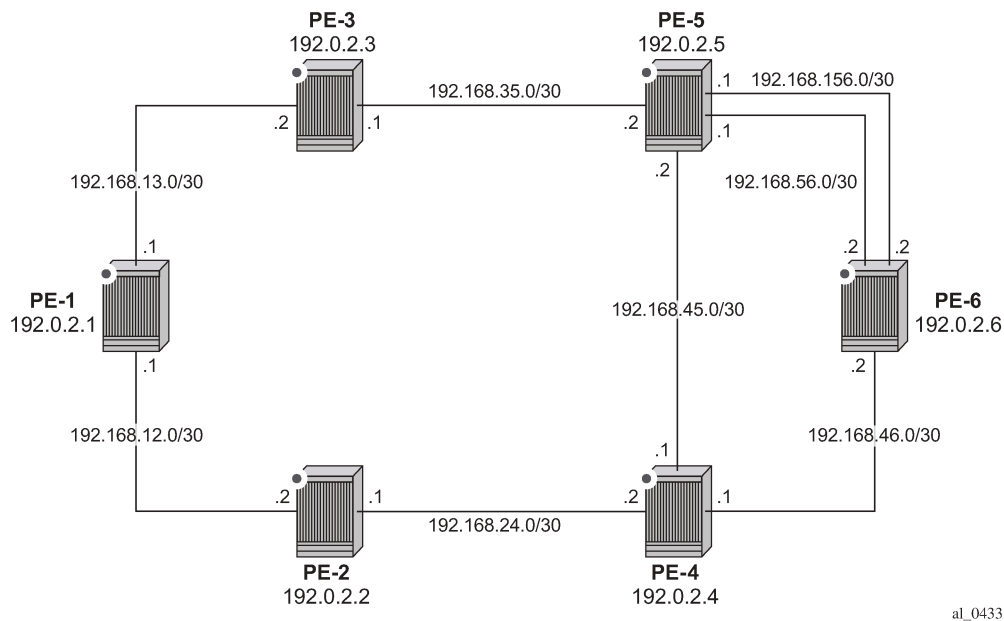
Automatic Creation of RSVP Single-Hop LSPs

As previously discussed, the purpose of a single-hop LSP mesh is to allow for ECMP load balancing of traffic using LDP over RSVP. ECMP load balancing could be implemented using LDP over a partial or full mesh of RSVP-TE LSPs, but the use of single-hop LSPs additionally allows for load balancing across a number of parallel RSVP LSPs between nodes. To illustrate ECMP load balancing over multiple parallel RSVP LSPs, the example topology of [Figure 180: Example Topology](#) is modified to include a parallel link

between PE-6 and PE-5 as shown in [Figure 182: Example Topology for Single-Hop LDP over RSVP with ECMP](#). In addition, all routers are enabled for ECMP=2, as follows.

```
[/]
configure {
  router "Base" {
    ecmp 2
  }
}
```

Figure 182: Example Topology for Single-Hop LDP over RSVP with ECMP



Unlike the automatically created RSVP-TE LSP mesh previously described, the automatically created single-hop RSVP-TE LSPs have no requirement for a prefix-list to be referenced containing the prefixes of the remote nodes that form part of the mesh. In the case of automatically created single-hop LSPs, the TE database keeps track of each TE link which comes up to a directly connected IGP neighbor. The system then establishes a single-hop LSP with a destination address matching the router ID of the neighbor and with a strict hop consisting of the address of the interface used by the TE link.

The first requirement is to create an LSP template containing the common parameters used to establish each single-hop LSP. For a single-hop LSP mesh, the **lsp-template** must be configured with type **p2p-one-hop**. Upon creation of the template, **cspf** is automatically enabled (and cannot be disabled), and the **hop-limit** is set to a value of **2**. The hop limit defines the number of nodes the LSP may traverse, and, because these are single-hop LSPs to adjacent neighbors, a limit of 2 is sufficient. The template must also reference a **default-path** before it can be enabled. The following example references a path named "loose-path" that has no strict or loose hops defined. When the RSVP PATH message is actually generated to create the one-hop LSP, it contains one strict-hop to the interface address of the neighbor; and as destination the system address of the adjacent node.

The next requirement is to trigger the creation of single-hop LSPs, and this is achieved using the **auto-lsp** command. In this example, the LSP template "Single-Hop-template" is referenced, and the command is completed with the keyword **one-hop** to indicate the creation of single-hop LSPs. Unlike an RSVP-

TE mesh, there is no requirement to reference a route policy. In the example, the auto-lsp "Full-Mesh-template" and LSP template "Full-Mesh-template" are removed on all PEs.

```
[/]
A:admin@a PE># configure {
  router "Base" {
    mpls {
      delete auto-lsp "Full-Mesh-template"
      delete lsp-template "Full-Mesh-template"
    }
  }
}
```

The following one-hop LSP template is created on all nodes:

```
[/]
A:admin@a PE># configure {
  router "Base" {
    mpls {
      admin-state enable
      path "loose-path" {
        admin-state enable
      }
      lsp-template "Single-Hop-template" {
        admin-state enable
        type p2p-rsvp-one-hop
        default-path "loose-path"
        hop-limit 2
        path-computation-method local-cspf
      }
      auto-lsp "Single-Hop-template" {
        one-hop true
      }
    }
  }
}
```

When the **auto-lsp** command is entered, the system starts the process of establishing the single-hop LSPs. A check is made of the TE database for every TE link to a directly connected IGP neighbor, and a single-hop LSP is established across each TE link. The following output is taken from PE-6 and shows the automatically created single-hop LSPs. The LSP names are automatically constructed as TemplateName-DestIPv4Address-TunnelId. The LSP name signaled in the session attribute object concatenates the LSP name with the path name (for example Single-Hop-template-192.0.2.4-61449::loose-path). Recall from [Figure 182: Example Topology for Single-Hop LDP over RSVP with ECMP](#) that PE-6 has a single TE-enabled link to PE-4, and two TE-enabled links to PE-5, so with ECMP=2, there is one LSP to PE-4 (192.0.2.4) and two LSPs to PE-5 (192.0.2.5). However, if ECMP=1, only one single-hop LSP would be signaled to PE-5.

```
[/]
A:admin@PE-6# show router mpls lsp

=====
MPLS LSPs (Originating)
=====
```

LSP Name To	Tun Id	Fastfail Config	Adm	Opr
Single-Hop-template-192.0.2.4-61448 192.0.2.4	61448	No	Up	Up
Single-Hop-template-192.0.2.5-61449	61449	No	Up	Up

```

192.0.2.5
Single-Hop-template-192.0.2.5-61450          61450  No      Up    Up
192.0.2.5
-----
LSPs : 3
=====

```

The purpose of single-hop LSPs is to enable ECMP load balancing using LDP over RSVP, so there is a requirement to configure T-LDP sessions between RSVP LSP endpoints. This can be implemented using static peer provisioning, or it can be done using automatic creation of T-LDP sessions, both of which have been previously described and they are therefore not repeated. In this example, the automatic creation of T-LDP sessions approach is used, and T-LDP sessions are created to adjacent neighbors that are capable of tunneling inside RSVP.

```

[/]
A:admin@PE-6# show router ldp session family ipv4

=====
LDP IPv4 Sessions
=====
Peer LDP Id          Adj Type  State           Msg Sent  Msg Recv  Up Time
-----
192.0.2.4:0         Targeted  Established     42        43        0d 00:02:55
192.0.2.5:0         Targeted  Established     43        45        0d 00:03:04
-----
No. of IPv4 Sessions: 2
=====

```

To validate the ECMP load balancing capability, PE-5 is configured to advertise prefix 172.16.5.0/24 to PE-6. In turn, PE-6 is configured for **ibgp-multipath** to enable load balancing over IGP links to the BGP next hop address, **next-hop-resolution shortcut-tunnel resolution-filter ldp** to enable tunneling of traffic destined toward the BGP next hop in MPLS, and **ecmp 2**. For more information about BGP multipath, see the [BGP Multipath](#) chapter.

```

[/]
A:admin@PE-6# configure {
  router "Base" {
    bgp {
      ibgp-multipath true
      next-hop-resolution {
        shortcut-tunnel {
          family ipv4 {
            resolution filter
            resolution-filter {
              ldp true
              delete rsvp
            }
          }
        }
      }
    }
  }
}

```

The prefix 172.16.5.0/24 advertised by PE-5 is learned on PE-6 and installed in the RIB/FIB with PE-5's system address (192.0.2.5) as next hop.

```

[/]
A:admin@PE-6# show router bgp routes 172.16.5.0/24

```

```

=====
BGP Router ID:192.0.2.6      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
  Nexthop (Router)                         Path-Id   IGP Cost
  As-Path                                   -         Label
-----
u*>i 172.16.5.0/24                          100      None
      192.0.2.5                             None     10
      No As-Path                             -
-----
Routes : 1
=====

```

Checking the FIB for the next hop address 192.0.2.5, it can be verified that both links are installed as next hop addresses, meaning that ECMP load balancing is active.

```

[/]
A:admin@PE-6# show router fib 1 ip-prefix-prefix-length 192.0.2.5/32
=====
FIB Display
=====
Prefix [Flags]                               Protocol
  NextHop
-----
192.0.2.5/32                                 ISIS
  192.168.56.1 (int-PE-6-PE-5)
  192.168.156.1 (int-PE-6-PE-5-2nd)
-----
Total Entries : 1
=====

```

Conclusion

Automatic creation of RSVP-TE LSPs provides a good solution for reducing the amount of provisioning activity required when configuring RSVP LSPs. However, there are some constraints with regard to the way that services are deployed on top of those LSPs. SDPs cannot explicitly reference automatically-created RSVP LSPs, which means that automatically created SDPs need to be used for Layer 2 services. In turn, automatically-created SDPs can only use LDP or BGP as a transport tunnel (not RSVP), so, to use the automatically created RSVP mesh, LDP over RSVP must be used. These restrictions need to be fully understood before considering deployment of automatically created RSVP-TE LSPs.

BFD for RSVP-TE and LDP LSPs

This chapter provides information about BFD for RSVP-TE LSPs and LDP LSPs.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written based on SR OS Release 15.0.R7, but the MD-CLI in the current edition corresponds to SR OS Release 23.3.R3.

Overview

SR OS supports RFC 5884 and enables LSPs to be monitored between the ingress and egress LERs, regardless of the number of LSRs that the LSP traverses. For continuity checks in MPLS LSPs, BFD packets are transmitted using the MPLS encapsulation, so they share fate with the LSP data path. When enabled, faults to individual LSPs can be detected quickly, so BFD for MPLS LSPs is ideal for monitoring LSPs carrying high-value services, where detecting forwarding failures in a minimum amount of time is critical. The LSPs can be established through RSVP-TE or LDP.

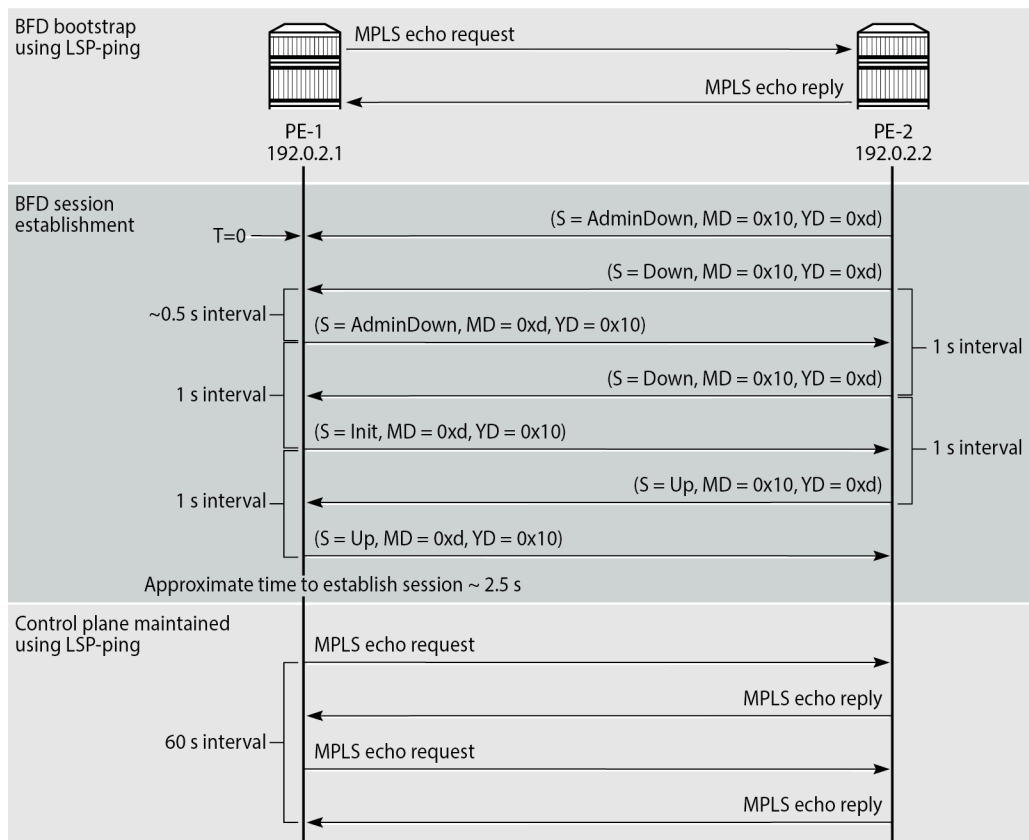
Enabling BFD for LSPs avoids manual hop-by-hop troubleshooting of each element along the LSP. BFD sessions are created and run end-to-end, from ingress to egress, so BFD session state is maintained in the ingress LER and egress LER, but not in intermediate LSRs. If an LSP BFD session changes state, an SNMP trap is generated. Because LSPs are unidirectional, a routed return path is used for the BFD control packets from the egress LER toward the ingress LER.

BFD is only used for fault detection, and will not redirect traffic to an alternate path. On detection of a failure, BFD informs other software components, which then can redirect traffic to avoid faulty links.

BFD is bootstrapped using an LSP ping. An MPLS echo request packet is transmitted along the LSP path, including a BFD discriminator TLV containing the head-end BFD discriminator value. The tail end responds with an echo reply packet, using the IP forwarding path, including the tail-end BFD discriminator value.

Afterward, BFD control packets establish a BFD session between the head end and tail end using the discriminator values from the bootstrap session. The egress LER will send a BFD control packet upon receipt. Every 60 s, the head end transmits an LSP ping for control plane verification. The minimum value for the LSP BFD control transmit interval is 100 ms. [RSVP-TE LSP BFD session establishment: BFD handshake](#) shows the MPLS LSP BFD session establishment with a BFD control transmit interval of 1 second.

Figure 183: MPLS LSP BFD session establishment: BFD handshake



Legend:
MD - My Discriminator
YD - Your Discriminator

35627

An MPLS LSP is created from head-end PE-1 to tail-end PE-2 with BFD enabled. A BFD template is created with transmit and receive interval each equal to 1000 ms in this example. This BFD template is applied to the MPLS LSP.

When the MPLS LSP is enabled, an LSP ping is used to bootstrap BFD to LSP with a default ping interval of 60 s. The BFD discriminators (MD = 0x10 and YD = 0xd) are negotiated using LSP ping, so the BFD session establishment starts with a message containing both discriminators. The MPLS echo reply is sent using IP routing instead of using another MPLS LSP.

Each session has its own pair of discriminators, so multiple discriminators are allocated by the system. The two-way handshake to establish a BFD session between network elements takes a few seconds in this example: 2.5 s to establish the BFD session after the BFD is bootstrapped. If the LSP BFD control transmit interval is as low as 100 ms, the handshake takes less than a second.

BFD can be used for RSVP-TE and LDP LSPs. If BFD is applied to RSVP-TE LSPs, it only runs on the currently active path. It cannot determine if any non-active paths (for example, a secondary path or primary path during reversion) that the system might switch to are up and forwarding. If BFD is applied to LDP LSPs, the session runs on the path defined by the underlying IGP.

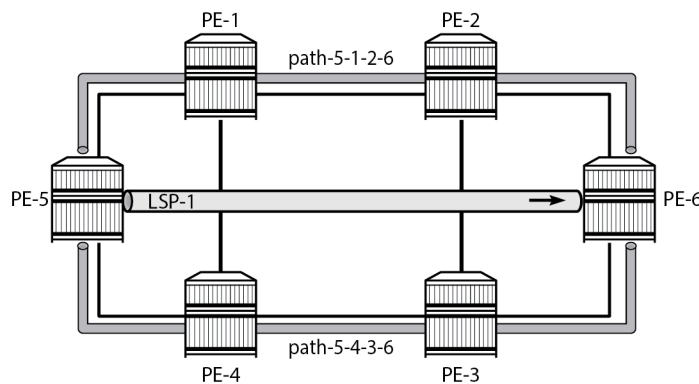
BFD for LSPs can be combined with a failure action. For RSVP-TE LSPs, the failure action can be *down* or *failover*; see the [BFD for RSVP-TE LSPs with Failure Action](#) chapter for more information. For LDP LSPs, the failure action can only be *down*. LSP BFD does not affect the operational state of an LSP, because the operational state is controlled by the control plane. Therefore, the failure action **down** will mark the LSP as unavailable for use by services.

Configuration

BFD for RSVP-TE LSPs

[Figure 184: BFD for RSVP-TE LSPs - topology](#) shows the example topology for BFD for RSVP-TE LSPs.

Figure 184: BFD for RSVP-TE LSPs - topology



27613

The initial configuration includes:

- Cards, MDAs, and ports
- Router interfaces
- IS-IS as IGP on all interfaces (alternatively, OSPF can be used), with traffic engineering enabled
- MPLS and RSVP enabled on all interfaces

Base configuration

The example topology from [Figure 184: BFD for RSVP-TE LSPs - topology](#) has an LSP defined on PE-5 using two strict paths, where *path-5-1-2-6* is taking the upper path and used as the primary path, and *path-5-4-3-6* is the lower path and used as the secondary, as follows:

```
# on PE-5:
configure {
  router "Base" {
    mpls {
      path "path-5-1-2-6" {
        admin-state enable
        hop 10 {
          ip-address 192.168.15.1
        }
      }
    }
  }
}
```

```
        type strict
      }
      hop 20 {
        ip-address 192.168.12.2
        type strict
      }
      hop 30 {
        ip-address 192.168.26.2
        type strict
      }
    }
  path "path-5-4-3-6" {
    admin-state enable
    hop 10 {
      ip-address 192.168.45.1
      type strict
    }
    hop 20 {
      ip-address 192.168.34.1
      type strict
    }
    hop 30 {
      ip-address 192.168.36.2
      type strict
    }
  }
}
lsp "lsp-1" {
  admin-state enable
  type p2p-rsvp
  to 192.0.2.6
  path-computation-method local-cspf
  primary "path-5-1-2-6" {
  }
  secondary "path-5-4-3-6" {
  }
}
```

BFD for RSVP-TE LSPs configuration

There are four steps to configure BFD for RSVP-TE LSPs:

1. Configure a BFD template.
2. Enable LSP BFD on the tail node.
3. Apply the BFD template to the LSP or LSP path.
4. Enable BFD on the LSP or LSP path.

Step 1: Configure a BGP template

The BFD template provides the control packet timer values for the BFD session to use at the LSP head end. The general command to define a BFD template is as follows:

```
configure {
  bfd {
    bfd-template <[32 chars max]> {
      transmit-interval <[10..100000] milli-seconds>
      receive-interval <[10..100000] milli-seconds>
      echo-receive <[100..100000] milli-seconds>
    }
  }
}
```

```
multiplier <[1..20]>
type {cpm-np}
}
```



Note:

The minimum transmit and receive interval for MPLS LSPs equals 100 ms. Intervals smaller than 100 ms require BFD type CPM-NP and can be used for SR-TE LSPs, but not for RSVP-TE LSPs.

Network processor BFD is not supported for RSVP-TE LSPs. An error is generated if a user tries to apply a BFD template of **type cpm-np** to an LSP, as follows.

```
# on PE-5:
configure {
  bfd {
    bfd-template "bfd-cpm-np-Tx100" {
      type cpm-np
    }
  }
}
```

```
*[ex:/configure router "Base" mpls lsp "lsp-1" bfd]
A:admin@PE-5# bfd-template "bfd-cpm-np-Tx100"

*[ex:/configure router "Base" mpls lsp "lsp-1" bfd]
A:admin@PE-5# commit
MINOR: MGMT_CORE #4001: configure router "Base" mpls lsp "lsp-1"
bfd bfd-template - bfd-template type is not valid for this application -
configure bfd bfd-template "bfd-cpm-np-Tx100" type
```

BFD templates may be used by different BFD applications (for example, LSPs or pseudowires). If the BFD timer values are changed in a template, the BFD sessions on LSPs or spoke-SDPs to which that template applies try to renegotiate their timers to the new values. An error will be generated when the user attempts to commit changes to a BFD template that is already applied to an LSP where the new values are invalid for LSP BFD.

In this example, the BFD template used is configured as follows:

```
# on PE-5:
configure {
  bfd {
    bfd-template "bfdt-1" {
      multiplier 5
      receive-interval 2000
      transmit-interval 2000
    }
  }
}
```

Step 2: Enable LSP BFD on the tail node

The BFD state machine at the tail end initially uses system-wide parameters, because an LSP is unidirectional so no configuration for the LSP exists at the tail end. The head end then attempts to adjust the control packet timer values when it transitions to the INIT state.

LSP BFD is enabled or disabled on a node-wide basis with the **bfd-sessions <maxlimit>** command in the **configure router lsp-bfd** context. The **maxlimit** parameter configures the maximum number of LSP BFD sessions that can be established. This is required at the tail end of the LSP. The BFD state machine at the tail end initially uses system-wide parameters, such as the transmit and receive intervals, which are both 1000 ms by default, but the minimum value is 100 ms.

In this example, the tail node is configured as follows:

```
# on PE-6:
configure {
  router "Base" {
    lsp-bfd {
      bfd-sessions 10          # must be set to non-zero value
      # tail-end {
        # transmit-interval 1000 # default
        # receive-interval 1000 # default
        # multiplier 3          # default
      # }
    }
  }
}
```

Because BFD resources are shared by different BFD applications, the limit defined here must provide sufficient resources for other applications.

Steps 3 and 4: Apply the BGP template to the LSP or LSP path and enable BFD

LSP BFD is applicable to configured RSVP LSPs as well as to mesh point-to-point and one-hop point-to-point auto-created LSPs. It is configured on an RSVP-TE LSP, or on the path of an RSVP-TE LSP, under the **bfd** context at the LSP head end.

A BFD template must always be configured first. BFD is then enabled using the **bfd-liveness** command.

To apply and enable the BFD template at RSVP-TE LSP level, the command is as follows:

```
configure {
  router "Base" {
    mpls {
      lsp <lsp-name> {
        bfd {
          bfd-liveness <boolean>
          bfd-template <name>
          # lsp-ping-interval 60          # default: 60 s
          # failure-action none          # default: none
          # wait-for-up-timer 4          # default: 4 s
        }
      }
    }
  }
}
```

When BFD is configured at the LSP level, BFD packets follow the currently active path of the LSP.

To apply and enable the BFD template at primary path level, the command is as follows:

```
configure {
  router "Base" {
    mpls {
      lsp <lsp-name> {
        primary <path-name> {
          bfd {
            bfd-liveness true
            bfd-template <name>
          }
        }
      }
    }
  }
}
```

It is possible to configure LSP BFD on a secondary path, but the corresponding BFD session will only be established when the secondary path becomes the active path after failover. It is not possible to configure LSP BFD on point-to-multipoint LSPs.

LSP BFD at the LSP level and the path level are mutually exclusive. That is, if LSP BFD is already configured for the LSP, its configuration for the path is blocked. Likewise, it cannot be configured on the LSP if it is already configured at the path level.

LSP BFD is supported on auto-created LSPs. In that case, LSP BFD is configured on mesh point-to-point and one-hop point-to-point auto-created LSPs using the LSP template.

In this example, on the head-end node, the BFD template is applied to the LSP and BFD is enabled, as follows:

```
# on PE-5:
configure {
  router "Base" {
    mpls {
      lsp "lsp-1" {
        bfd {
          bfd-liveness true
          bfd-template "bfdt-1"
        }
      }
    }
  }
}
```

BFD verification

The details of the MPLS LSP show that BFD is enabled and using BFD template *bfdt-1*, as follows:

```
[/]
A:admin@PE-5# show router mpls lsp detail

=====
MPLS LSPs (Originating) (Detail)
=====
Legend :
+ - Inherited
=====
-----
Type : Originating
-----
LSP Name      : lsp-1
LSP Type      : RegularLsp          LSP Tunnel ID      : 1
LSP Index     : 1                  TTM Tunnel Id     : 1
From          : 192.0.2.5
To           : 192.0.2.6
Adm State     : Up                  Oper State         : Up
LSP Up Time   : 0d 00:26:45         LSP Down Time     : 0d 00:00:00
Transitions   : 1                  Path Changes      : 1
Retry Limit   : 0                  Retry Timer       : 30 sec
Signaling     : RSVP              Resv. Style       : SE
Hop Limit     : 255               Negotiated MTU    : 8982
Adaptive      : Enabled            ClassType         : 0
FastReroute   : Disabled          Oper FR           : Disabled
PathCompMethod : local-cspf       ADSPEC            : Disabled
FallbkPathComp : not-applicable
Metric        : N/A               Metric Type       : igp
Load Bal Wt   : N/A               ClassForwarding   : Disabled
Include Grps  :                   Exclude Grps      :
None          :                   None
Least Fill    : Disabled          Soft Preemption   : Enabled
BFD Template : bfdt-1          BFD Ping Intvl   : 60
BFD Enable   : True           BFD Failure-action : None
WaitForUpTimer : 4

---snip---

Primary(a)    : path-5-1-2-6
Up Time       : 0d 00:26:45
```

```

Bandwidth      : 0 Mbps
Secondary      : path-5-4-3-6
Down Time      : 0d 00:26:45
Bandwidth      : 0 Mbps
=====

```

Initially, the BFD session is running over path *path-5-1-2-6*, as follows:

```

[/]
A:admin@PE-5# show router bfd session

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id          State      Tx Pkts   Rx Pkts
Rem Addr/Info/SdpId Multipl   Tx Intvl  Rx Intvl
Protocols           Type     LAG Port  LAG ID
Loc Addr            LAG name
-----
lsp-1::path-5-1-2-6      Up        795       732
192.0.2.6             5         2000      2000
rsvpLsp               central    N/A       N/A
192.0.2.5
-----
No. of BFD sessions: 1
=====

```

At the head end, the BFD session details are as follows:

```

[/]
A:admin@PE-5# show router bfd session detail lsp-rsvp head

=====
BFD On LSP Session
=====
Rsvp Session      : lsp-1::path-5-1-2-6
Remote Address    : 192.0.2.6
Lsp Id            : 52224
Oper State        : Up
Recd Msgs         : 786
Up Time           : 0d 00:25:00
Last Down Time    : 0d 00:00:04
Tunnel Id         : 1
Protocols         : rsvpLsp
Sent Msgs         : 855
Up Transitions    : 1
Down Transitions  : 0
Version Mismatch  : 0

Forwarding Information

Local Discr       : 1
Local Diag        : 0 (None)
Local Mode        : Async
Local Min Tx      : 2000
Last Sent         : 07/10/2023 09:04:18
Type              : central
Remote Discr      : 1
Remote Diag       : 0 (None)
Remote Min Tx     : 1000
Remote C-flag     : 1
Last Recv         : 07/10/2023 09:04:18
Local State       : Up
Local Mult        : 5
Local Min Rx      : 2000
Remote State      : Up
Remote Mode       : Async
Remote Mult       : 3
Remote Min Rx     : 1000
=====

```

At the tail end, the BFD session details are as follows:

```
[/]
A:admin@PE-6# show router bfd session detail lsp-rsvp tail

=====
BFD On LSP Session
=====
Rsvp Session      : lsp-1::path-5-1-2-6
Remote Address    : 192.0.2.5
Lsp Id            : 52224                Tunnel Id      : 1
Oper State        : Up                  Protocols      : rsvpLsp
Up Time           : 0d 00:29:55         Up Transitions : 1
Last Down Time    : 0d 00:00:04         Down Transitions : 0
                                           Version Mismatch : 0

Forwarding Information

Local Discr       : 1                   Local State    : Up
Local Diag        : 0 (None)
Local Mode        : Async
Local Min Tx      : 1000                Local Mult     : 3
Last Sent (ms)    : 1                   Local Min Rx   : 1000
Type              : cpm-np
Remote Discr      : 1                   Remote State   : Up
Remote Diag       : 0 (None)           Remote Mode    : Async
Remote Min Tx     : 2000                Remote Mult    : 5
Remote C-flag     : 0
Last Recv (ms)   : 1                   Remote Min Rx  : 2000
=====
```

A failure is emulated by bringing down the link between PE-1 and PE-2. BFD detects the failure in the upper path quickly, which results in the BFD session being re-established on *path-5-4-3-6*, as follows:

```
[/]
A:admin@PE-5# show router bfd session

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id          State      Tx Pkts  Rx Pkts
Rem Addr/Info/SdpId:VcId  Multipl  Tx Intvl Rx Intvl
Protocols           Type      LAG Port  LAG ID
Loc Addr                               LAG name
-----
lsp-1::path-5-4-3-6      Up         6         8
192.0.2.6                5         2000      2000
rsvpLsp                  central    N/A       N/A
192.0.2.5
-----
No. of BFD sessions: 1
=====
```

Bringing up the link between PE-1 and PE-2 will result in the primary path becoming active again.

The ping bootstrap and periodic verification information for BFD on LSPs can be displayed at the head end, as follows:

```
[/]
A:admin@PE-5# show test-oam lsp-bfd lsp-name "lsp-1"

-----
LSP Ping Bootstrap and Periodic Verification Information for BFD on LSPs
-----
OAM Operational State      : Bootstrapped - Sending Periodic Verification
FEC Type                   : RSVP
LSP Name                   : lsp-1
LSP Path Status            : active
Source Address             : 192.0.2.5
Replying Node              : 192.0.2.6
Latest Return Code        : EgressRtr (3)
Latest Return Subcode     : 1
Local BFD Discriminator    : 2           Remote BFD Discriminator : 2
LSP Ping Tx Interval (s)  : 60          Bootstrap Retry Count   : 0
Tx LSP Ping Requests      : 1           Rx LSP Ping Replies     : 1
-----
No. of matching BFD on LSP sessions: 1
-----
```

BFD sessions changing state are trapped, so these are logged in log 99, as follows:

```
120 2023/07/10 09:10:21.071 UTC MINOR: BFD #2002 Base 192.0.2.6
"The lspHead BFD session with Local Discriminator 2 on 192.0.2.6 is up (Tunnel Id 1, Path LSP
ID 52226)"

121 2023/07/10 09:11:23.500 UTC WARNING: MPLS #2011 Base VR 1:
"LSP path lsp-1::path-5-1-2-6 is operationally enabled ('no shutdown')"
```

```
122 2023/07/10 09:11:24.494 UTC MINOR: BFD #2004 Base 192.0.2.6
"The protocol (RSVP LSP) using BFD session on node 192.0.2.6 has been added"
```

```
123 2023/07/10 09:11:25.494 UTC MINOR: MPLS #2027 Base VR 1:
"LSP lsp-1 active path lsp-1::path-5-4-3-6 has changed to active path lsp-1::path-5-1-2-6"
```

```
124 2023/07/10 09:11:25.494 UTC MINOR: BFD #2004 Base 192.0.2.6
"The protocol (RSVP LSP) using BFD session on node 192.0.2.6 has been cleared"
```

```
125 2023/07/10 09:11:25.494 UTC MINOR: BFD #2003 Base 192.0.2.6
"The lspHead BFD Session with Local Discriminator 2 on 192.0.2.6 has been deleted"
```

```
126 2023/07/10 09:11:25.495 UTC MAJOR: SVCMGR #2316 Base
"Processing of a SDP state change event is finished and the status of all affected SDP Bindings
on SDP 56 has been updated."
```

```
127 2023/07/10 09:11:28.494 UTC WARNING: MPLS #2012 Base VR 1:
"LSP path lsp-1::path-5-4-3-6 is operationally disabled ('shutdown') because nonActive
Secondary"
```

```
128 2023/07/10 09:11:29.231 UTC MINOR: BFD #2002 Base 192.0.2.6
"The lspHead BFD session with Local Discriminator 3 on 192.0.2.6 is up (Tunnel Id 1, Path LSP
ID 52228)"
```

The **tools** command for displaying LSP details at the head end also includes BFD related information, if applicable, as follows:

```
[/]
A:admin@PE-5# tools dump router mpls lspinfo "lsp-1" detail
```

```
LSP "lsp-1" LspIdx 1 LspType Dynamic State LSPS_UP Flags 0x2000
AdminState Up OperState Up RowStatus Active
From N/A To 192.0.2.6
NumPaths 2 NumSdps 1 NumCBFSdps 0 NumFltrEntries 0
ActivePath lsp-1::path-5-1-2-6(LspId 52228)
HoldTimeRemaining 0secs ClassType 0 SoftPreemption TRUE Metric 0 OperMetric 30
LDPoSvp Include VprnAutoBind Include IgpShortCut Include BgpShortCut Include
BgpTransTunnel Include IpShCutTtlPropLocal TRUE IpShCutTtlPropTans TRUE
RelativeMetricOffset 2147483647 MTU 8982 InUseByLdp FALSE TTMPref 7
EntropyLabel inherit OperEntropyLabel enable NegEntropyLabel disable
ClassForwarding: Disabled
BFD Enabled Template bfdt-1 PingInterval 60 FailureAction None WaitForUp 4sec
PCE Report: Disabled PCE Control: Disabled
Path Profile:
None
Admin Tags:
None
Lsp-self-ping: Config: inherit, Oper: Disabled, TimedOutCnt: 0 OamNoRsc: 0
Path "path-5-1-2-6" LspId 52228 LspPathIndex 2 PathType Primary ActivePath Yes
RowStatus Active LastChange 000 00:51:09.090
AdminState Up OperState Up OperStateChange 000 00:14:35.230
TE Computed Hop List:
Hop[1] IngIp 192.0.2.5 IngLnkId 0 EgrIp 192.168.15.2 EgrLnkId 0 RtrId 192.0.2.5 Flag 0x0
Hop[2] IngIp 192.168.15.1 IngLnkId 0 EgrIp 192.168.12.1 EgrLnkId 0 RtrId 192.0.2.1 Flag
0x0
Hop[3] IngIp 192.168.12.2 IngLnkId 0 EgrIp 192.168.26.1 EgrLnkId 0 RtrId 192.0.2.2 Flag
0x0
Hop[4] IngIp 192.168.26.2 IngLnkId 0 EgrIp 192.0.2.6 EgrLnkId 0 RtrId 192.0.2.6 Flag 0x0
LspPath FsmState LSP_PATH_S_UP Flags 0x0 miscFlags 0x2
RetryAttempts 0 RetryInterval 30 NextRetryIn 0secs
FailNode 0.0.0.0 FailCode noError
Class Type 0 SetupPri 7 HoldPri 0 Pref 0 HopLimit 255 BW 0Mbps
TotIgpCost 30 OperMetric 30 MTU 8982
BFD Disabled Template n/a PingInterval 60 WaitForUp 4sec
Degraded No
Oper Values:
Class Type 0 SetupPri 7 HoldPri 0 HopLimit 255 BW 0Mbps
RecordRoute RecordLabel No Adspec
No PropagateAdminGroup Exclude 0x00000000 Include 0x00000000
No FRR
Metric 30 CSPF No Least Fill Intra-area
NegotiatedEntropyLabel Disabled
PCE-Computed No PCE-Reported No PCE-Controlled No
BFD State Up InitTime 0d 00:14:34 UpTime 0d 00:14:30
OldMBBPathsCleanedUp Yes
Path "path-5-4-3-6" LspId 52226 LspPathIndex 3 PathType Secondary ActivePath No
RowStatus Active LastChange 000 00:51:09.090
AdminState Up OperState Down OperStateChange 000 00:14:30.230
LspPath FsmState LSP_PATH_S_DOWN Flags 0x40000 miscFlags 0x2
RetryAttempts 0 RetryInterval 30 NextRetryIn 0secs
FailNode 192.0.2.5 FailCode nonActiveSecondary
Class Type 0 SetupPri 7 HoldPri 0 Pref 255 HopLimit 255 BW 0Mbps
TotIgpCost 0 OperMetric 16777215 MTU 0
SRLG Disabled SRLGDisjoint No
BFD Disabled Template n/a PingInterval 60 WaitForUp 4sec
Degraded No
OldMBBPathsCleanedUp Yes

Total Ingress LSP Count : 1
```

The current BFD session information for RSVP LSPs can be displayed using the following **tools** command at the head end:

```
[/]
A:admin@PE-5# tools dump router bfd lsp-rsvp head
-----
FEC: (PTR 0x10eb60f50)
RSVP : vrId: 1 (To: 192.0.2.6 - 1 - 192.0.2.5), Sender (192.0.2.5 - 52228)

Session: lsp-1::path-5-1-2-6    refCnt = 1
PingIntvl: 60 Flags: 0x6 ProtNhidx: 13 NumNextHop: 1
TempLName: bfdt-1 LspName: lsp-1 TunnelId: 1 NumLspUser: 0
NextHop: 192.168.15.1 IfIndex: 1 Flags: 0x0 isBackup: N
PGId: 0 [State: N/A] NhIdx: 13
Label:- [0]524286
      BFD Handle: 3 State: UP LastEvent: UP
      BFD UserId: 24 TmrActive: N [0] NumRetry: 0
      DstAddr: 127.0.0.3 LocalDiscr: 3 RemoteDiscr: 0
-----
Total FEC Count in Head: 1
```

Other **tools** commands can display BFD LSP information at the tail end, as follows:

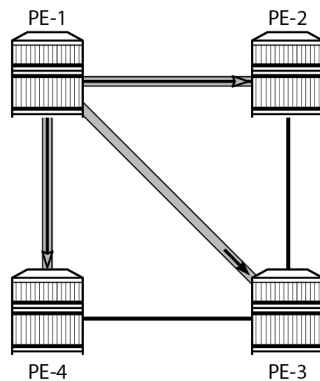
```
[/]
A:admin@PE-6# tools dump test-oam lsp-bfd tail
-----
Total Number of Active Tail Cache Sessions : 1
-----

VrId           : 1
RemoteBfdDisc  : 3
LocalBfdDisc   : 3
FecType        : rsvp_ipv4(3)
LspId          : 52228
TunnelId       : 1
SenderIp       : 192.0.2.5
TunnEndIp      : 192.0.2.6
ExtTunnId      : 192.0.2.5
Bootstrap Echo Rx : rcvd 2023/07/10 09:11:24.00 UTC
                  handle 3 seqNum 2 rc 3 rsc 1
Last Echo Req Rx  : rcvd 2023/07/10 09:28:30.00 UTC
                  handle 3 seqNum 19 rc 3 rsc 1
-----
Number of Matched Tail Cache Sessions : 1
-----
```

BFD for LDP LSPs

[Figure 185: BFD for LDP LSPs - topology](#) shows the example topology for BFD for LDP LSPs.

Figure 185: BFD for LDP LSPs - topology



27614

The initial configuration includes:

- Cards, MDAs, and ports
- Router interfaces
- IS-IS as IGP on all interfaces (alternatively, OSPF can be used)

Base configuration

The example topology from [Figure 185: BFD for LDP LSPs - topology](#) has LDP configured on all interfaces. LDP automatically generates and distributes labels across the network, so for the topology in [Figure 185: BFD for LDP LSPs - topology](#), LDP tunnels are created so that every node can reach any other node; only the tunnels originating in PE-1 are shown. The LDP configuration for PE-1 is as follows; the LDP configuration for PE-2, PE-3, and PE-4 is similar.

```
# on PE-1:
configure {
  router "Base" {
    ldp {
      interface-parameters {
        interface "int-PE-1-PE-2" {
          ipv4 {
          }
        }
        interface "int-PE-1-PE-4" {
          ipv4 {
          }
        }
      }
    }
  }
}
```

BFD for LDP LSPs configuration

There are six steps to configure BFD for LDP LSPs:

1. Create a BFD template.
2. Enable LSP BFD on the tail node.

3. Create a prefix list.
4. Configure LSP BFD for LDP.
5. Apply the BFD template to the LDP LSP.
6. Enable BFD on the LDP LSP.

Step 1: Create a BFD template

The command to define a BFD template is the same as for RSVP-TE LSPs. The BFD template used for the LDP LSPs is configured as follows:

```
# on PE-1:
configure
  bfd {
    bfd-template "bfdt-2" {
      receive-interval 1000
      transmit-interval 1000
    }
  }
```

Step 2: Enable LSP BFD on the tail node

The command to enable or disable LSP BFD on a node-wide basis at the tail end of the tunnels is the same as for RSVP-TE LSPs. In this example, the tail nodes PE-2 and PE-3 are configured as follows:

```
# on PE-2 and PE-3:
configure {
  router "Base" {
    lsp-bfd {
      bfd-sessions 5                # must be set to non-zero value
      # tail-end {
        # transmit-interval 1000    # default
        # receive-interval 1000    # default
        # multiplier 3             # default
      # }
    }
  }
}
```

Step 3: Create a prefix list

When high-value services are relying on the LDP tunnels between PE-1, PE-2, and PE-3, a prefix list with the system IP addresses (or other routable loopback addresses) of PE-2 and PE-3 can be used in PE-1 for monitoring these tunnels. In this example, the *pfx-lst-1* prefix list is defined as follows:

```
# on PE-1
configure {
  policy-options {
    prefix-list "pfx-lst-1" {
      prefix 192.0.2.2/32 type exact {
      }
      prefix 192.0.2.3/32 type exact {
      }
    }
  }
}
```

Steps 4, 5, and 6: Configure LSP BFD for LDP, apply BFD template, enable BFD

LSP BFD is configured for LDP using the following commands:

```
configure {
  router "Base" {
    ldp {
      lsp-bfd <prefix-list-name> {
        priority <priority-level>           # default: 1
        bfd-template <bfd-template-name>
        source-address <ip-address>
        bfd-liveness <boolean>
        lsp-ping-interval <seconds>         # default: 60
        failure-action <down>              # default: none
      }
    }
  }
}
```

The priority level is set to one, by default, and is used in case a prefix appears in multiple prefix lists; see the *7450 ESS*, *7750 SR*, *7950 XRS*, and *VSR MPLS Guide* for more information. The source address can be any local address routable by the other nodes in the network; by default, the system IP address is used. The LSP ping interval defines how frequently ping messages must be sent on the LSP. The only possible failure action for LDP LSPs is *down*, which makes the LSP unavailable for user traffic. By default, no failure action is configured.

In this example, the BFD template is applied to the head-end node in the **lsp-bfd** context, as follows. The **lsp-bfd** command takes the prefix list name defined in [step 3](#) as its argument. BFD template must always be applied first. BFD is then enabled using the **bfd-liveness** command.

```
# on PE-1:
configure {
  router "Base" {
    ldp {
      lsp-bfd "pfx-lst-1" {
        bfd-template "bfdt-2"
        source-address 192.0.2.1
        bfd-liveness true
      }
    }
  }
}
```

BFD verification

The prefix lists applied to LDP BFD are the following:

```
[/]
A:admin@PE-1# show router ldp lsp-bfd

=====
BFD on LDP LSP Configuration Summary
=====
Prio  Prefix List Name                Enabled  Prefixes
-----
1     pfx-lst-1                       Yes     2
-----
No. of prefix lists: 1
=====
```

The LDP BFD information for prefix list *pfx-lst-1* is as follows:

```
[/]
```

```
A:admin@PE-1# show router ldp lsp-bfd "pfx-lst-1"

=====
BFD on LDP LSP Configuration Detail
=====
Prefix List      : pfx-lst-1
Prefix Count    : 2
BFD Template    : bfdt-2
Source Address  : 192.0.2.1
BFD Enable      : Yes           Failure Action   : none
LSP Ping Interval : 60 seconds  Priority       : 1
=====
```

The prefixes of prefix list *pfx-lst-1* to which the system tries to establish BFD sessions are the following:

```
[/]
A:admin@PE-1# show router ldp lsp-bfd "pfx-lst-1" prefixes

=====
BFD on LDP LSP Prefix List "pfx-lst-1" (Enabled)
=====
Prefix                                     Operational State
-----
192.0.2.2/32                               Up
192.0.2.3/32                               Up
-----
No. of prefixes: 2
=====
```

The LDP BFD session data created and maintained at the head end PE-1 is as follows:

```
[/]
A:admin@PE-1# show router bfd session lsp-ldp head

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id          State      Tx Pkts  Rx Pkts
Rem Addr/Info/SdpId:VcId  Multipl  Tx Intvl  Rx Intvl
Protocols           Type     LAG Port   LAG ID
Loc Addr                                     LAG name
-----
192.0.2.2/32        Up         469       427
N/A                 3         1000      1000
ldpLsp              central    N/A       N/A
192.0.2.1
192.0.2.3/32        Up         471       426
N/A                 3         1000      1000
ldpLsp              central    N/A       N/A
192.0.2.1
-----
No. of BFD sessions: 2
=====
```

The following command shows the LDP BFD session information at the tail end PE-3:

```
[/]
A:admin@PE-3# show router bfd session lsp-ldp tail
```

```

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path   pp = Protecting path
=====
BFD Session
=====
Session Id           State      Tx Pkts   Rx Pkts
Rem Addr/Info/SdpId Multipl   Tx Intvl  Rx Intvl
Protocols           Type     LAG Port  LAG ID
Loc Addr
-----
192.0.2.3/32        Up         N/A       N/A
192.0.2.1           3         1000     1000
ldpLsp              cpm-np    N/A       N/A
192.0.2.3
-----
No. of BFD sessions: 1
=====

```

The ping bootstrap and periodic verification information for BFD on LSPs can be displayed at the head end, as follows:

```

[/]
A:admin@PE-1# show test-oam lsp-bfd
-----
LSP Ping Bootstrap and Periodic Verification Information for BFD on LSPs
-----
OAM Operational State : Bootstrapped - Sending Periodic Verification
FEC Type              : LDP
Prefix                : 192.0.2.2/32
Source Address        : 192.0.2.1
Replying Node         : 192.0.2.2
Latest Return Code    : EgressRtr (3)
Latest Return Subcode : 1
Local BFD Discriminator : 1           Remote BFD Discriminator : 1
LSP Ping Tx Interval (s) : 60         Bootstrap Retry Count   : 0
Tx LSP Ping Requests   : 10         Rx LSP Ping Replies    : 10
-----
OAM Operational State : Bootstrapped - Sending Periodic Verification
FEC Type              : LDP
Prefix                : 192.0.2.3/32
Source Address        : 192.0.2.1
Replying Node         : 192.0.2.3
Latest Return Code    : EgressRtr (3)
Latest Return Subcode : 1
Local BFD Discriminator : 2           Remote BFD Discriminator : 1
LSP Ping Tx Interval (s) : 60         Bootstrap Retry Count   : 0
Tx LSP Ping Requests   : 10         Rx LSP Ping Replies    : 10
-----
No. of matching BFD on LSP sessions: 2
-----

```

BFD sessions changing state are trapped, so these are logged to log 99. When the link between PE-1 and PE-2 is restored, the following messages are logged in log 99:

```

93 2023/07/10 14:13:21.913 UTC WARNING: ISIS #2045 Base VR: 1 ISIS (0) Adjacency state
"Adjacency status changed to initializing for interface: int-PE-1-PE-2, for level: l1l2, LSP-
id: 1920.0000.2002.00-00 "

```



```

94 2023/07/10 14:13:21.915 UTC WARNING: ISIS #2045 Base VR: 1 ISIS (0) Adjacency state
"Adjacency status changed to up for interface: int-PE-1-PE-2, for level: l1l2, LSP-id:
1920.0000.2002.00-00 "

95 2023/07/10 14:13:22.583 UTC WARNING: RSVP #2003 Base VR 1:
"Neighbor 192.168.12.2 on interface int-PE-1-PE-2 changed to active state"

96 2023/07/10 14:13:22.942 UTC MINOR: BFD #2004 Base 192.0.2.2
"The protocol (LDP LSP) using BFD session on node 192.0.2.2 has been cleared"

97 2023/07/10 14:13:22.942 UTC MINOR: BFD #2004 Base 192.0.2.3
"The protocol (LDP LSP) using BFD session on node 192.0.2.3 has been cleared"

98 2023/07/10 14:13:22.943 UTC MINOR: BFD #2003 Base 192.0.2.2
"The lspHead BFD Session with Local Discriminator 1 on 192.0.2.2 has been deleted"

99 2023/07/10 14:13:22.943 UTC MINOR: BFD #2003 Base 192.0.2.3
"The lspHead BFD Session with Local Discriminator 2 on 192.0.2.3 has been deleted"

100 2023/07/10 14:13:23.382 UTC MINOR: BFD #2004 Base 192.0.2.2
"The protocol (LDP LSP) using BFD session on node 192.0.2.2 has been added"

101 2023/07/10 14:13:23.414 UTC MINOR: BFD #2004 Base 192.0.2.3
"The protocol (LDP LSP) using BFD session on node 192.0.2.3 has been added"

102 2023/07/10 14:13:26.874 UTC MINOR: BFD #2002 Base 192.0.2.3
"The lspHead BFD session with Local Discriminator 4 on 192.0.2.3 is up"

103 2023/07/10 14:13:33.155 UTC MINOR: BFD #2002 Base 192.0.2.2
"The lspHead BFD session with Local Discriminator 3 on 192.0.2.2 is up"

```

The current BFD session information for LDP LSPs can be displayed using a **tools** command, as follows:

```

[/]
A:admin@PE-1# tools dump router bfd lsp-ldp prefix 192.0.2.3/32
-----
FEC: (PTR 0x10eb60ff0)
LDP: vrId: 1 (To: 192.0.2.3/32), Sender (192.0.2.1)
PingIntvl: 60 Flags: 0x6 ProtNhidx: 16 NumNextHop: 1
TempName: bfdt-2 LspName: TunnelId: 65538 NumLspUser: 0
NextHop: 192.168.12.2 IfIndex: 1 Flags: 0x0 isBackup: N
PGId: 9 [State: UP] NhIdx: 16
Label:- [0]524283
      BFD Handle: 4 State: UP LastEvent: UP
      BFD UserId: 25 TmrActive: N [0] NumRetry: 0
      DstAddr: 127.0.0.4 LocalDiscr: 4 RemoteDiscr: 0
-----
Total FEC Count in Head: 1
Total FEC Count in Tail: 0

```

The BFD templates used by LDP can also be listed using a **tools** command, as follows:

```

[/]
A:admin@PE-1# tools dump router ldp lsp-bfd bfd-templates-in-use
=====
BFD on LDP LSP BFD Template Summary
=====
Prefix List Name          BFD Template Name
-----
pfx-lst-1                 bfdt-2
-----

```

```
No. of prefix lists: 1
```

```
=====
```

Conclusion

BFD is supported for RSVP-TE and LDP LSPs and is ideal for monitoring LSPs carrying high-value services, where detecting failures in a minimum amount of time is critical.

BFD for RSVP-TE LSPs with Failure Action

This chapter describes BFD for RSVP-TE LSPs with failure action.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written based on SR OS Release 15.0.R7, but the MD-CLI in the current edition corresponds to SR OS Release 23.7.R1.

The chapter [BFD for RSVP-TE and LDP LSPs](#) is prerequisite reading.

Overview

Using the **failure-action** command, the operator can configure the action taken by the system if a BFD session fails for an RSVP LSP or LDP prefix list.

When **failure-action failover** is configured, and the LSP BFD session goes down on the currently active path, the LSP switches from the primary path to a secondary path, or from the currently active secondary path to the next best preference secondary path if the currently active path was a secondary.

When **failure-action down** is configured, the LSP is registered as unusable in the tunnel table manager (TTM) when BFD on the LSP goes down. A tunnel being registered as unusable in TTM is not available to the routing table manager (RTM) and all routes using that tunnel are withdrawn. SDP auto-bind will not use an LSP until it is registered as usable. Traffic cannot pass through that LSP, even when secondary paths are available for that LSP.

In either case, SNMP traps are raised when the BFD state machine for the LSP transitions.

Nokia recommends configuring the BFD control packet timer intervals long enough to deal with transient data path disruptions that may occur when the underlying transport network recovers following a failure.

LSP BFD only runs on the currently active path. It cannot determine if any non-active paths (for example, a secondary path or primary path during reversion) that the system might switch to are up and forwarding.

When BFD failure action is configured on an RSVP-TE LSP directly, the action can be failover or down. When BFD failure action is configured on an RSVP-TE LSP indirectly, through an LSP template, the only action available is down. This chapter only covers the direct configuration of a failure action.

Configuration

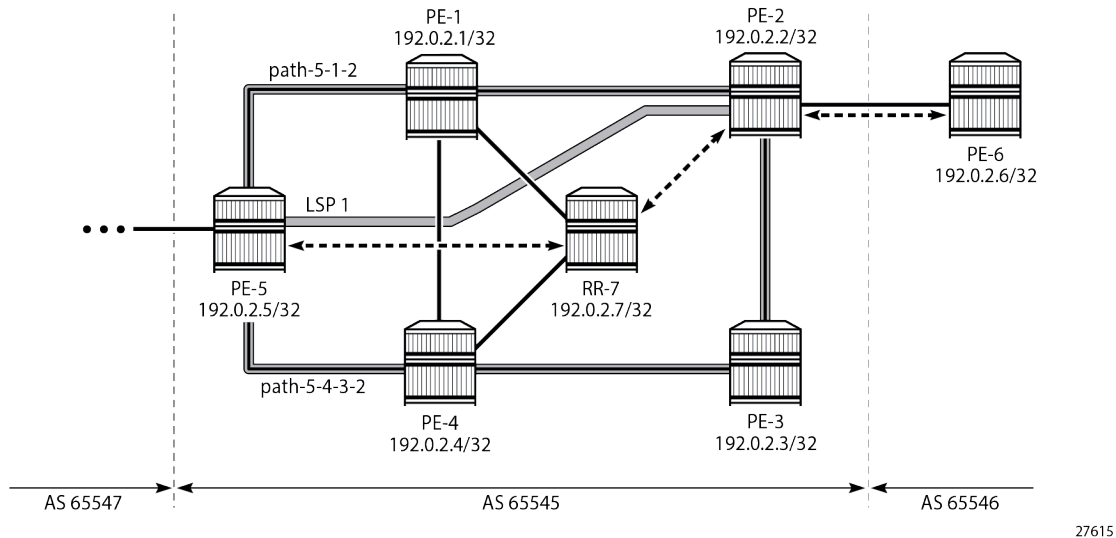
The following scenarios are described in this section:

- [Failure action failover](#)
- [Failure action down](#)

Failure action failover

Figure 186: Topology for failure action failover shows the topology used for failure action failover. A BGP shortcut is defined in AS 65545 running between the autonomous system border routers (ASBRs) PE-5 and PE-2. That shortcut is RSVP-TE LSP *lsp-1* composed of two paths, where the first path is the upper path from PE-5 via PE-1 to PE-2, and the second path is the lower path from PE-5 via PE-4 and PE-3 to PE-2.

Figure 186: Topology for failure action failover



The initial configuration includes:

- Cards, MDAs, and ports
- Router interfaces
- IS-IS as IGP on all interfaces (alternatively, OSPF can be used), with traffic engineering enabled
- MPLS and RSVP enabled on all interfaces
- BGP configured, with RR-7 being the route reflector in AS 65545 for clients PE-2 and PE-5, and PE-6 located in AS 65546 and connected to PE-2. PE-6 advertises its prefix 192.0.2.6/32 to AS 65545.

The *lsp-1* LSP is defined with primary path *path-5-1-2* and secondary path *path-5-4-3-2*. The two paths are established when the LSP is brought up, to minimize traffic loss in case of a failure. BFD template *bfd-1* with a failure action failover is applied to *lsp-1* at the LSP level.

```
# on ASBR PE-5:
configure {
  router "Base" {
    mpls {
      path "path-5-1-2" {
        admin-state enable
        hop 10 {
          ip-address 192.168.15.1
          type strict
        }
        hop 20 {
          ip-address 192.168.12.2
        }
      }
    }
  }
}
```

```

        type strict
    }
}
path "path-5-4-3-2" {
    admin-state enable
    hop 10 {
        ip-address 192.168.45.1
        type strict
    }
    hop 20 {
        ip-address 192.168.34.1
        type strict
    }
    hop 30 {
        ip-address 192.168.23.1
        type strict
    }
}
lsp "lsp-1" {
    admin-state enable
    type p2p-rsvp
    to 192.0.2.2
    path-computation-method local-cspf
    bfd {
        bfd-liveness true
        bfd-template "bfdt-1"
        failure-action failover
    }
    primary "path-5-1-2" {
    }
    secondary "path-5-4-3-2" {
        standby true
    }
}
}

```

The details of the LSP show the configured failure action, as follows:

```

[/]
A:admin@PE-5# show router mpls lsp "lsp-1" detail

=====
MPLS LSPs (Originating) (Detail)
=====
Legend :
+ - Inherited
=====
-----
Type : Originating
-----
LSP Name      : lsp-1
LSP Type      : RegularLsp          LSP Tunnel ID      : 1
LSP Index     : 1                  TTM Tunnel Id      : 1
From          : 192.0.2.5
To            : 192.0.2.2
Adm State     : Up
LSP Up Time   : 0d 00:13:02
Transitions   : 1
Retry Limit   : 0
Signaling     : RSVP
---snip---

BFD Template  : bfdt-1              BFD Ping Intvl     : 60
BFD Enable    : True                BFD Failure-action : Failover

```

```

WaitForUpTimer : 4
---snip---

Primary(a)      : path-5-1-2
                                     Up Time      : 0d 00:13:02
Bandwidth       : 0 Mbps
Standby         : path-5-4-3-2
                                     Up Time      : 0d 00:13:02
Bandwidth       : 0 Mbps
=====

```

With this configuration, the BFD session is running over the upper path, as follows:

```

[/]
A:admin@PE-5# show router bfd session

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id          State      Tx Pkts   Rx Pkts
Rem Addr/Info/SdpId Multipl   Tx Intvl  Rx Intvl
Protocols           Type     LAG Port   LAG ID
Loc Addr                               LAG name
-----
lsp-1::path-5-1-2   Up        461       421
192.0.2.2           5         2000      2000
rsvpLsp             central   N/A       N/A
192.0.2.5
-----
No. of BFD sessions: 1
=====

```

BGP route 192.0.2.6/32 is advertised by PE-6 out of AS 65546, as follows. This route has next hop 192.0.2.2, which is the system address of PE-2.

```

[/]
A:admin@PE-5# show router bgp routes

=====
BGP Router ID:192.0.2.5      AS:65545      Local AS:65545
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag  Network          LocalPref  MED
      Nexthop (Router) Path-Id     IGP Cost
      As-Path         Label
-----
u*>i  192.0.2.6/32       100        None
      192.0.2.2       1          20
      65546           -
-----
Routes : 1
=====

```

To keep the core of AS 65545 BGP free, traffic is tunneled through the *lsp-1* LSP, as follows:

```
[/]
A:admin@PE-5# show router route-table 192.0.2.6/32

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type  Proto   Age           Pref
  Next Hop[Interface Name]                       Metric
-----
192.0.2.6/32                                     Remote BGP       00h04m31s  170
  192.0.2.2 (tunneled:RSVP:1)                   20
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

The LSP and path details can also be shown using a **tools dump** command, as follows. LSP *lsp-1* is up, *path-5-1-2* is the active path taking three hops, so the operational metric is 20, and *path-5-4-3-2* is the standby path, but not active.

```
[/]
A:admin@PE-5# tools dump router mpls lspinfo "lsp-1" detail
LSP "lsp-1" LspIdx 1 LspType Dynamic State LSPS_UP  Flags 0x2000
AdminState Up OperState Up RowStatus Active
From N/A To 192.0.2.2
NumPaths 2 NumSdps 0 NumCBFSdps 0 NumFltrEntries 0
ActivePath lsp-1::path-5-1-2(LspId 56320)
HoldTimeRemaining 0secs ClassType 0 SoftPreemption TRUE Metric 0 OperMetric 20
LDPoSvp Include VprnAutoBind Include IgpShortCut Include BgpShortCut Include
BgpTransTunnel Include IpShCutTtlPropLocal TRUE IpShCutTtlPropTans TRUE
RelativeMetricOffset 2147483647 MTU 8982 InUseByLdp FALSE TTMPref 7
EntropyLabel inherit OperEntropyLabel enable NegEntropyLabel disable
ClassForwarding: Disabled
BFD Enabled Template bfdt-1 PingInterval 60 FailureAction failover WaitForUp 4sec
PCE Report: Disabled PCE Control: Disabled
Path Profile:
None
Admin Tags:
None
Lsp-self-ping: Config: inherit, Oper: Disabled, TimedOutCnt: 0 OamNoRsc: 0
Path "path-5-1-2" LspId 56320 LspPathIndex 1 PathType Primary ActivePath Yes
RowStatus Active LastChange 000 00:14:05.560
AdminState Up OperState Up OperStateChange 000 00:14:05.550
TE Computed Hop List:
Hop[1] IngIp 192.0.2.5 IngLnkId 0 EgrIp 192.168.15.2 EgrLnkId 0 RtrId 192.0.2.5 Flag 0x0
Hop[2] IngIp 192.168.15.1 IngLnkId 0 EgrIp 192.168.12.1 EgrLnkId 0 RtrId 192.0.2.1 Flag
0x0
Hop[3] IngIp 192.168.12.2 IngLnkId 0 EgrIp 192.0.2.2 EgrLnkId 0 RtrId 192.0.2.2 Flag 0x0
LspPath FsmState LSP_PATH_S_UP Flags 0x0 miscFlags 0x2
RetryAttempts 0 RetryInterval 30 NextRetryIn 0secs
FailNode 0.0.0.0 FailCode noError
Class Type 0 SetupPri 7 HoldPri 0 Pref 0 HopLimit 255 BW 0Mbps
TotIgpCost 20 OperMetric 20 MTU 8982
BFD Disabled Template n/a PingInterval 60 WaitForUp 4sec
Degraded No
Oper Values:
Class Type 0 SetupPri 7 HoldPri 0 HopLimit 255 BW 0Mbps
```

```

RecordRoute RecordLabel No Adspec
No PropagateAdminGroup Exclude 0x00000000 Include 0x00000000
No FRR
Metric 20 CSPF No Least Fill Intra-area
NegotiatedEntropyLabel Disabled
PCE-Computed No PCE-Reported No PCE-Controlled No
BFD State Up InitTime 0d 00:14:06 UpTime 0d 00:14:01
OldMBBPathsCleanedUp Yes
Path "path-5-4-3-2" LspId 56322 LspPathIndex 2 PathType Standby ActivePath No
RowStatus Active LastChange 000 00:14:05.560
AdminState Up OperState Up OperStateChange 000 00:14:05.550
TE Computed Hop List:
Hop[1] IngIp 192.0.2.5 IngLnkId 0 EgrIp 192.168.45.2 EgrLnkId 0 RtrId 192.0.2.5 Flag 0x0
Hop[2] IngIp 192.168.45.1 IngLnkId 0 EgrIp 192.168.34.2 EgrLnkId 0 RtrId 192.0.2.4 Flag
0x0
Hop[3] IngIp 192.168.34.1 IngLnkId 0 EgrIp 192.168.23.2 EgrLnkId 0 RtrId 192.0.2.3 Flag
0x0
Hop[4] IngIp 192.168.23.1 IngLnkId 0 EgrIp 192.0.2.2 EgrLnkId 0 RtrId 192.0.2.2 Flag 0x0
LspPath FsmState LSP_PATH_S_UP Flags 0x0 miscFlags 0x2
RetryAttempts 0 RetryInterval 30 NextRetryIn 0secs
FailNode 0.0.0.0 FailCode noError
Class Type 0 SetupPri 7 HoldPri 0 Pref 255 HopLimit 255 BW 0Mbps
TotIgpCost 30 OperMetric 30 MTU 8982
SRLG Disabled SRLGDisjoint No
BFD Disabled Template n/a PingInterval 60 WaitForUp 4sec
Degraded No
Oper Values:
Class Type 0 SetupPri 7 HoldPri 0 HopLimit 255 BW 0Mbps
RecordRoute RecordLabel No Adspec
No PropagateAdminGroup Exclude 0x00000000 Include 0x00000000
No FRR
Metric 30 CSPF No Least Fill Intra-area
NegotiatedEntropyLabel Disabled
PCE-Computed No PCE-Reported No PCE-Controlled No
BFD State N/A
OldMBBPathsCleanedUp Yes

Total Ingress LSP Count          : 1
    
```

Bringing down the link between PE-1 and PE-2 results in the secondary path, *path-5-4-3-2* of LSP *lsp-1*, becoming active, and the BFD session is re-established on that path, as follows:

```

[/]
A:admin@PE-5# show router bfd session

=====
Legend:
Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id                               State      Tx Pkts   Rx Pkts
Rem Addr/Info/SdpId:VcId                 Multipl   Tx Intvl  Rx Intvl
Protocols                                 Type      LAG Port  LAG ID
Loc Addr                                  LAG name
-----
lsp-1::path-5-4-3-2                     Up       4         6
192.0.2.2                                 5         2000     2000
rsvpLsp                                    central   N/A      N/A
192.0.2.5
-----
No. of BFD sessions: 1
    
```


BFD sessions changing state are logged in the trap log, as follows:

```

=====
91 2023/08/11 11:50:08.101 CEST WARNING: MPLS #2012 Base VR 1:
"LSP path lsp-1::path-5-1-2 is operationally disabled ('shutdown') because resvTear"

92 2023/08/11 11:50:08.101 CEST MINOR: MPLS #2027 Base VR 1:
"LSP lsp-1 active path lsp-1::path-5-1-2 has changed to active path lsp-1::path-5-4-3-2"

93 2023/08/11 11:50:08.101 CEST MINOR: BFD #2004 Base 192.0.2.2
"The protocol (RSVP LSP) using BFD session on node 192.0.2.2 has been cleared"

94 2023/08/11 11:50:08.101 CEST MINOR: BFD #2003 Base 192.0.2.2
"The lspHead BFD Session with Local Discriminator 1 on 192.0.2.2 has been deleted"

95 2023/08/11 11:50:08.102 CEST MINOR: BFD #2004 Base 192.0.2.2
"The protocol (RSVP LSP) using BFD session on node 192.0.2.2 has been added"

96 2023/08/11 11:50:12.944 CEST MINOR: BFD #2002 Base 192.0.2.2
"The lspHead BFD session with Local Discriminator 2 on 192.0.2.2 is up (Tunnel Id 1, Path LSP
ID 56322)"

```

The **tools dump** command shows that *lsp-1* is still up, and *path-5-4-3-2* is active with four hops, so the LSP operational metric is 30, as follows:

```

[/]
A:admin@PE-5# tools dump router mpls lspinfo "lsp-1" detail
LSP "lsp-1" LspIdx 1 LspType Dynamic State LSPS_UP Flags 0x2000
AdminState Up OperState Up RowStatus Active
From N/A To 192.0.2.2
NumPaths 2 NumSdps 0 NumCBFSdps 0 NumFltrEntries 0
ActivePath lsp-1::path-5-4-3-2(LspId 56322)
HoldTimeRemaining 0secs ClassType 0 SoftPreemption TRUE Metric 0 OperMetric 30
LDPoSvnp Include VprnAutoBind Include IgpShortCut Include BgpShortCut Include
BgpTransTunnel Include IpShCutTtlPropLocal TRUE IpShCutTtlPropTans TRUE
RelativeMetricOffset 2147483647 MTU 8982 InUseByLdp FALSE TTMPref 7
EntropyLabel inherit OperEntropyLabel enable NegEntropyLabel disable
ClassForwarding: Disabled
BFD Enabled Template bfdt-1 PingInterval 60 FailureAction failover WaitForUp 4sec
PCE Report: Disabled PCE Control: Disabled
Path Profile:
None
Admin Tags:
None
Lsp-self-ping: Config: inherit, Oper: Disabled, TimedOutCnt: 0 OamNoRsc: 0
Path "path-5-1-2" LspId 56324 LspPathIndex 1 PathType Primary ActivePath No
RowStatus Active LastChange 000 00:15:50.820
AdminState Up OperState Down OperStateChange 000 00:01:07.400
LspPath FsmState LSP_PATH_S_DOWN Flags 0x0 miscFlags 0x2
RetryAttempts 2 RetryInterval 30 NextRetryIn 32secs
FailNode 192.0.2.5 FailCode noCspfRouteToDestination
Class Type 0 SetupPri 7 HoldPri 0 Pref 0 HopLimit 255 BW 0Mbps
TotIgpCost 0 OperMetric 16777215 MTU 0
BFD Disabled Template n/a PingInterval 60 WaitForUp 4sec
Degraded No
OldMBBPathsCleanedUp Yes
Path "path-5-4-3-2" LspId 56322 LspPathIndex 2 PathType Standby ActivePath Yes
RowStatus Active LastChange 000 00:15:50.820
AdminState Up OperState Up OperStateChange 000 00:15:50.810
TE Computed Hop List:
Hop[1] IngIp 192.0.2.5 IngLnkId 0 EgrIp 192.168.45.2 EgrLnkId 0 RtrId 192.0.2.5 Flag 0x0

```

```

Hop[2] IngIp 192.168.45.1 IngLnkId 0 EgrIp 192.168.34.2 EgrLnkId 0 RtrId 192.0.2.4 Flag
0x0
Hop[3] IngIp 192.168.34.1 IngLnkId 0 EgrIp 192.168.23.2 EgrLnkId 0 RtrId 192.0.2.3 Flag
0x0
Hop[4] IngIp 192.168.23.1 IngLnkId 0 EgrIp 192.0.2.2 EgrLnkId 0 RtrId 192.0.2.2 Flag 0x0
LspPath FsmState LSP_PATH_S_UP Flags 0x0 miscFlags 0x2
RetryAttempts 0 RetryInterval 30 NextRetryIn 0secs
FailNode 0.0.0.0 FailCode noError
Class Type 0 SetupPri 7 HoldPri 0 Pref 255 HopLimit 255 BW 0Mbps
TotIgpCost 30 OperMetric 30 MTU 8982
SRLG Disabled SRLGDisjoint No
BFD Disabled Template n/a PingInterval 60 WaitForUp 4sec
Degraded No
Oper Values:
Class Type 0 SetupPri 7 HoldPri 0 HopLimit 255 BW 0Mbps
RecordRoute RecordLabel No Adspec
No PropagateAdminGroup Exclude 0x00000000 Include 0x00000000
No FRR
Metric 30 CSPF No Least Fill Intra-area
NegotiatedEntropyLabel Disabled
PCE-Computed No PCE-Reported No PCE-Controlled No
BFD State Up InitTime 0d 00:01:07 UpTime 0d 00:01:03
OldMBBPathsCleanedUp Yes

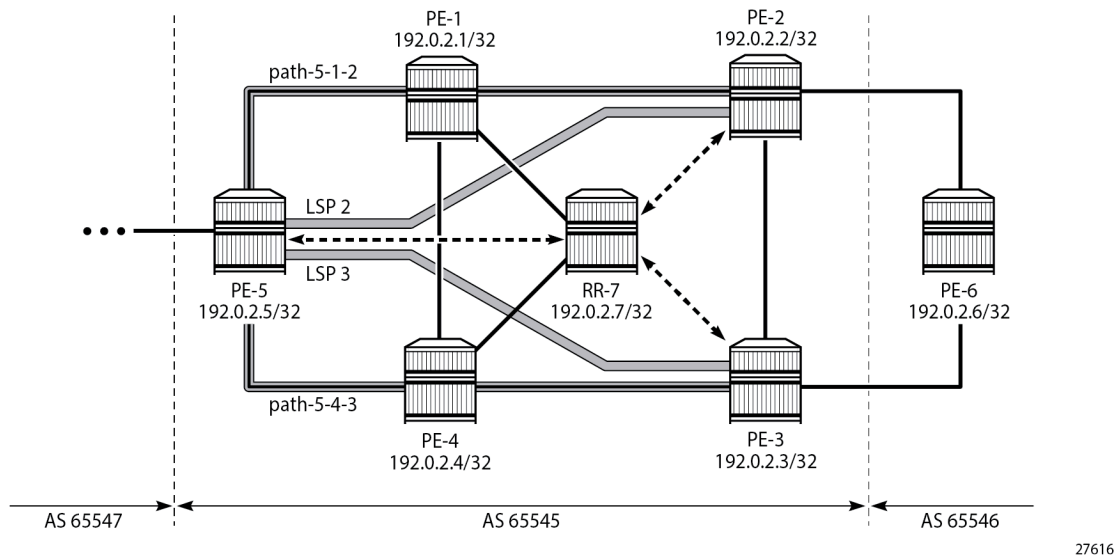
Total Ingress LSP Count          : 1
    
```

The secondary path, *path-5-4-3-2*, becoming active does not result in a change of the BGP next-hop. Traffic continues to flow from PE-5 to PE-6 via LSP *lsp-1*, but now via the lower path. The BFD failure action failover combined with standby secondary paths can help detect failures faster, with minimal traffic loss, which is especially useful in larger domains, or when the LSP passes through multiple domains.

Failure action down

[Figure 187: Topology for failure action down](#) shows the topology used for failure action down. BGP shortcuts are defined in AS 65545 running between the ASBRs PE-5, PE-2, and PE-3. A first shortcut is offered through an RSVP-TE LSP called *lsp-2*, with a single path from PE-5 via PE-1 to PE-2; the second shortcut is offered through another RSVP-TE LSP called *lsp-3*, with a single path from PE-5 via PE-4 to PE-3. When LSP *lsp-2* fails and the failure gets detected by BFD, LSP *lsp-2* is unavailable in the TTM and traffic starts using *lsp-3*, implying a change of the BGP next hop; this scenario being an edge prefix-independent convergence (PIC) scenario. See the [BGP Fast Reroute](#) chapter for more information about edge PIC.

Figure 187: Topology for failure action down



The initial configuration includes:

- Cards, MDAs, and ports
- Router interfaces
- IS-IS as IGP on all interfaces (alternatively, OSPF can be used), with traffic engineering enabled
- MPLS and RSVP-TE enabled on all interfaces
- BGP configured, with RR-7 being the route reflector for clients PE-2, PE-3, and PE-5 in AS 65545, and PE-6 located in AS 65546 and connected to PE-2 and PE-3. PE-6 exports prefix 192.0.2.6/32 to PE-2 and PE-3.

The LSPs from [Figure 187: Topology for failure action down](#) are configured with a single path, as follows. The paths referred to from these LSPs are fully strict paths, using interface IP addresses. Both *lsp-2* and *lsp-3* have BFD enabled with failure action down.

```
# on PE-5
configure {
  router "Base" {
    mpls {
      path "path-5-1-2" {
        admin-state enable
        hop 10 {
          ip-address 192.168.15.1
          type strict
        }
        hop 20 {
          ip-address 192.168.12.2
          type strict
        }
      }
      path "path-5-4-3" {
        admin-state enable
        hop 10 {
          ip-address 192.168.45.1
          type strict
        }
      }
    }
  }
}
```

```

    }
    hop 20 {
        ip-address 192.168.34.1
        type strict
    }
}
lsp "lsp-2" {
    admin-state enable
    type p2p-rsvp
    to 192.0.2.2
    path-computation-method local-cspf
    bfd {
        bfd-liveness true
        bfd-template "bfdt-1"
        failure-action down
    }
    primary "path-5-1-2" {
    }
}
lsp "lsp-3" {
    admin-state enable
    type p2p-rsvp
    to 192.0.2.3
    path-computation-method local-cspf
    bfd {
        bfd-liveness true
        bfd-template "bfdt-1"
        failure-action down
    }
    primary "path-5-4-3" {
    }
}
}

```

The details of the LSP show the configured failure action, as follows:

```

[/]
A:admin@PE-5# show router mpls lsp "lsp-2" detail

=====
MPLS LSPs (Originating) (Detail)
=====
Legend :
+ - Inherited
=====
-----
Type : Originating
-----
LSP Name      : lsp-2
LSP Type      : RegularLsp           LSP Tunnel ID      : 2
LSP Index     : 2                   TTM Tunnel Id      : 2
From          : 192.0.2.5
To            : 192.0.2.2
Adm State     : Up                   Oper State          : Up
LSP Up Time   : 0d 00:03:42          LSP Down Time      : 0d 00:00:00
Transitions   : 1                   Path Changes        : 1
Retry Limit   : 0                   Retry Timer         : 30 sec
Signaling     : RSVP                 Resv. Style         : SE
Hop Limit     : 255                 Negotiated MTU      : 8982
Adaptive      : Enabled              ClassType           : 0
FastReroute   : Disabled            Oper FR             : Disabled
PathCompMethod : local-cspf          ADSPEC              : Disabled
FallbkPathComp : not-applicable
Metric        : N/A                 Metric Type         : igp

```

```

Load Bal Wt      : N/A
Include Grps    :
None
Least Fill      : Disabled
BFD Template    : bfdt-1
BFD Enable      : True
WaitForUpTimer  : 4
ClassForwarding : Disabled
Exclude Grps    :
None
Soft Preemption : Enabled
BFD Ping Intvl  : 60
BFD Failure-action : Down

Revert Timer    : Disabled
Entropy Label   : Enabled+
Negotiated EL   : Disabled
Auto BW         : Disabled
LdpOverRsvp    : Enabled
VprnAutoBind   : Enabled
IGP Shortcut    : Enabled
IGP LFA         : Disabled
AllowSrOverSrte : Disabled
BGPTransTun    : Enabled
Oper Metric     : 20
Prop Adm Grp    : Disabled
PCE Report      : Disabled+
PCE Control     : Disabled
Path Profile    : None
Admin Tags      : None
Lsp Self Ping   : Disabled+
SelfPingOAMFail* : 0

Primary(a)      : path-5-1-2
Bandwidth       : 0 Mbps
Up Time         : 0d 00:03:42

BGP Shortcut    : Enabled
IGP Rel Metric  : Disabled

Self Ping Timeouts : 0

=====
* indicates that the corresponding row element may have been truncated.

```

Multiple BGP paths are available out of PE-5 to reach PE-6, as follows. The path via PE-2 is the currently active path, the path via PE-3 is the backup path.

```

[/]
A:admin@PE-5# show router route-table protocol bgp alternative

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age          Pref
  Next Hop[Interface Name]          Metric
  Alt-NextHop                       Alt-
                                      Metric
-----
192.0.2.6/32                       Remote BGP    00h03m19s  170
   192.0.2.2 (tunneled:RSVP:2)      20
192.0.2.6/32 (Backup)              Remote BGP    00h03m19s  170
   192.0.2.3 (tunneled:RSVP:3)      20
-----

No. of Routes: 2
Flags: n = Number of times nexthop is repeated
       Backup = BGP backup route
       LFA = Loop-Free Alternate nexthop
       S = Sticky ECMP requested
=====

```

The first of the following BFD sessions is running over the active path:

```
[/]
```

```
A:admin@PE-5# show router bfd session
```

```
=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path   pp = Protecting path
=====
BFD Session
=====
Session Id           State      Tx Pkts   Rx Pkts
  Rem Addr/Info/SdpId:VcId  Multipl   Tx Intvl  Rx Intvl
  Protocols           Type      LAG Port   LAG ID
  Loc Addr                               LAG name
-----
lsp-2::path-5-1-2           Up        143       135
  192.0.2.2                 5         2000      2000
  rsvpLsp                 central    N/A       N/A
  192.0.2.5
lsp-3::path-5-4-3           Up         145       135
  192.0.2.3                 5         2000      2000
  rsvpLsp                 central    N/A       N/A
  192.0.2.5
-----
No. of BFD sessions: 2
=====
```

The BFD session running over the active path of the *lsp-2* LSP is also indicated in the output of the following **tools** command:

```
[/]
A:admin@PE-5# tools dump router mpls lspinfo "lsp-2" detail
LSP "lsp-2" LspIdx 2 LspType Dynamic State LSPS_UP Flags 0x2000
AdminState Up OperState Up RowStatus Active
From N/A To 192.0.2.2
NumPaths 1 NumSdps 0 NumCBFSdps 0 NumFltrEntries 0
ActivePath lsp-2::path-5-1-2(LspId 56832)
HoldTimeRemaining 0secs ClassType 0 SoftPreemption TRUE Metric 0 OperMetric 20
LDPoRsvp Include VprnAutoBind Include IgpShortCut Include BgpShortCut Include
BgpTransTunnel Include IpShCutTtlPropLocal TRUE IpShCutTtlPropTans TRUE
RelativeMetricOffset 2147483647 MTU 8982 InUseByLdp FALSE TTMPref 7
EntropyLabel inherit OperEntropyLabel enable NegEntropyLabel disable
ClassForwarding: Disabled
BFD Enabled Template bfdt-1 PingInterval 60 FailureAction down WaitForUp 4sec
PCE Report: Disabled PCE Control: Disabled
Path Profile:
  None
Admin Tags:
  None
Lsp-self-ping: Config: inherit, Oper: Disabled, TimedOutCnt: 0 OamNoRsc: 0
Path "path-5-1-2" LspId 56832 LspPathIndex 1 PathType Primary ActivePath Yes
  RowStatus Active LastChange 000 00:04:25.220
  AdminState Up OperState Up OperStateChange 000 00:04:25.220
  TE Computed Hop List:
    Hop[1] IngIp 192.0.2.5 IngLnkId 0 EgrIp 192.168.15.2 EgrLnkId 0 RtrId 192.0.2.5 Flag 0x0
    Hop[2] IngIp 192.168.15.1 IngLnkId 0 EgrIp 192.168.12.1 EgrLnkId 0 RtrId 192.0.2.1 Flag
0x0
    Hop[3] IngIp 192.168.12.2 IngLnkId 0 EgrIp 192.0.2.2 EgrLnkId 0 RtrId 192.0.2.2 Flag 0x0
  LspPath FsmState LSP_PATH_S_UP Flags 0x0 miscFlags 0x2
  RetryAttempts 0 RetryInterval 30 NextRetryIn 0secs
  FailNode 0.0.0.0 FailCode noError
  Class Type 0 SetupPri 7 HoldPri 0 Pref 0 HopLimit 255 BW 0Mbps
  TotIgpCost 20 OperMetric 20 MTU 8982
  BFD Disabled Template n/a PingInterval 60 WaitForUp 4sec
```

```

Degraded No
Oper Values:
  Class Type 0 SetupPri 7 HoldPri 0 HopLimit 255 BW 0Mbps
  RecordRoute RecordLabel No Adspec
  No PropagateAdminGroup Exclude 0x00000000 Include 0x00000000
  No FRR
  Metric 20 CSPF No Least Fill Intra-area
  NegotiatedEntropyLabel Disabled
  PCE-Computed No PCE-Reported No PCE-Controlled No
  BFD State Up InitTime 0d 00:04:25 UpTime 0d 00:04:20
  OldMBBPathsCleanedUp Yes

Total Ingress LSP Count      : 1
    
```

Emulating a path failure by bringing down port 1/1/c2/1 on PE-2 brings the primary path in the *lsp-2* LSP down, as follows:

```

[/]
A:admin@PE-5# show router mpls lsp "lsp-2" path

=====
MPLS LSP lsp-2 Path
=====
-----
LSP Name      : lsp-2
From          : 192.0.2.5
To            : 192.0.2.2
Adm State     : Up
Oper State    : Down
-----
Path Name      Next Hop      Type      Out I/F  Adm  Opr
-----
path-5-1-2    n/a                Primary   n/a      Up   Dwn
=====
    
```

The only remaining BFD session is now the BFD session for the *lsp-3* LSP with next hop 192.0.2.3, as follows:

```

[/]
A:admin@PE-5# show router bfd session

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path  pp = Protecting path
=====
BFD Session
=====
Session Id      State      Tx Pkts  Rx Pkts
Rem Addr/Info/SdpId:VcId  Multipl  Tx Intvl  Rx Intvl
Protocols      Type      LAG Port   LAG ID
Loc Addr                               LAG name
-----
lsp-3::path-5-4-3      Up        188        173
192.0.2.3              5          2000       2000
  rsvpLsp          central    N/A        N/A
  192.0.2.5
-----
No. of BFD sessions: 1
=====
    
```

BGP traffic is diverted into *lsp-3*, and the BGP next hop is 192.0.2.3, as follows:

```
[/]
A:admin@PE-5# show router route-table protocol bgp alternative
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]          Metric
  Alt-NextHop                        Alt-
                                       Metric
-----
192.0.2.6/32                       Remote BGP     00h00m49s 170
   192.0.2.3 (tunneled:RSVP:3)          20
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      Backup = BGP backup route
      LFA = Loop-Free Alternate nexthop
      S = Sticky ECMP requested
=====
```



Note: Though it is possible to configure **failure-action down** for LSPs with secondary paths, no failover to any secondary path will take place. When the active path goes down, the entire LSP is registered as unusable in the TTM, regardless of any secondary paths. Because a BFD session running on a secondary unused path can be confusing to operators and is taking up resources, Nokia recommends defining the LSPs to only use a single path when failure action down is configured, as in the example.

Conclusion

The BFD failure action failover or down can help detect failures faster with minimal traffic loss on switchover, which is especially useful in larger domains or when the LSP passes through multiple domains.

DiffServ Traffic Engineering

This chapter provides information about DiffServ traffic engineering.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 14.0.R2. The CLI in the current edition corresponds to SR OS Release 23.3.R1.

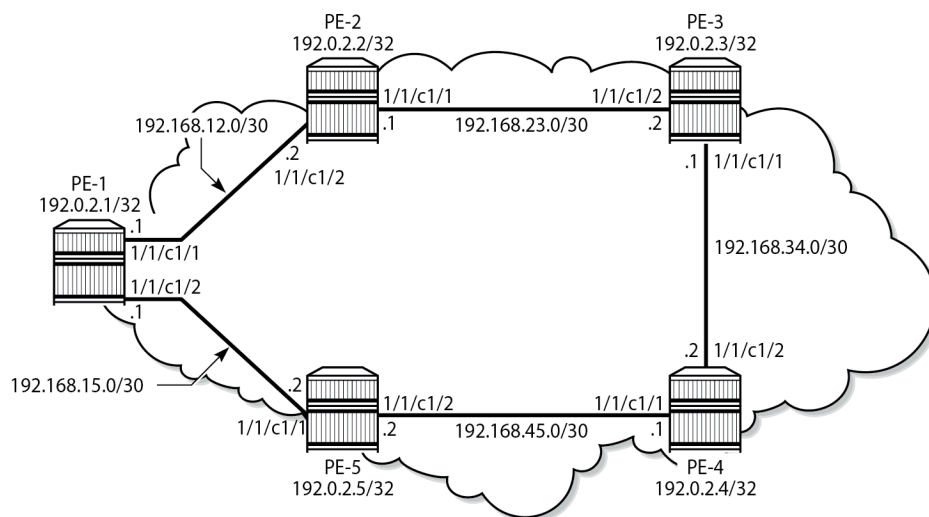
Overview

Differentiated Services (DiffServ) is a mechanism to classify and manage network traffic to provide Quality of Service (QoS). DiffServ Traffic Engineering (DiffServ TE) reserves bandwidth for Label Switched Paths (LSPs) on ReSource reservation Protocol (RSVP) interfaces on a per TE class basis.

Example Topology

[Figure 188: Example topology](#) shows the example topology that contains five 7750 SRs in a ring topology.

Figure 188: Example topology



38584

Definitions

The following definitions are used in this chapter:

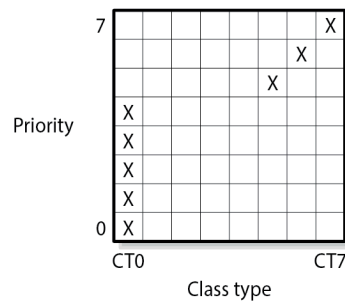
- Forwarding Classes (FCs) classify micro-flows into macro-flows. FCs can be mapped to Class Types (CTs).
- A CT is macro-flow crossing a link governed by a specific Bandwidth Constraint (BC). The BC is defined on a per-link and per-CT basis. A CT can be considered as a network-wide FC, advertised by the IGP (OSPF opaque link state advertisement (LSA), IS-IS TE Type Length Value (TLV)).
 - IGP-TE can reserve bandwidth per CT on a link (BC).
 - RSVP-TE can reserve bandwidth per LSP path, based on TE class.
- A TE class is a combination of a CT and a preemption priority.

There are eight FCs that can be mapped to CTs. The CTs range from CT0 (lowest) to CT7 (highest) and each gets a percentage of the bandwidth of the link. Each CT has eight different priority levels that are used for preemption. Even though there are 64 different potential combinations of CT and priority, only eight different combinations can be defined for TE classes. All CTs and priorities must be manually configured.

The system allows up to eight TE classes to be configured. The more TE classes are defined, the more RSVP LSPs need to be configured for each service. TE classes are consistently configured on all TE-aware Label Switching Routers (LSRs) throughout the network and advertised through the IGP.

The following shows a DiffServ TE configuration where each CT can reserve up to 10% of the maximum reservable bandwidth meaning that 20% of the bandwidth is not allocated to any CT. MPLS needs to be shut down when DiffServ TE is configured. [Figure 189: Mapping of TE classes](#) shows the mapping of the TE classes.

Figure 189: Mapping of TE classes



25847

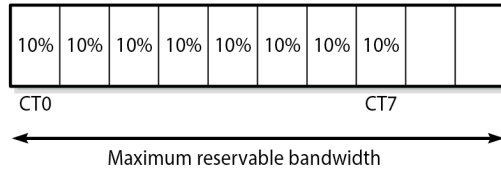
```
# on PE-1:
configure {
  router "Base" {
    rsvp {
      preemption-timer 5
      diffserv-te {
        admission-control-model mam
        class-type-bw {
          ct0 10
          ct1 10
          ct2 10
          ct3 10
        }
      }
    }
  }
}
```

```
        ct4 10
        ct5 10
        ct6 10
        ct7 10
    }
    te-class 0 {
        class-type 0
        priority 0
    }
    te-class 1 {
        class-type 0
        priority 1
    }
    te-class 2 {
        class-type 0
        priority 2
    }
    te-class 3 {
        class-type 0
        priority 3
    }
    te-class 4 {
        class-type 0
        priority 4
    }
    te-class 5 {
        class-type 5
        priority 5
    }
    te-class 6 {
        class-type 6
        priority 6
    }
    te-class 7 {
        class-type 7
        priority 7
    }
    fc af {
        class-type 0
    }
    fc be {
        class-type 0
    }
    fc ef {
        class-type 5
    }
    fc h1 {
        class-type 6
    }
    fc h2 {
        class-type 0
    }
    fc l1 {
        class-type 0
    }
    fc l2 {
        class-type 0
    }
    fc nc {
        class-type 7
    }
}
```

This configuration is applied on all TE-aware LSRs for consistency.

Figure 190: Bandwidth reservation for the CTs shows the bandwidth reservation for the CTs.

Figure 190: Bandwidth reservation for the CTs



25848

Eight TE classes are defined, each with a different priority, using four CTs (0, 5, 6 and 7). There is no need to assign any bandwidth to the unused CTs. The configuration in this example is not a recommendation. In this example, the BC for each CT is 10% of the maximum reservable bandwidth.

On each node, the RSVP status shows whether DiffServ TE is enabled and how it is configured, as follows:

```
[/]
A:admin@PE-1# show router rsvp status

=====
RSVP Status
=====
Admin Status      : Up                Oper Status      : Up
Keep Multiplier   : 3                  Refresh Time     : 30 sec
Message Pacing    : Disabled          Pacing Period    : 100 msec
Max Packet Burst  : 650 msgs         Refresh Bypass   : Disabled
Rapid Retransmit  : 5 hmsec          Rapid Retry Limit: 3
Graceful Shutdown: Disabled          SoftPreemptionTimer: 5 sec
GR Max Recovery   : 300 sec         GR Max Restart   : 120 sec
Implicit Null Label: Disabled        Node-id in RRO   : Exclude
P2P Merge Point Ab*: Disabled      P2MP Merge Point A*: Disabled
Auth Over Bypass  : Disabled
DiffServTE AdmModel: Mam          Entropy Label    : Disabled
Percent Link Bw CT0: 10           Percent Link Bw CT4: 10
Percent Link Bw CT1: 10           Percent Link Bw CT5: 10
Percent Link Bw CT2: 10           Percent Link Bw CT6: 10
Percent Link Bw CT3: 10           Percent Link Bw CT7: 10
TE0 -> Class Type : 0             Priority          : 0
TE1 -> Class Type : 0             Priority          : 1
TE2 -> Class Type : 0             Priority          : 2
TE3 -> Class Type : 0             Priority          : 3
TE4 -> Class Type : 0             Priority          : 4
TE5 -> Class Type : 5             Priority          : 5
TE6 -> Class Type : 6             Priority          : 6
TE7 -> Class Type : 7             Priority          : 7
FCName            : af             Class Type       : 0
FCName            : be             Class Type       : 0
FCName            : ef             Class Type       : 5
FCName            : h1             Class Type       : 6
FCName            : h2             Class Type       : 0
FCName            : l1             Class Type       : 0
FCName            : l2             Class Type       : 0
FCName            : nc             Class Type       : 7
IgpThresholdUpdate : Disabled
Up Thresholds(%)   : 0 15 30 45 60 75 80 85 90 95 96 97 98 99 100
Down Thresholds(%) : 100 99 98 97 96 95 90 85 80 75 60 45 30 15 0
Update Timer       : N/A
Update on CAC Fail : Disabled
=====
```

* indicates that the corresponding row element may have been truncated.

The OSPF LSAs contain BC information, as shown in the following output taken from PE-1:

```
[/]
A:admin@PE-1# show router ospf opaque-database adv-router 192.0.2.3 detail
=====
Rtr Base OSPFv2 Instance 0 Opaque Link State Database (type: All) (detail)
=====
-----
Opaque LSA
-----
Area Id       : 0.0.0.0           Adv Router Id  : 192.0.2.3
Link State Id  : 1.0.0.1           LSA Type       : Area Opaque
Sequence No    : 0x80000002        Checksum       : 0x9a28
Age           : 232                Length        : 28
Options       : E
Advertisement  : Traffic Engineering
                ROUTER-ID TLV (0001) Len 4 : 192.0.2.3
-----
Opaque LSA
-----
Area Id       : 0.0.0.0           Adv Router Id  : 192.0.2.3
Link State Id  : 1.0.0.2           LSA Type       : Area Opaque
Sequence No    : 0x8000000a        Checksum       : 0xe962
Age           : 57                 Length        : 164
Options       : E
Advertisement  : Traffic Engineering
                LINK INFO TLV (0002) Len 140 :
                Sub-TLV: 1      Len: 1      LINK_TYPE   : 1
                Sub-TLV: 2      Len: 4      LINK_ID     : 192.0.2.2
                Sub-TLV: 3      Len: 4      LOC_IP_ADDR : 192.168.23.2
                Sub-TLV: 4      Len: 4      REM_IP_ADDR : 192.168.23.1
                Sub-TLV: 5      Len: 4      TE_METRIC   : 10
                Sub-TLV: 6      Len: 4      MAX_BDWTH  : 10000000 Kbps
                Sub-TLV: 7      Len: 4      RSRVBL_BDWTH : 10000000 Kbps
                Sub-TLV: 8      Len: 32     UNRSRVD_CLS0 :
                P0: 1000000 Kbps P1: 1000000 Kbps P2: 1000000 Kbps P3: 1000000 Kbps
                P4: 1000000 Kbps P5: 1000000 Kbps P6: 1000000 Kbps P7: 1000000 Kbps
                Sub-TLV: 9      Len: 4      ADMIN_GROUP : 0 None
                Sub-TLV: 17     Len: 36     TELK_BW_CONST:
                BW Model : MAM
                BC0: 1000000 Kbps BC1: 1000000 Kbps BC2: 1000000 Kbps BC3: 1000000 Kbps
                BC4: 1000000 Kbps BC5: 1000000 Kbps BC6: 1000000 Kbps BC7: 1000000 Kbps
-----
---snip---
```

In the preceding output, only the output for the interface between PE-2 and PE-3 is shown; the output is similar for the other interfaces. On each interface between nodes, there are eight BCs for the eight CTs: from BC0 to BC7. In this example, each of the BCs has the same constraint of 1 Gb/s, which corresponds to 10% of the 10 Gb/s interfaces. As long as no LSP is configured with a CT and a bandwidth, no bandwidth is reserved. The BCs for an interface, such as the interface between PE-1 and PE-2, can be shown as follows:

```
[/]
A:admin@PE-1# show router rsvp interface "int-PE-1-PE-2" detail
=====
RSVP Interface (Detailed) : int-PE-1-PE-2
=====
```

```

-----
Interface : int-PE-1-PE-2
-----
Interface       : int-PE-1-PE-2
Port ID        : 1/1/cl/1
Admin State    : Up
Oper State     : Up
Active Sessions : 0
Active Resvs   : 0
Total Sessions : 0
Subscription   : 100 %
Port Speed     : 10000 Mbps
Total BW       : 10000 Mbps
Aggregate      : Dsabl
Hello Interval : 3000 ms
Hello Timeouts : 0
Key Type Auth  : Disabled
Keychain Auth  : Disabled
Auth Rx Seq Num : n/a
Auth Tx Seq Num : n/a
Auth Key Id    : n/a
Auth Win Size  : n/a
Refresh Reduc. : Disabled
Reliable Deli. : Disabled
Bfd Enabled    : No
Graceful Shut. : Disabled
ImplicitNullLabel : Disabled*
GR helper      : Disabled

Percent Link Bandwidth for Class Types*
Link Bw CT0    : 10
Link Bw CT1    : 10
Link Bw CT2    : 10
Link Bw CT3    : 10
Link Bw CT4    : 10
Link Bw CT5    : 10
Link Bw CT6    : 10
Link Bw CT7    : 10

Bandwidth Constraints for Class Types (Kbps)
BC0            : 1000000
BC1            : 1000000
BC2            : 1000000
BC3            : 1000000
BC4            : 1000000
BC5            : 1000000
BC6            : 1000000
BC7            : 1000000

Bandwidth for TE Class Types (Kbps)
TE0-> Resv. Bw : 0
TE1-> Resv. Bw : 0
TE2-> Resv. Bw : 0
TE3-> Resv. Bw : 0
TE4-> Resv. Bw : 0
TE5-> Resv. Bw : 0
TE6-> Resv. Bw : 0
TE7-> Resv. Bw : 0
Unresv. Bw    : 1000000
Unresv. Bw    : 1000000
Unresv. Bw    : 1000000
Unresv. Bw    : 1000000
Unresv. Bw    : 1000000
Unresv. Bw    : 1000000
Unresv. Bw    : 1000000
Unresv. Bw    : 1000000

IGP Update
Up Thresholds(%) : 0 15 30 45 60 75 80 85 90 95 96 97 98 99 100 *
Down Thresholds(%) : 100 99 98 97 96 95 90 85 80 75 60 45 30 15 0 *
IGP Update Pending : No
Next Update       : N/A

No Neighbors.
* indicates inherited values
=====

```

In this example, all BCs for the CTs are equal to 1 Gb/s, which is 10% of the maximum reservable bandwidth of 10 Gb/s. Currently no bandwidth is reserved for any of the TE classes. The unreserved bandwidth equals 1 Gb/s for TE0 through TE7.

The maximum bandwidth that can be allocated depends on the bandwidth of the link and the subscription percentage. When the subscription percentage is doubled to 200%, the BCs are doubled too, as follows:

```

# on PE-1:
configure {
  router "Base" {
    rsvp {
      interface "int-PE-1-PE-2" {

```

```

        subscription 200
    }

[/]
A:admin@PE-1# show router rsvp interface "int-PE-1-PE-2" detail

=====
RSVP Interface (Detailed) : int-PE-1-PE-2
=====
-----
Interface : int-PE-1-PE-2
-----
---snip---
Percent Link Bandwidth for Class Types*
Link Bw CT0      : 10                Link Bw CT4      : 10
Link Bw CT1      : 10                Link Bw CT5      : 10
Link Bw CT2      : 10                Link Bw CT6      : 10
Link Bw CT3      : 10                Link Bw CT7      : 10

Bandwidth Constraints for Class Types (Kbps)
BC0      : 2000000                BC4      : 2000000
BC1      : 2000000                BC5      : 2000000
BC2      : 2000000                BC6      : 2000000
BC3      : 2000000                BC7      : 2000000

Bandwidth for TE Class Types (Kbps)
TE0-> Resv. Bw   : 0                Unresv. Bw      : 2000000
TE1-> Resv. Bw   : 0                Unresv. Bw      : 2000000
TE2-> Resv. Bw   : 0                Unresv. Bw      : 2000000
TE3-> Resv. Bw   : 0                Unresv. Bw      : 2000000
TE4-> Resv. Bw   : 0                Unresv. Bw      : 2000000
TE5-> Resv. Bw   : 0                Unresv. Bw      : 2000000
TE6-> Resv. Bw   : 0                Unresv. Bw      : 2000000
TE7-> Resv. Bw   : 0                Unresv. Bw      : 2000000
---snip---
=====

```

The subscription percentage is restored to its default value of 100% as follows:

```

# on PE-1:
configure {
    router "Base" {
        rsvp {
            interface "int-PE-1-PE-2" {
                delete subscription
            }
        }
    }
}

```

Bandwidth constraint models

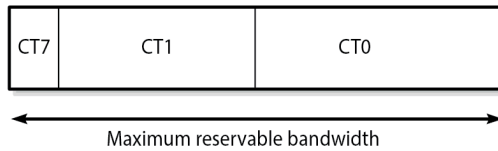
Two models are available for the bandwidth calculation that is required during the LSP setup: the Maximum Allocation Model (MAM) and the Russian Doll Model (RDM). [Table 12: Comparison bandwidth constraint models](#) shows a comparison between the two models.

Table 12: Comparison bandwidth constraint models

Maximum Allocation Model (MAM)	Russian Doll Model (RDM)
Fixed BC per CT. No bandwidth sharing between CTs.	Maps one BC to one or more CTs. Lower CTs are allowed to reserve from the unused bandwidth of the pools defined for higher CTs.
Achieves isolation between CTs and guaranteed bandwidth to CTs without the need for preemption.	No isolation between CTs. Requires preemption to guarantee bandwidth to CTs other than the premium.
Bandwidth may be wasted.	Efficient use of bandwidth.
Easy to manage.	More complex.

Figure 191: Bandwidth reservation in Maximum Allocation Model for three CTs shows the reserved bandwidth for the different class types according to the MAM model. In this example, there is only bandwidth reserved for CT0, CT1, and CT2.

Figure 191: Bandwidth reservation in Maximum Allocation Model for three CTs

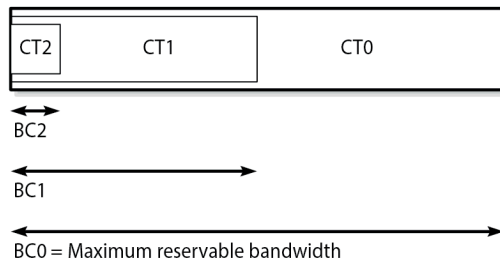


25849

Bandwidth that is reserved for a specific CT cannot be used by any other CT. Therefore, bandwidth may be wasted.

The Russian Doll Model is more flexible: when CT1 has some spare bandwidth that may be used by CT0, this is allowed. Depending on the configured setup priority and hold priority, this may be reversed when CT1 requires the bandwidth. The bandwidth reservation in the Russian Doll Model is shown in [Figure 192: Bandwidth reservation in Russian Doll Model for three CTs](#).

Figure 192: Bandwidth reservation in Russian Doll Model for three CTs



25850

Backup class types

The main CT is defined at LSP level or primary path level. The main CT is used at the first attempt for the initial establishment and re-signal Make-Before-Break (MBB) of the LSP primary path. Re-signaling of the LSP path can be triggered manually or timer-based. Subsequent retries use the backup CT, which is configured on the primary path level. Secondary paths are always signaled with the main CT. There is no verification whether the backup CT is lower than the main CT. This applies to CSPF and non-CSPF LSPs. An example of an LSP with main CT1 and backup CT0 is as follows:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      path "dyn" {
        admin-state enable
      }
      lsp "LSP-PE-1-PE-3-withBackupCT" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.3
        path-computation-method local-cspf
        class-type 1
        primary "dyn" {
          bandwidth 50
          backup-class-type 0
          priority {
            setup-priority 4
            hold-priority 4
          }
        }
      }
    }
  }
}
```

Possible triggers for using the backup CT are:

- Local interface failure or control plane failure (hello timeout)
- Received Resv message with the local-protection-in-use flag set (global revertive trigger)
- Received Patherr message with Fast ReRoute (FRR) protection active notification (global revertive trigger)
- Received Patherr message with error code 34 (Reroute) and value 1 (soft preemption trigger)
- Received Patherr message with Preemption pending flag set (soft preemption trigger)

- Received ResvTear message

When the reservable bandwidth for a CT (including the bandwidth for the inner dolls in case of RDM) is insufficient, this does not trigger the backup CT to be used. If possible, an alternate path is used for the LSP requiring this bandwidth.

Priorities

Two different priorities are linked to an LSP in a range from 0 to 7, where 0 is the highest priority and 7 the lowest. These values are important when preemption occurs, as follows:

```
# on PE-1:
A:admin@PE-1# configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3" {
        primary "dyn" {
          priority ?

priority

hold-priority          - Priority of an LSP session at preemption action
setup-priority         - Priority when insufficient bandwidth for LSP setup

where:
hold-priority <number>
<number> - <0..7>
Default - 0
and
setup-priority <number>
<number> - <0..7>
Default - 7
```

The following shows an LSP with both priorities equal to 4:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.3
        path-computation-method local-cspf
        primary "dyn" {
          bandwidth 50
          priority {
            setup-priority 4
            hold-priority 4
          }
        }
      }
    }
  }
}
```

- The first priority in the configuration is the setup priority. When an LSP is signaled and there is not enough bandwidth available on the egress Label Edge Router (eLER) or LSR, the LSP can preempt an established LSP with a hold priority lower than this setup priority. For a setup priority of 4, existing LSPs with a hold priority of 5, 6, or 7 can be preempted in case of insufficient bandwidth.
- The second priority in the configuration is the hold priority. When this LSP is established and a new LSP needs to be established, and there is insufficient bandwidth, this LSP can only be preempted by an LSP

with a higher setup priority than this hold priority. For a hold priority of 4, the LSP can be preempted by any LSP with a setup priority of 0, 1, 2, or 3.

The default values are a setup priority of 7 and a hold priority of 0. A low setup priority of 7 means the LSP cannot preempt any LSP. A high hold priority of 0 implies that the LSP cannot be preempted by any other LSP.

The setup priority needs to be lower than or equal to the hold priority to avoid preemption loops. Nokia recommends that the setup priority and the hold priority are set to equal values.

Bandwidth, CT information, and priorities are shown as follows:

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-3" path detail

=====
MPLS LSP LSP-PE-1-PE-3 Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use
  b - Bandwidth Protected      n - Node Protected
  s - Soft Preemption
  S - Strict                   L - Loose
  A - ABR                      + - Inherited
=====
-----
LSP LSP-PE-1-PE-3
Path dyn
-----
LSP Name      : LSP-PE-1-PE-3
From          : 192.0.2.1
To            : 192.0.2.3
Admin State   : Up
Path Name     : dyn
Path LSP ID   : 43008
Path Admin    : Up
Out Interface : 1/1/c1/1
Out Label     : 524287
---snip---
Neg MTU       : 1564
Bandwidth     : 50 Mbps
Hop Limit     : 255
Record Route  : Record
Record Label  : Record
Setup Priority : 4
Hold Priority  : 4
Class Type    : 0
Backup CT     : None
MainCT Retry  : n/a
Rem           :
MainCT Retry  : 0
Limit        :
---snip---
=====
```

When the LSP is being established, the path message contains the setup and hold priorities, and the required bandwidth, as follows:

```
# on PE-1:
debug
  router
    rsvp
      packet
```

```

path detail
exit

1 2023/04/12 21:29:24.744 CEST MINOR: DEBUG #2001 Base RSVP
"RSVP: PATH Msg
Send PATH From:192.0.2.1, To:192.0.2.3
      TTL:255, Checksum:0xe791, Flags:0x0
Session   - EndPt:192.0.2.3, TunnId:2, ExtTunnId:192.0.2.1
SessAttr - Name:LSP-PE-1-PE-3::dyn
          SetupPri:4, HoldPri:4, Flags:0x46
RSVPHop  - Ctype:1, Addr:192.168.12.1, LIH:2
TimeValue - RefreshPeriod:30
SendTempl - Sender:192.0.2.1, LspId:43008
SendTSpec - Ctype:QOS, CDR:50.000 Mbps, PBS:50.000 Mbps, PDR:infinity
          MPU:20, MTU:1564
LabelReq - IfType:General, L3ProtID:2048
RR0      - IpAddr:192.168.12.1, Flags:0x0
ER0      - IPv4Prefix 192.168.12.2/32, Strict
          IPv4Prefix 192.168.23.2/32, Strict
"

```

As soon as the LSP is established, the bandwidth is reserved on the interface int-PE-1-PE-2 on PE-1 in TE class 4 (configured as a combination of CT0 and priority 4), as follows:

```

[/]
A:admin@PE-1# show router rsvp interface "int-PE-1-PE-2" detail

=====
RSVP Interface (Detailed) : int-PE-1-PE-2
=====
-----
Interface : int-PE-1-PE-2
-----
Interface      : int-PE-1-PE-2
Port ID       : 1/1/c1/1
Admin State   : Up
Active Sessions : 1
Total Sessions : 1
Subscription  : 100 %
Total BW      : 10000 Mbps
Port Speed    : 10000 Mbps
Aggregate     : Dsabl
---snip---
Percent Link Bandwidth for Class Types*
Link Bw CT0   : 10
Link Bw CT1   : 10
Link Bw CT2   : 10
Link Bw CT3   : 10
Link Bw CT4   : 10
Link Bw CT5   : 10
Link Bw CT6   : 10
Link Bw CT7   : 10

Bandwidth Constraints for Class Types (Kbps)
BC0           : 1000000
BC1           : 1000000
BC2           : 1000000
BC3           : 1000000
BC4           : 1000000
BC5           : 1000000
BC6           : 1000000
BC7           : 1000000

Bandwidth for TE Class Types (Kbps)
TE0-> Resv. Bw : 0
TE1-> Resv. Bw : 0
TE2-> Resv. Bw : 0
TE3-> Resv. Bw : 0
TE4-> Resv. Bw : 50000
TE5-> Resv. Bw : 0
TE6-> Resv. Bw : 0
TE7-> Resv. Bw : 0
Unresv. Bw   : 1000000
Unresv. Bw   : 1000000
Unresv. Bw   : 1000000
Unresv. Bw   : 1000000
Unresv. Bw   : 950000
Unresv. Bw   : 1000000
Unresv. Bw   : 1000000
Unresv. Bw   : 1000000

```

---snip---

Configuration

The example topology consists of five 7750 SRs in a ring topology, as shown in [Figure 188: Example topology](#).

Initial Configuration

The nodes have the following initial configuration:

- Cards, MDAs, ports
- Router interfaces. For PE-1:

```
# on PE-1:
configure {
  router "Base" {
    interface "int-PE-1-PE-2" {
      port 1/1/c1/1
      ipv4 {
        primary {
          address 192.168.12.1
          prefix-length 30
        }
      }
    }
    interface "int-PE-1-PE-5" {
      port 1/1/c1/2
      ipv4 {
        primary {
          address 192.168.15.1
          prefix-length 30
        }
      }
    }
    interface "system" {
      ipv4 {
        primary {
          address 192.0.2.1
          prefix-length 32
        }
      }
    }
  }
}
```

- IGP: OSPF (alternatively, IS-IS could have been used) with TE enabled. For PE-1:

```
# on PE-1:
configure {
  router "Base" {
    ospf 0 {
      admin-state enable
      traffic-engineering true
      area 0.0.0.0 {
        interface "int-PE-1-PE-2" {
          interface-type point-to-point
        }
      }
    }
  }
}
```

```

        interface "int-PE-1-PE-5" {
            interface-type point-to-point
        }
        interface "system" {
        }
    }
}

```

- MPLS and RSVP enabled on all interfaces. For PE-1:

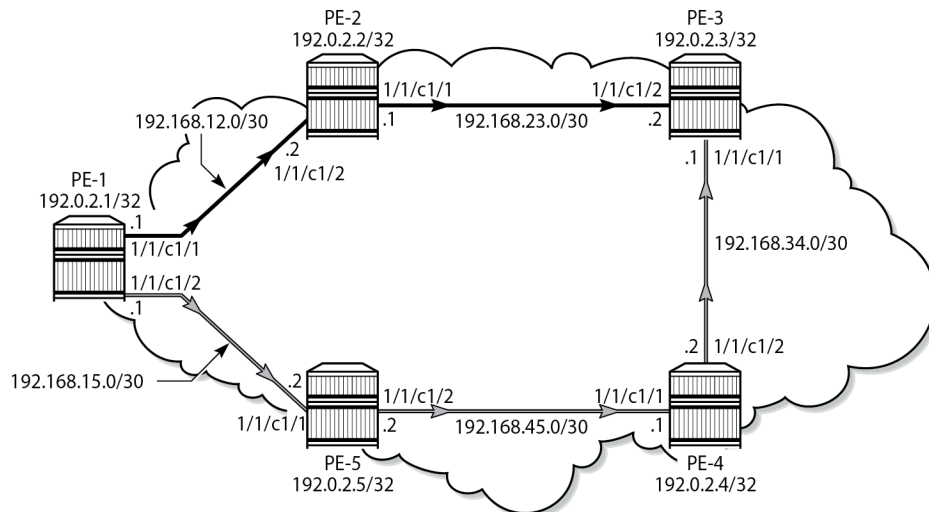
```

# on PE-1:
configure {
    router "Base" {
        mpls {
            admin-state enable
            interface "int-PE-1-PE-2" {
            }
            interface "int-PE-1-PE-5" {
            }
        }
        rsvp {
            admin-state enable
            interface "int-PE-1-PE-2" {
            }
            interface "int-PE-1-PE-5" {
            }
        }
    }
}

```

LSPs are established from PE-1 to PE-3 with the short path via PE-2 as preferred. If insufficient bandwidth is available on the short path via PE-2, the longer path via PE-5 and PE-4 is taken, as shown in [Figure 193: Paths from PE-1 to PE-3](#).

Figure 193: Paths from PE-1 to PE-3



38585

Initially, the default BC model, which is MAM, is enabled. Different LSPs are created with different class type and priority. When an LSP is established, the bandwidth reservation is verified on the interfaces of PE-1. The same LSPs are later established for the second BC model, RDM, where the bandwidth reservation is more efficient. For simplicity, FRR is not enabled on the LSPs.

Maximum Allocation Model

Enable DiffServ MAM

The DiffServ TE configuration must be consistent on all the nodes in the setup. In this example, DiffServ TE is configured as follows:

```
# on PE-1:
configure {
  router "Base" {
    rsvp {
      diffserv-te {
        admission-control-model mam
        class-type-bw {
          ct0 50
          ct1 40
          ct7 10
        }
        te-class 0 {
          class-type 0
          priority 7
        }
        te-class 1 {
          class-type 0
          priority 4
        }
        te-class 2 {
          class-type 1
          priority 7
        }
        te-class 3 {
          class-type 1
          priority 4
        }
        te-class 4 {
          class-type 2
          priority 7
        }
        te-class 5 {
          class-type 2
          priority 2
        }
        fc af {
          class-type 1
        }
        fc be {
          class-type 0
        }
        fc nc {
          class-type 2
        }
      }
    }
  }
}
```

The sum of bandwidth percentages can be lower than, but must not exceed 100%, as follows:

```
A:admin@PE-1# configure {
  router "Base" {
    rsvp {
      diffserv-te {
```

```

admission-control-model mam
class-type-bw {
    ct0 50
    ct1 50
    ct7 10
}
MINOR: RSVP #1005: configure router "Base" rsvp diffserv-te class-type-bw ct0 - Invalid
operation for RSVP instance - Total CT percent exceeds 100

```

Fewer than eight classes can be configured, as in this example.

In the example, only three CTs are used by the TE classes: CT0, CT1, and CT2. However, 10% of the maximum reservable bandwidth is allocated to CT7. Because the MAM model does not allow bandwidth allocated to a CT to be used by other CTs, only 90% of the bandwidth can be reserved: 50% to be divided between TE0 and TE1, and 40% to be divided between TE2 and TE3. TE4 and TE5 do not have any bandwidth allocated. The bandwidth allocated to CT7 is completely wasted. This is just an example, not a recommendation.

The same settings are repeated in the RDM model, where the bandwidth is not wasted. The following bandwidth information can be seen on any interface:

```

[/]
A:admin@PE-1# show router rsvp interface "int-PE-1-PE-2" detail

=====
RSVP Interface (Detailed) : int-PE-1-PE-2
=====
-----
Interface : int-PE-1-PE-2
-----
---snip---
Percent Link Bandwidth for Class Types*
Link Bw CT0      : 50          Link Bw CT4      : 0
Link Bw CT1      : 40          Link Bw CT5      : 0
Link Bw CT2      : 0           Link Bw CT6      : 0
Link Bw CT3      : 0           Link Bw CT7      : 10

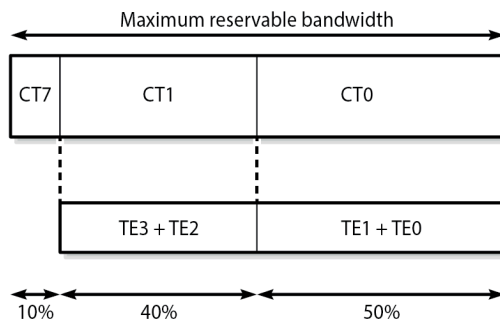
Bandwidth Constraints for Class Types (Kbps)
BC0              : 5000000      BC4              : 0
BC1              : 4000000      BC5              : 0
BC2              : 0           BC6              : 0
BC3              : 0           BC7              : 1000000

Bandwidth for TE Class Types (Kbps)
TE0-> Resv. Bw : 0           Unresv. Bw      : 5000000
TE1-> Resv. Bw : 0           Unresv. Bw      : 5000000
TE2-> Resv. Bw : 0           Unresv. Bw      : 4000000
TE3-> Resv. Bw : 0           Unresv. Bw      : 4000000
TE4-> Resv. Bw : 0           Unresv. Bw      : 0
TE5-> Resv. Bw : 0           Unresv. Bw      : 0
TE6-> Resv. Bw : 0           Unresv. Bw      : 0
TE7-> Resv. Bw : 0           Unresv. Bw      : 0
---snip---
=====

```

Figure 194: MAM bandwidth allocation shows the bandwidth allocation for the CTs and TE classes.

Figure 194: MAM bandwidth allocation



25852

Establishing LSPs

TE class 5 corresponds to CT2 and priority 2. No bandwidth can be reserved for an RSVP LSP with CT2, setup priority 2, and hold priority 2. This can be verified by configuring an empty path and an LSP, as follows:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      path "dyn" {
        admin-state enable
      }
      lsp "LSP-PE-1-PE-3-TE5" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.3
        class-type 2
        path-computation-method local-cspf
        primary "dyn" {
          bandwidth 1000
          priority {
            setup-priority 2
            hold-priority 2
          }
        }
      }
    }
  }
}
```

The path computation method must be local CSPF. The class type is by default CT0, but can be changed to CT2 by configuration. The class type can be configured in the **lsp** context, as shown here, or in the **primary path** context. The setup priority and hold priority are configured in the **primary path** context.

The LSP cannot be established, because no bandwidth is allocated to TE class 5, as follows:

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-3-TE5" path detail | match "Failure Code"
Failure Code      : noCspfRouteToDestination
```

No bandwidth is reserved on the RSVP interfaces, as follows:

```
[/]
A:admin@PE-1# show router rsvp interface
```

```

=====
RSVP Interfaces
=====
Interface                Total    Active    Total BW  Resv BW  Adm Opr
                        Sessions Sessions (Mbps)   (Mbps)
-----
system                   -        -         -         -         Up  Up
int-PE-1-PE-2           0        0        10000     0         Up  Up
int-PE-1-PE-5           0        0        10000     0         Up  Up
-----
Interfaces : 3
=====

```

Bandwidth can be reserved for an LSP with CT1 and priorities 4, as for the following LSP:

```

# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3-TE3" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.3
        path-computation-method local-cspf
        primary "dyn" {
          bandwidth 2000
          class-type 1
          priority {
            setup-priority 4
            hold-priority 4
          }
        }
      }
    }
  }
}

```

In the example, the CT is configured in the **primary path** context whereas the CT in the previous example was configured in the **lsp** context.

The path is set up via PE-2 to PE-3, as follows:

```

[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-3-TE3" path detail

```

```

=====
MPLS LSP LSP-PE-1-PE-3-TE3 Path (Detail)
=====

```

Legend :

```

@ - Detour Available          # - Detour In Use
b - Bandwidth Protected      n - Node Protected
s - Soft Preemption
S - Strict                    L - Loose
A - ABR                       + - Inherited

```

```

-----
LSP LSP-PE-1-PE-3-TE3

```

Path dyn

```

-----
LSP Name      : LSP-PE-1-PE-3-TE3
From          : 192.0.2.1
To            : 192.0.2.3
Admin State   : Up                Oper State    : Up
Path Name     : dyn
Path LSP ID   : 55296              Path Type     : Primary
Path Admin    : Up                Path Oper     : Up
Out Interface : 1/1/c1/1           Out Label     : 524287
---snip---

```

```

Actual Hops      :
  192.168.12.1(192.0.2.1)
-> 192.168.12.2(192.0.2.2)
-> 192.168.23.2(192.0.2.3)
---snip---
Record Label    : N/A
Record Label    : 524287
Record Label    : 524287
=====
    
```

Bandwidth is reserved in TE class 3, as follows:

```

[/]
A:admin@PE-1# show router rsvp interface "int-PE-1-PE-2" detail

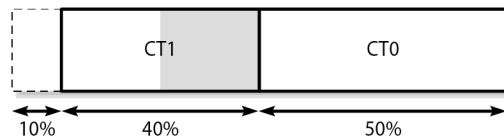
=====
RSVP Interface (Detailed) : int-PE-1-PE-2
=====

-----
Interface : int-PE-1-PE-2
-----

---snip---
Bandwidth for TE Class Types (Kbps)
TE0-> Resv. Bw : 0           Unresv. Bw : 5000000
TE1-> Resv. Bw : 0           Unresv. Bw : 5000000
TE2-> Resv. Bw : 0           Unresv. Bw : 2000000
TE3-> Resv. Bw : 2000000    Unresv. Bw : 2000000
TE4-> Resv. Bw : 0           Unresv. Bw : 0
TE5-> Resv. Bw : 0           Unresv. Bw : 0
TE6-> Resv. Bw : 0           Unresv. Bw : 0
TE7-> Resv. Bw : 0           Unresv. Bw : 0
---snip---
=====
    
```

Figure 195: Reserved and unreserved bandwidth shows the bandwidth reservation for CT1 on interface int-PE-1-PE-2 on PE-1 and on interface int-PE-2-PE-3 on PE-2.

Figure 195: Reserved and unreserved bandwidth



25853

An additional LSP is configured with CT1 and priority 4 and with CT0 as backup CT. The backup CT is not used when the amount of unreserved bandwidth for CT1 is insufficient, as in the following case where int-PE-1-PE-2 and int-PE-1-PE-5 have insufficient unreserved bandwidth for CT1:

```

# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3-TE3-backupTE1" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.3
        path-computation-method local-cspf
        primary "dyn" {
          bandwidth 5000
          class-type 1
          backup-class-type 0
        }
      }
    }
  }
}
    
```

```
priority {
  setup-priority 4
  hold-priority 4
}
```

The LSP does not come up, as follows:

```
[/]
A:admin@PE-1# show router mpls lsp

=====
MPLS LSPs (Originating)
=====
LSP Name          Tun   Fastfail  Adm  Opr
To                Id     Config
-----
LSP-PE-1-PE-3-TE5      3     No        Up   Dwn
  192.0.2.3
LSP-PE-1-PE-3-TE3      4     No        Up   Up
  192.0.2.3
LSP-PE-1-PE-3-TE3-backupTE1  5     No        Up   Dwn
  192.0.2.3
-----
LSPs : 3
=====
```

The bandwidth requirement is lowered, as follows:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3-TE3-backupTE1" {
        primary "dyn" {
          bandwidth 2500
        }
      }
    }
  }
}
```

Interface int-PE-1-PE-2 does not have sufficient bandwidth for CT1, but the longer path via PE-5 and PE-4 has sufficient unreserved bandwidth for CT1. The LSP is operationally up, as follows:

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-3-TE3-backupTE1" path detail

=====
MPLS LSP LSP-PE-1-PE-3-TE3-backupTE1 Path (Detail)
=====
Legend :
@ - Detour Available          # - Detour In Use
b - Bandwidth Protected      n - Node Protected
s - Soft Preemption
S - Strict                    L - Loose
A - ABR                       + - Inherited
=====
LSP LSP-PE-1-PE-3-TE3-backupTE1
Path dyn
-----
LSP Name      : LSP-PE-1-PE-3-TE3-backupTE1
From          : 192.0.2.1
To           : 192.0.2.3
Admin State   : Up
Oper State    : Up
```

```

Path Name      : dyn
Path LSP ID    : 32256
Path Admin     : Up
Out Interface  : 1/1/cl/2
---snip---
Setup Priority : 4
Hold Priority  : 4
Class Type    : 1
Backup CT     : 0
---snip---
Actual Hops    :
  192.168.15.1(192.0.2.1)
-> 192.168.15.2(192.0.2.5)
-> 192.168.45.1(192.0.2.4)
-> 192.168.34.1(192.0.2.3)
---snip---
=====
Path Type      : Primary
Path Oper     : Up
Out Label     : 524287
Oper SetupPriority: 4
Oper HoldPriority : 4
Oper CT       : 1
Record Label  : N/A
Record Label  : 524287
Record Label  : 524287
Record Label  : 524286
=====

```

The bandwidth reservation on RSVP interface int-PE-1-PE-2 remains unchanged, because the bandwidth is reserved on int-PE-1-PE-5, as follows:

```

[/]
A:admin@PE-1# show router rsvp interface "int-PE-1-PE-5" detail

=====
RSVP Interface (Detailed) : int-PE-1-PE-5
=====
-----
Interface : int-PE-1-PE-5
-----
Interface      : int-PE-1-PE-5
Port ID       : 1/1/cl/2
---snip---
Percent Link Bandwidth for Class Types*
Link Bw CT0   : 50
Link Bw CT1   : 40
Link Bw CT2   : 0
Link Bw CT3   : 0
Link Bw CT4   : 0
Link Bw CT5   : 0
Link Bw CT6   : 0
Link Bw CT7   : 10

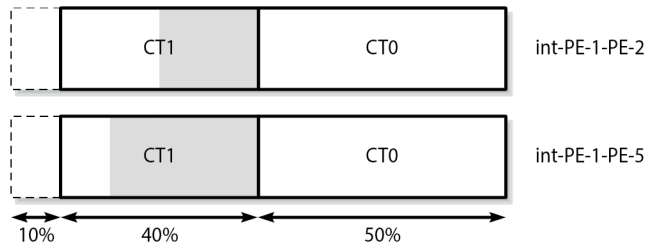
Bandwidth Constraints for Class Types (Kbps)
BC0           : 5000000
BC1           : 4000000
BC2           : 0
BC3           : 0
BC4           : 0
BC5           : 0
BC6           : 0
BC7           : 1000000

Bandwidth for TE Class Types (Kbps)
TE0-> Resv. Bw : 0
TE1-> Resv. Bw : 0
TE2-> Resv. Bw : 0
TE3-> Resv. Bw : 2500000
TE4-> Resv. Bw : 0
TE5-> Resv. Bw : 0
TE6-> Resv. Bw : 0
TE7-> Resv. Bw : 0
Unresv. Bw   : 5000000
Unresv. Bw   : 5000000
Unresv. Bw   : 1500000
Unresv. Bw   : 1500000
Unresv. Bw   : 0
Unresv. Bw   : 0
Unresv. Bw   : 0
Unresv. Bw   : 0
---snip---
=====

```

Figure 196: Reserved and unreserved bandwidth on PE-1 shows the reserved and unreserved bandwidth on the RSVP interfaces on PE-1.

Figure 196: Reserved and unreserved bandwidth on PE-1



25854

Trigger backup class-type

This mechanism is described for MAM, but it is also supported in RDM.

On PE-1, port 1/1/c1/2 is shut down. The long path via PE-5 and PE-4 can no longer be used. However, the LSP has a backup CT (CT0), which is triggered by the port being down, as follows:

```
# on PE-1:
configure {
  port 1/1/c1/2 {
    admin-state disable
  }
}
```

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-3-TE3-backupTE1" path detail
```

```
=====
MPLS LSP LSP-PE-1-PE-3-TE3-backupTE1 Path (Detail)
=====
```

Legend :

```
@ - Detour Available          # - Detour In Use
b - Bandwidth Protected      n - Node Protected
s - Soft Preemption
S - Strict                    L - Loose
A - ABR                       + - Inherited
```

```
-----
LSP LSP-PE-1-PE-3-TE3-backupTE1
Path dyn
-----
```

```
LSP Name      : LSP-PE-1-PE-3-TE3-backupTE1
From          : 192.0.2.1
To           : 192.0.2.3
Admin State   : Up                Oper State    : Up
Path Name     : dyn
Path LSP ID   : 32258              Path Type     : Primary
Path Admin    : Up                Path Oper     : Up
Out Interface : 1/1/c1/1           Out Label     : 524286
---snip---
Setup Priority : 4                Oper SetupPriority: 4
Hold Priority  : 4                Oper HoldPriority : 4
Class Type     : 1                Oper CT        : 0
Backup CT      : 0
---snip---
Actual Hops    :
```

```

192.168.12.1(192.0.2.1)          Record Label      : N/A
-> 192.168.12.2(192.0.2.2)      Record Label      : 524286
-> 192.168.23.2(192.0.2.3)      Record Label      : 524286
---snip---
=====

```

The bandwidth for this LSP is reserved in TE class 1, as follows:

```

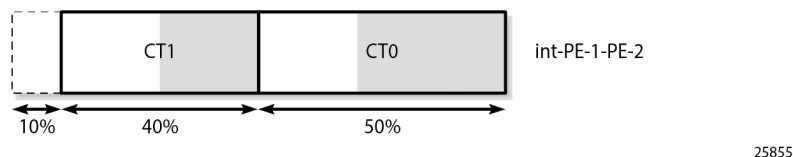
[/]
A:admin@PE-1# show router rsvp interface "int-PE-1-PE-2" detail

=====
RSVP Interface (Detailed) : int-PE-1-PE-2
=====
-----
Interface : int-PE-1-PE-2
-----
Interface      : int-PE-1-PE-2
Port ID       : 1/1/c1/1
Admin State   : Up                Oper State      : Up
Active Sessions : 2                Active Resvs    : 2
Total Sessions : 2
---snip---
Bandwidth for TE Class Types (Kbps)
TE0-> Resv. Bw : 0                Unresv. Bw     : 2500000
TE1-> Resv. Bw : 2500000          Unresv. Bw     : 2500000
TE2-> Resv. Bw : 0                Unresv. Bw     : 2000000
TE3-> Resv. Bw : 2000000          Unresv. Bw     : 2000000
TE4-> Resv. Bw : 0                Unresv. Bw     : 0
TE5-> Resv. Bw : 0                Unresv. Bw     : 0
TE6-> Resv. Bw : 0                Unresv. Bw     : 0
TE7-> Resv. Bw : 0                Unresv. Bw     : 0
---snip---
=====

```

Figure 197: Bandwidth reservation shows the bandwidth reservation on interface int-PE-1-PE-2. The bandwidth reservation on interface int-PE-2-PE-3 on PE-2 is identical.

Figure 197: Bandwidth reservation



The preceding examples illustrate that bandwidth can be wasted in the MAM model. The bandwidth allocated to CT7 cannot be used because there is no TE class configured with CT7. The bandwidth cannot be shared between CTs. The next section describes how the same LSPs are used in the RDM model. They are established one-by-one and, therefore, they are shut down, as follows:

```

# on PE-1:
configure {
  port 1/1/c1/2 {
    admin-state enable
  }
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3-TE5" {

```

```
        admin-state disable
    }
    lsp "LSP-PE-1-PE-3-TE3" {
        admin-state disable
    }
    lsp "LSP-PE-1-PE-3-TE3-backupTE1" {
        admin-state disable
    }
}
```

Russian Doll Model

Enable DiffServ RDM

The DiffServ TE configuration needs to be consistent on all the nodes in the network, as follows:

```
# on PE-1:
configure {
    router "Base" {
        rsvp {
            diffserv-te {
                admission-control-model rdm
                class-type-bw {
                    ct0 50
                    ct1 40
                    ct7 10
                }
                te-class 0 {
                    class-type 0
                    priority 7
                }
                te-class 1 {
                    class-type 0
                    priority 4
                }
                te-class 2 {
                    class-type 1
                    priority 7
                }
                te-class 3 {
                    class-type 1
                    priority 4
                }
                te-class 4 {
                    class-type 2
                    priority 7
                }
                te-class 5 {
                    class-type 2
                    priority 2
                }
                fc af {
                    class-type 1
                }
                fc be {
                    class-type 0
                }
                fc nc {
                    class-type 2
                }
            }
        }
    }
}
```



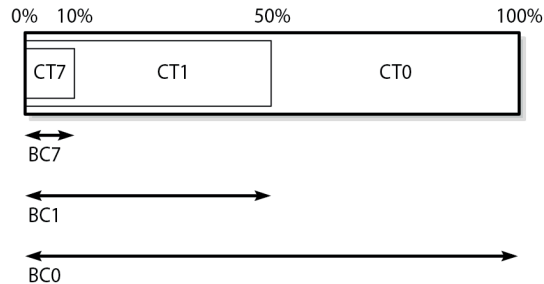
```
}

```

In this example, three FCs are mapped to CTs. FC BE corresponds to CT0, FC AF to CT1, and FC NC to CT2. CT0 can be mapped to TE class 0 for priority 7, and to TE class 1 for priority 4. The mapping is similar for CT1 (with priorities 7 and 4), and CT2 (with priorities 7 or 2).

The RDM model allows the outer dolls (lower CT) to use the unused bandwidth allocated to the inner dolls (higher CT), as shown in [Figure 198: Russian Doll Model for three class types](#).

Figure 198: Russian Doll Model for three class types



25856

The calculation of the BCs takes into account the BCs of the inner dolls, as shown in the OSPF LSAs in the opaque database, as follows:

```
[/]
A:admin@PE-1# show router ospf opaque-database adv-router 192.0.2.1 detail

=====
Rtr Base OSPFv2 Instance 0 Opaque Link State Database (type: All) (detail)
=====
-----
Opaque LSA
-----
Area Id       : 0.0.0.0           Adv Router Id  : 192.0.2.1
Link State Id : 1.0.0.1           LSA Type      : Area Opaque
Sequence No   : 0x80000003       Checksum      : 0x9035
Age           : 23              Length       : 28
Options       : E
Advertisement  : Traffic Engineering
                ROUTER-ID TLV (0001) Len 4 : 192.0.2.1
-----
Opaque LSA
-----
Area Id       : 0.0.0.0           Adv Router Id  : 192.0.2.1
Link State Id : 1.0.0.2           LSA Type      : Area Opaque
Sequence No   : 0x80000023       Checksum      : 0x302
Age           : 71              Length       : 164
Options       : E
Advertisement  : Traffic Engineering
                LINK INFO TLV (0002) Len 140 :
                Sub-TLV: 1      Len: 1      LINK_TYPE   : 1
                Sub-TLV: 2      Len: 4      LINK_ID     : 192.0.2.2
                Sub-TLV: 3      Len: 4      LOC_IP_ADDR : 192.168.12.1
                Sub-TLV: 4      Len: 4      REM_IP_ADDR : 192.168.12.2
                Sub-TLV: 5      Len: 4      TE_METRIC   : 10
                Sub-TLV: 6      Len: 4      MAX_BDWTH  : 10000000 Kbps
                Sub-TLV: 7      Len: 4      RSRVBL_BDWTH : 10000000 Kbps
                Sub-TLV: 8      Len: 32     UNRSRVD_CLS0 :
```

```

P0: 10000000 Kbps P1: 10000000 Kbps P2: 5000000 Kbps P3: 5000000 Kbps
P4: 1000000 Kbps P5: 1000000 Kbps P6: 0 Kbps P7: 0 Kbps
Sub-TLV: 9 Len: 4 ADMIN_GROUP : 0 None
Sub-TLV: 17 Len: 36 TELK_BW_CONST:
BW Model : RDM
BC0: 10000000 Kbps BC1: 5000000 Kbps BC2: 1000000 Kbps BC3: 1000000 Kbps
BC4: 1000000 Kbps BC5: 1000000 Kbps BC6: 1000000 Kbps BC7: 1000000 Kbps
---snip---
=====

```

Six TE classes are defined:

- TE0 and TE1 are defined for CT0. They can reserve all the available bandwidth, if it is not required by the other TE classes (100% = 50% for CT0 + 40% for CT1 + 10% for CT7)
- TE2 and TE3 are defined for CT1. They can reserve 50% of the bandwidth (50% = 40% for CT1 + 10% for CT7)
- TE4 and TE5 are defined for CT2. They can reserve 10% of the bandwidth, even though the configured bandwidth percentage for CT2 is 0. The 10% allocated to higher class CT7 can be used.

Bandwidth is more efficiently used in RDM than in MAM.

The BCs and bandwidth per TE class type show that bandwidth can be shared with the outer dolls, as follows:

```

[/]
A:admin@PE-1# show router rsvp interface "int-PE-1-PE-2" detail

=====
RSVP Interface (Detailed) : int-PE-1-PE-2
=====
-----
Interface : int-PE-1-PE-2
-----
Interface          : int-PE-1-PE-2
Port ID            : 1/1/cl/1
Admin State        : Up
Oper State         : Up
Active Sessions    : 0
Active Resvs       : 0
Total Sessions     : 0
Subscription       : 100 %
Port Speed         : 10000 Mbps
Total BW           : 10000 Mbps
Aggregate          : Dsabl
---snip---
Percent Link Bandwidth for Class Types*
Link Bw CT0       : 50
Link Bw CT1       : 40
Link Bw CT2       : 0
Link Bw CT3       : 0
Link Bw CT4       : 0
Link Bw CT5       : 0
Link Bw CT6       : 0
Link Bw CT7       : 10

Bandwidth Constraints for Class Types (Kbps)
BC0                : 10000000
BC1                : 5000000
BC2                : 1000000
BC3                : 1000000
BC4                : 1000000
BC5                : 1000000
BC6                : 1000000
BC7                : 1000000

Bandwidth for TE Class Types (Kbps)
TE0-> Resv. Bw    : 0
TE0-> Unresv. Bw  : 10000000
TE1-> Resv. Bw    : 0
TE1-> Unresv. Bw  : 10000000
TE2-> Resv. Bw    : 0
TE2-> Unresv. Bw  : 5000000
TE3-> Resv. Bw    : 0
TE3-> Unresv. Bw  : 5000000
TE4-> Resv. Bw    : 0
TE4-> Unresv. Bw  : 1000000
TE5-> Resv. Bw    : 0
TE5-> Unresv. Bw  : 1000000
TE6-> Resv. Bw    : 0
TE6-> Unresv. Bw  : 0
TE7-> Resv. Bw    : 0
TE7-> Unresv. Bw  : 0

```

```
---snip---
```

Establishing LSPs

LSP-PE-1-PE-3-TE5 could not be established in the MAM model, because there was no bandwidth assigned to TE5 (CT2). However, in the RDM model, TE5 can use the bandwidth of the inner doll CT7 and the LSP is operationally up, as follows:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3-TE5" {
        admin-state enable
      }
    }
  }
}
```

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-3-TE5" path detail
```

```
=====
MPLS LSP LSP-PE-1-PE-3-TE5 Path (Detail)
=====
```

Legend :

@ - Detour Available	# - Detour In Use
b - Bandwidth Protected	n - Node Protected
s - Soft Preemption	
S - Strict	L - Loose
A - ABR	+ - Inherited

```
-----
LSP LSP-PE-1-PE-3-TE5
```

```
Path dyn
-----
```

```
LSP Name      : LSP-PE-1-PE-3-TE5
From          : 192.0.2.1
To           : 192.0.2.3
Admin State   : Up
Oper State    : Up
Path Name     : dyn
Path LSP ID   : 6144
Path Admin    : Up
Path Oper     : Up
Out Interface : 1/1/c1/1
Out Label     : 524287
```

```
---snip---
```

```
Setup Priority : 2
Oper SetupPriority: 2
Hold Priority   : 2
Oper HoldPriority: 2
Class Type     : 2
Oper CT        : 2
Backup CT      : None
```

```
---snip---
```

```
Actual Hops :
  192.168.12.1(192.0.2.1)
-> 192.168.12.2(192.0.2.2)
-> 192.168.23.2(192.0.2.3)
Record Label : N/A
Record Label : 524287
Record Label : 524287
```

```
---snip---
```

The bandwidth reservation on interface int-PE-1-PE-2 is as follows:

```
[/]
A:admin@PE-1# show router rsvp interface "int-PE-1-PE-2" detail
```

```

=====
RSVP Interface (Detailed) : int-PE-1-PE-2
=====
-----
Interface : int-PE-1-PE-2
-----
---snip---
Percent Link Bandwidth for Class Types*
Link Bw CT0      : 50          Link Bw CT4      : 0
Link Bw CT1      : 40          Link Bw CT5      : 0
Link Bw CT2      : 0           Link Bw CT6      : 0
Link Bw CT3      : 0           Link Bw CT7      : 10

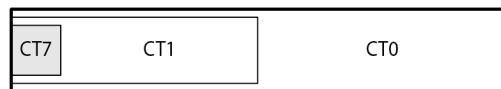
Bandwidth Constraints for Class Types (Kbps)
BC0      : 10000000          BC4      : 1000000
BC1      : 5000000          BC5      : 1000000
BC2      : 1000000          BC6      : 1000000
BC3      : 1000000          BC7      : 1000000

Bandwidth for TE Class Types (Kbps)
TE0-> Resv. Bw : 0          Unresv. Bw      : 9000000
TE1-> Resv. Bw : 0          Unresv. Bw      : 9000000
TE2-> Resv. Bw : 0          Unresv. Bw      : 4000000
TE3-> Resv. Bw : 0          Unresv. Bw      : 4000000
TE4-> Resv. Bw : 0          Unresv. Bw      : 0
TE5-> Resv. Bw : 10000000    Unresv. Bw      : 0
TE6-> Resv. Bw : 0          Unresv. Bw      : 0
TE7-> Resv. Bw : 0          Unresv. Bw      : 0
---snip---
=====

```

This LSP uses all the available bandwidth for CT7. Because TE5 is defined with the best priority (2) of all TE classes, this LSP is not preempted when a new LSP is enabled. Therefore, this bandwidth is subtracted from the amount of unreserved bandwidth. The remaining unreserved bandwidth is for CT0 and CT1 only. LSPs with other CTs cannot be established on this interface. [Figure 199: Reserved bandwidth for LSP with CT2 \(one session\)](#) shows the reserved bandwidth on interface int-PE-1-PE-2 for this LSP.

Figure 199: Reserved bandwidth for LSP with CT2 (one session)



25857

Another LSP is established: LSP-PE-1-PE-3-TE3, with CT1 and priority 4, as follows:

```

# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3-TE3" {
        admin-state enable
      }
    }
  }
}

[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-3-TE3" path detail
=====

```

```

MPLS LSP LSP-PE-1-PE-3-TE3 Path (Detail)
=====
Legend :
@ - Detour Available          # - Detour In Use
b - Bandwidth Protected      n - Node Protected
s - Soft Preemption
S - Strict                   L - Loose
A - ABR                      + - Inherited
=====
-----
LSP LSP-PE-1-PE-3-TE3
Path dyn
-----
LSP Name      : LSP-PE-1-PE-3-TE3
From          : 192.0.2.1
To            : 192.0.2.3
Admin State   : Up                Oper State      : Up
Path Name     : dyn
Path LSP ID   : 55298              Path Type       : Primary
Path Admin    : Up                Path Oper       : Up
Out Interface : 1/1/c1/1           Out Label       : 524286
---snip---
Setup Priority : 4                 Oper SetupPriority: 4
Hold Priority   : 4                 Oper HoldPriority : 4
Class Type     : 1                 Oper CT          : 1
Backup CT      : None
---snip---
Actual Hops    :
  192.168.12.1(192.0.2.1)          Record Label    : N/A
  -> 192.168.12.2(192.0.2.2)       Record Label    : 524286
  -> 192.168.23.2(192.0.2.3)       Record Label    : 524286
---snip---
=====

```

The bandwidth reservation on RSVP interface int-PE-1-PE-2 is as follows:

```

[/]
A:admin@PE-1# show router rsvp interface "int-PE-1-PE-2" detail
=====
RSVP Interface (Detailed) : int-PE-1-PE-2
=====
-----
Interface : int-PE-1-PE-2
-----
Interface      : int-PE-1-PE-2
Port ID        : 1/1/c1/1
Admin State    : Up                Oper State      : Up
Active Sessions : 2                Active Resvs    : 2
Total Sessions : 2
---snip---
Bandwidth Constraints for Class Types (Kbps)
BC0            : 10000000           BC4            : 1000000
BC1            : 5000000           BC5            : 1000000
BC2            : 1000000           BC6            : 1000000
BC3            : 1000000           BC7            : 1000000

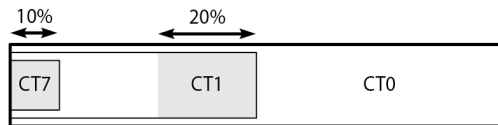
Bandwidth for TE Class Types (Kbps)
TE0-> Resv. Bw : 0                 Unresv. Bw    : 7000000
TE1-> Resv. Bw : 0                 Unresv. Bw    : 7000000
TE2-> Resv. Bw : 0                 Unresv. Bw    : 2000000
TE3-> Resv. Bw : 2000000           Unresv. Bw    : 2000000
TE4-> Resv. Bw : 0                 Unresv. Bw    : 0

```

```
TE5-> Resv. Bw : 1000000      Unresv. Bw : 0
TE6-> Resv. Bw : 0           Unresv. Bw : 0
TE7-> Resv. Bw : 0           Unresv. Bw : 0
---snip---
```

Figure 200: Bandwidth reservation for LSP with CT2 and LSP with CT1 (two sessions) shows the bandwidth reservation for the two active sessions.

Figure 200: Bandwidth reservation for LSP with CT2 and LSP with CT1 (two sessions)



25858

Another LSP is established for CT1, requesting more bandwidth than the short path via PE-2 has available. Therefore, the longer path via PE-5 and PE-4 is set up for this LSP, as follows:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3-TE3-backupTE1" {
        admin-state enable
      }
    }
  }
}
```

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-3-TE3-backupTE1" path detail
```

```
=====
MPLS LSP LSP-PE-1-PE-3-TE3-backupTE1 Path (Detail)
=====
Legend :
  @ - Detour Available      # - Detour In Use
  b - Bandwidth Protected   n - Node Protected
  s - Soft Preemption
  S - Strict                L - Loose
  A - ABR                   + - Inherited
=====
-----
LSP LSP-PE-1-PE-3-TE3-backupTE1
Path dyn
-----
LSP Name      : LSP-PE-1-PE-3-TE3-backupTE1
From          : 192.0.2.1
To           : 192.0.2.3
Admin State   : Up                Oper State      : Up
Path Name    : dyn
Path LSP ID  : 32260              Path Type       : Primary
Path Admin   : Up                Path Oper       : Up
Out Interface: 1/1/c1/2          Out Label       : 524287
---snip---
Setup Priority : 4                Oper SetupPriority: 4
Hold Priority  : 4                Oper HoldPriority : 4
Class Type    : 1                Oper CT         : 1
Backup CT     : 0
---snip---
```

```

Actual Hops      :
  192.168.15.1(192.0.2.1)
-> 192.168.15.2(192.0.2.5)
-> 192.168.45.1(192.0.2.4)
-> 192.168.34.1(192.0.2.3)
---snip---
Record Label    : N/A
Record Label    : 524287
Record Label    : 524287
Record Label    : 524285
=====

```

The bandwidth for this LSP is reserved on interface int-PE-1-PE-5, because the amount of unreserved bandwidth for TE3 is insufficient and inner dolls cannot use bandwidth assigned to outer dolls. Inner dolls are of higher priority than outer dolls, as follows:

```

[/]
A:admin@PE-1# show router rsvp interface "int-PE-1-PE-5" detail

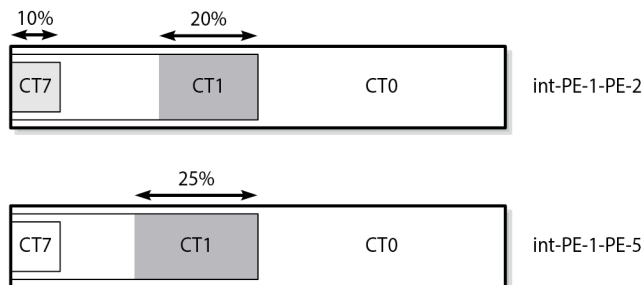
=====
RSVP Interface (Detailed) : int-PE-1-PE-5
=====
-----
Interface : int-PE-1-PE-5
-----
---snip---
Bandwidth Constraints for Class Types (Kbps)
BC0      : 10000000      BC4      : 1000000
BC1      : 5000000      BC5      : 1000000
BC2      : 1000000      BC6      : 1000000
BC3      : 1000000      BC7      : 1000000

Bandwidth for TE Class Types (Kbps)
TE0-> Resv. Bw : 0          Unresv. Bw : 7500000
TE1-> Resv. Bw : 0          Unresv. Bw : 7500000
TE2-> Resv. Bw : 0          Unresv. Bw : 2500000
TE3-> Resv. Bw : 2500000    Unresv. Bw : 2500000
TE4-> Resv. Bw : 0          Unresv. Bw : 1000000
TE5-> Resv. Bw : 0          Unresv. Bw : 1000000
TE6-> Resv. Bw : 0          Unresv. Bw : 0
TE7-> Resv. Bw : 0          Unresv. Bw : 0
---snip---
=====

```

Figure 201: Reserved bandwidth on both interfaces of PE-1 (three sessions) shows the reserved bandwidth on both interfaces of PE-1.

Figure 201: Reserved bandwidth on both interfaces of PE-1 (three sessions)



25859

The following LSP is configured on PE-1:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3-TE1" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.3
        path-computation-method local-cspf
        primary "dyn" {
          bandwidth 3000
          priority {
            setup-priority 4
            hold-priority 4
          }
        }
      }
    }
  }
}
```

The class type is by default 0. CT0 and priority 4 corresponds to TE1. There is sufficient bandwidth available on the short path via PE-2. The bandwidth reservation on RSVP interface int-PE-1-PE-2 is as follows:

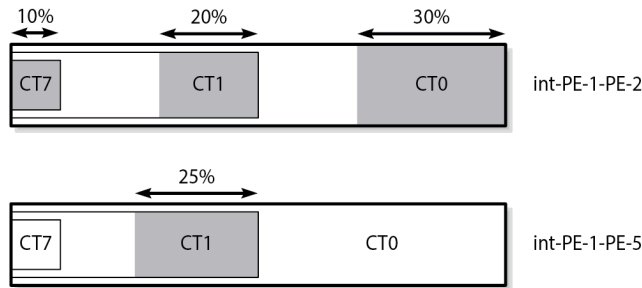
```
[/]
A:admin@PE-1# show router rsvp interface "int-PE-1-PE-2" detail

=====
RSVP Interface (Detailed) : int-PE-1-PE-2
=====
-----
Interface : int-PE-1-PE-2
-----
---snip---
Bandwidth for TE Class Types (Kbps)
TE0-> Resv. Bw   : 0                Unresv. Bw   : 4000000
TE1-> Resv. Bw   : 3000000          Unresv. Bw   : 4000000
TE2-> Resv. Bw   : 0                Unresv. Bw   : 2000000
TE3-> Resv. Bw   : 2000000          Unresv. Bw   : 2000000
TE4-> Resv. Bw   : 0                Unresv. Bw   : 0
TE5-> Resv. Bw   : 1000000          Unresv. Bw   : 0
TE6-> Resv. Bw   : 0                Unresv. Bw   : 0
TE7-> Resv. Bw   : 0                Unresv. Bw   : 0
---snip---
```

None of the established LSPs can be preempted. Therefore, the sum of the reserved and unreserved bandwidth does not exceed the total bandwidth.

[Figure 202: Reserved bandwidth on both interfaces on PE-1 \(four sessions\)](#) shows the bandwidth reservation on both interfaces.

Figure 202: Reserved bandwidth on both interfaces on PE-1 (four sessions)



25860

The following LSP with CT0 and priority 7 is configured on PE-1:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3-TE0" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.3
        path-computation-method local-cspf
        primary "dyn" {
          bandwidth 4000
          priority {
            setup-priority 7
            hold-priority 7
          }
        }
      }
    }
  }
}
```

The bandwidth is reserved in TE class 0. There is sufficient bandwidth on the short path to PE-3. The bandwidth is now reserved for 100%, as follows:

```
[/]
A:admin@PE-1# show router rsvp interface

=====
RSVP Interfaces
=====
Interface                               Total   Active   Total BW   Resv BW   Adm Opr
Sessions Sessions (Mbps) (Mbps)
-----
system                                  -       -         -         -         Up  Up
int-PE-1-PE-2                          4     4     10000    10000   Up  Up
int-PE-1-PE-5                          1       1       10000     2500     Up  Up
-----
Interfaces : 3
=====
```

The bandwidth reservation on int-PE-1-PE-2 is as follows:

```
[/]
A:admin@PE-1# show router rsvp interface "int-PE-1-PE-2" detail

=====
RSVP Interface (Detailed) : int-PE-1-PE-2
=====
```

```
-----
Interface : int-PE-1-PE-2
-----
---snip---
Percent Link Bandwidth for Class Types*
Link Bw CT0      : 50          Link Bw CT4      : 0
Link Bw CT1      : 40          Link Bw CT5      : 0
Link Bw CT2      : 0           Link Bw CT6      : 0
Link Bw CT3      : 0           Link Bw CT7      : 10

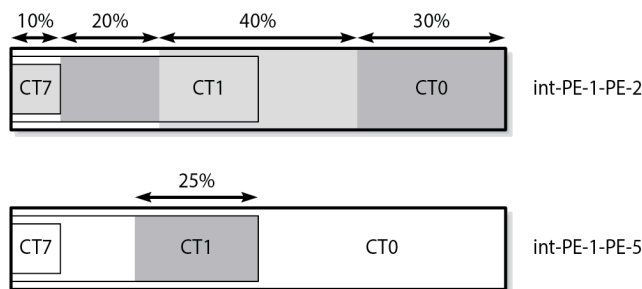
Bandwidth Constraints for Class Types (Kbps)
BC0      : 10000000          BC4      : 10000000
BC1      : 5000000          BC5      : 10000000
BC2      : 10000000         BC6      : 10000000
BC3      : 10000000         BC7      : 10000000

Bandwidth for TE Class Types (Kbps)
TE0-> Resv. Bw : 4000000    Unresv. Bw : 0
TE1-> Resv. Bw : 3000000    Unresv. Bw : 4000000
TE2-> Resv. Bw : 0           Unresv. Bw : 0
TE3-> Resv. Bw : 2000000    Unresv. Bw : 2000000
TE4-> Resv. Bw : 0           Unresv. Bw : 0
TE5-> Resv. Bw : 1000000    Unresv. Bw : 0
TE6-> Resv. Bw : 0           Unresv. Bw : 0
TE7-> Resv. Bw : 0           Unresv. Bw : 0
---snip---
=====
```

Even though the sum of the reserved bandwidth equals the maximum reservable bandwidth on the link, there is still unreserved bandwidth for specific TE classes. When an additional LSP is established requiring bandwidth in TE3 or TE1 (which have setup priority 4), it can preempt another LSP with a lower hold priority. LSPs requiring bandwidth in TE class TE2 have a setup priority 7 and cannot preempt any other LSP. The setup priority in TE1 and TE3 is 4, which is higher than the hold priority in TE2 and TE0 (7 is the lowest priority). There are no LSPs in TE2, so the only LSPs to preempt have bandwidth reserved in TE0.

Figure 203: Reserved bandwidth on both interfaces of PE-1 (five sessions) shows the bandwidth reservation on the interfaces of PE-1.

Figure 203: Reserved bandwidth on both interfaces of PE-1 (five sessions)



25861

Preemption

The following LSP is configured with CT1, setup priority 4, and hold priority 4, which corresponds to TE class 3:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3-TE3-2nd" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.3
        class-type 1
        path-computation-method local-cspf
        primary "dyn" {
          bandwidth 750
          priority {
            setup-priority 4
            hold-priority 4
          }
        }
      }
    }
  }
}
```

Because the setup priority 4 exceeds the hold priority 7 of LSP-PE-1-PE-3-TE0, this LSP preempts the existing one. The following output shows that the next hop for LSP-PE-1-PE-3-TE0 is 192.168.15.2 (PE-5), while the next hop for LSP-PE-1-PE-3-TE3-2nd is 192.168.12.2 (PE-2):

```
[/]
A:admin@PE-1# show router mpls lsp path

=====
MPLS LSP Path
=====
-----
LSP Name   : LSP-PE-1-PE-3-TE5
From       : 192.0.2.1
To         : 192.0.2.3
Adm State  : Up
Oper State : Up
-----
Path Name      Next Hop      Type      Out I/F  Adm  Opr
-----
dyn            192.168.12.2 Primary    1/1/c1/1 Up   Up
-----
LSP Name   : LSP-PE-1-PE-3-TE3
From       : 192.0.2.1
To         : 192.0.2.3
Adm State  : Up
Oper State : Up
-----
Path Name      Next Hop      Type      Out I/F  Adm  Opr
-----
dyn            192.168.12.2 Primary    1/1/c1/1 Up   Up
-----
LSP Name   : LSP-PE-1-PE-3-TE3-backupTE1
From       : 192.0.2.1
To         : 192.0.2.3
Adm State  : Up
Oper State : Up
```

```

-----
Path Name                Next Hop          Type           Out I/F   Adm  Opr
-----
dyn
                        192.168.15.2    Primary       1/1/c1/2  Up   Up

-----
LSP Name   : LSP-PE-1-PE-3-TE1
From       : 192.0.2.1
To         : 192.0.2.3
Adm State  : Up
Oper State : Up
-----
Path Name                Next Hop          Type           Out I/F   Adm  Opr
-----
dyn
                        192.168.12.2    Primary       1/1/c1/1  Up   Up

-----
LSP Name   : LSP-PE-1-PE-3-TE0
From       : 192.0.2.1
To         : 192.0.2.3
Adm State  : Up
Oper State : Up
-----
Path Name                Next Hop          Type           Out I/F   Adm  Opr
-----
dyn
                        192.168.15.2    Primary       1/1/c1/2  Up   Up

-----
LSP Name   : LSP-PE-1-PE-3-TE3-2nd
From       : 192.0.2.1
To         : 192.0.2.3
Adm State  : Up
Oper State : Up
-----
Path Name                Next Hop          Type           Out I/F   Adm  Opr
-----
dyn
                        192.168.12.2    Primary       1/1/c1/1  Up   Up
=====

```

The bandwidth reservation on RSVP interface int-PE-1-PE-2 is as follows:

```

[/]
A:admin@PE-1# show router rsvp interface "int-PE-1-PE-2" detail

=====
RSVP Interface (Detailed) : int-PE-1-PE-2
=====
Interface : int-PE-1-PE-2
-----
---snip---
Bandwidth for TE Class Types (Kbps)
TE0-> Resv. Bw : 0           Unresv. Bw : 3250000
TE1-> Resv. Bw : 3000000    Unresv. Bw : 3250000
TE2-> Resv. Bw : 0           Unresv. Bw : 1250000
TE3-> Resv. Bw : 2750000    Unresv. Bw : 1250000
TE4-> Resv. Bw : 0           Unresv. Bw : 0
TE5-> Resv. Bw : 1000000    Unresv. Bw : 0
TE6-> Resv. Bw : 0           Unresv. Bw : 0

```

```
TE7-> Resv. Bw : 0 Unresv. Bw : 0
---snip---
```

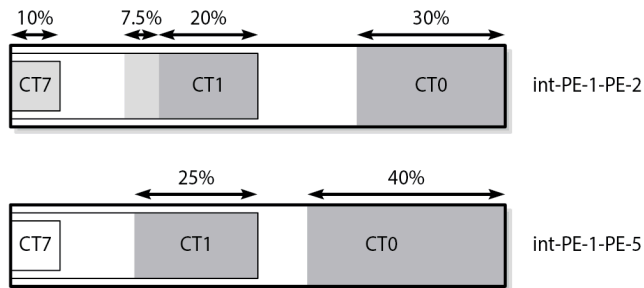
The bandwidth reservation on RSVP interface int-PE-1-PE-5 is as follows

```
[/]
A:admin@PE-1# show router rsvp interface "int-PE-1-PE-5" detail

=====
RSVP Interface (Detailed) : int-PE-1-PE-5
=====
-----
Interface : int-PE-1-PE-5
-----
---snip---
Bandwidth for TE Class Types (Kbps)
TE0-> Resv. Bw : 4000000 Unresv. Bw : 3500000
TE1-> Resv. Bw : 0 Unresv. Bw : 7500000
TE2-> Resv. Bw : 0 Unresv. Bw : 2500000
TE3-> Resv. Bw : 2500000 Unresv. Bw : 2500000
TE4-> Resv. Bw : 0 Unresv. Bw : 1000000
TE5-> Resv. Bw : 0 Unresv. Bw : 1000000
TE6-> Resv. Bw : 0 Unresv. Bw : 0
TE7-> Resv. Bw : 0 Unresv. Bw : 0
---snip---
```

Figure 204: Reserved bandwidth on both interfaces on PE-1 (six sessions) shows the bandwidth reservation on both interfaces on PE-1 for the six sessions.

Figure 204: Reserved bandwidth on both interfaces on PE-1 (six sessions)



25862

Preemption can also be within the same CT. An LSP with CT0 and priority 4 (TE1) could have preempted the LSP with CT0 and priority 7 (TE0) equally well.

Bandwidth availability check

A **tools** command can be launched to verify the available bandwidth toward a node for a specific class type (by default CT0) and priority (by default setup priority 7 and hold priority 0). The options for this command are as follows:

```
[/]
A:admin@PE-1# tools perform router mpls cspf to 192.0.2.3 ?
```

```

cspf to <IP address> [from <IP address>] [bandwidth <number>] [include-bitmap <string>]
[exclude-
  bitmap <string>] [hop-limit <number>] [exclude-address <ipv4 address>] [metric-type-te]
[strict-
  srlg] [srlg-group <number>] [exclude-node <ipv4 address>] [skip-interface <string>] [ds-
class-
  type <number>] [cspf-reqtype <keyword>] [least-fill-min-thd <number>] [setup-priority
<number>]
  [hold-priority <number>]

bandwidth          - rate-in-mbps
cspf-reqtype       - all|random|least-fill : keywords
ds-class-type      - <number> - <0..7>
exclude-address    - a.b.c.d (outbound interface)
exclude-bitmap     - string '<1..40 characters>'
exclude-node       - [a.b.c.d] (outbound interface)
from               - ipv4 address '<d.d.d.d>' or ipv6 address
                   '<(x:x:x:x:x:x:x:x)|<x:x:x:x:x:x:d.d.d.d> >'
hold-priority      - <number> - <0..7>
hop-limit          - <number> - <2..255>
include-bitmap     - [0..4294967295] - accepted in decimal, hex(0x) or binary(0b)
least-fill-min-thd - <number> - <1..100>
metric-type-te     - This element has no values
setup-priority     - <number> - <0..7>
skip-interface     - [max 32 chars]
srlg-group         - <number> - <0..4294967295>
strict-srlg        - This element has no values

```

The following verifies whether an LSP can be set up from PE-1 to PE-3 requesting 100 Mb/s with CT0 (default) and both priorities equal to 4:

```

[/]
A:admin@PE-1# tools perform router mpls cspf to 192.0.2.3 bandwidth 100 setup-priority 4 hold-
priority 4
Req CSPF for all ECMP paths
  from: this node to: 192.0.2.3 w/(DiffServ = RDM) class: 0 , setup Priority 4, Hold Priority
  4 TE Class: 1

CSPF Path
To      : 192.0.2.3
Path 1  : (cost 20)
  Src:   192.0.2.1 (= Rtr)
  Egr:   192.168.12.1          -> Ingr:   192.168.12.2          Rtr:
192.0.2.2 (met 10)
  Egr:   192.168.23.1         -> Ingr:   192.168.23.2          Rtr:
192.0.2.3 (met 10)
  Dst:   192.0.2.3 (= Rtr)

```

The short path via PE-2 has sufficient bandwidth for an LSP with these TE requirements (TE class 1 with CT0 and both priorities 4). This is different for TE class 5 (CT2 and priorities 2), where the bandwidth is completely reserved. The following shows that the longer path via PE-5 and PE-4 must be taken:

```

[/]
A:admin@PE-1# tools perform router mpls cspf to 192.0.2.3 bandwidth 100 ds-class-type 2 setup-
priority 2 hold-priority 2
Req CSPF for all ECMP paths
  from: this node to: 192.0.2.3 w/(DiffServ = RDM) class: 2 , setup Priority 2, Hold Priority
  2 TE Class: 5

CSPF Path
To      : 192.0.2.3
Path 1  : (cost 30)

```

```
Src: 192.0.2.1 (= Rtr)
Egr: 192.168.15.1 -> Ingr: 192.168.15.2 Rtr:
192.0.2.5 (met 10)
Egr: 192.168.45.2 -> Ingr: 192.168.45.1 Rtr:
192.0.2.4 (met 10)
Egr: 192.168.34.2 -> Ingr: 192.168.34.1 Rtr:
192.0.2.3 (met 10)
Dst: 192.0.2.3 (= Rtr)
```

This **tools** command can only be launched when a TE class is defined with the requested CT and priority. An error is raised when the request cannot be fulfilled, as follows:

```
[/]
A:admin@PE-1# tools perform router mpls cspf to 192.0.2.3 setup-priority 5
MINOR: CLI #2005: Error while processing command - No Te class mapped to Class Type 0 , Setup
Priority 5.

[/]
A:admin@PE-1# tools perform router mpls cspf to 192.0.2.3 setup-priority 4
MINOR: CLI #2005: Error while processing command - No Te class mapped to Class Type 0 , Hold
Priority 0.
```

Conclusion

DiffServ TE enforces different BCs for different classes of traffic. DiffServ TE controls overbooking and supports preemption. Two BC models are described in this chapter: MAM and RDM.

IGP Shortcuts

This chapter provides information about IGP shortcuts.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter is applicable to SR OS when the feature is not related to BGP. There are no other prerequisites for this configuration. This chapter was initially written for SR OS Release 12.0.R3, but the MD-CLI in the current edition corresponds to SR OS Release 21.2.R1.

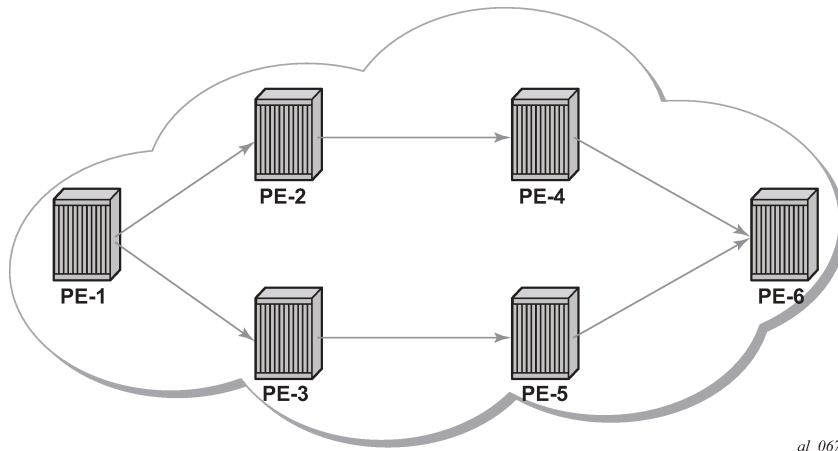
Overview

Interior Gateway Protocols (IGPs) are routing protocols that operate inside an Autonomous System (AS). An AS is a network domain that is managed under a single administration. Because the scope of operation of an IGP is usually within an AS, IGPs are also called intra-AS protocols. The purpose of an IGP is to provide reachability information to destination nodes that are inside the domain. IGPs can be one or more of a variety of protocols, including routing protocols such as Routing Information Protocol (RIP) version 1 or 2, Open Shortest Path First (OSPF), and Intermediate System to Intermediate System (IS-IS).

IGPs such as OSPF and IS-IS are link-state protocols that use a Shortest Path First (SPF) algorithm to compute the shortest path tree to all nodes in a network. The results of such computations indicate the destination node, next hop address, and output interface, where the output interface is a physical interface. Optionally, Multi-Protocol Label Switching (MPLS) Label Switched Paths (LSPs) can be included in the SPF algorithm on the node performing the calculations, as LSPs behave as logical interfaces directly connected to remote nodes in the network. Because the SPF algorithm treats the LSPs in the same way as a physical interface (being a potential output interface), the computation results could be to select a destination node together with an output LSP, using the LSP as a shortcut through the network to the destination node.

[Figure 205: Normal SPF Tree Sourced by PE-1](#) shows a normal SPF tree sourced by PE-1 (Provider Edge-1).

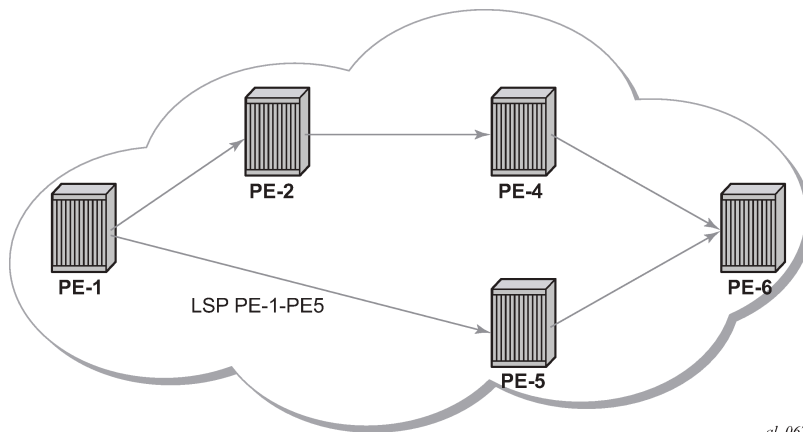
Figure 205: Normal SPF Tree Sourced by PE-1



al_0674

If there is an LSP that connects PE-1 to PE-5, and IGP shortcuts are configured on PE-1, the SPF tree will be as shown in [Figure 206: SPF Tree Sourced by PE-1 Using LSP Shortcuts](#).

Figure 206: SPF Tree Sourced by PE-1 Using LSP Shortcuts



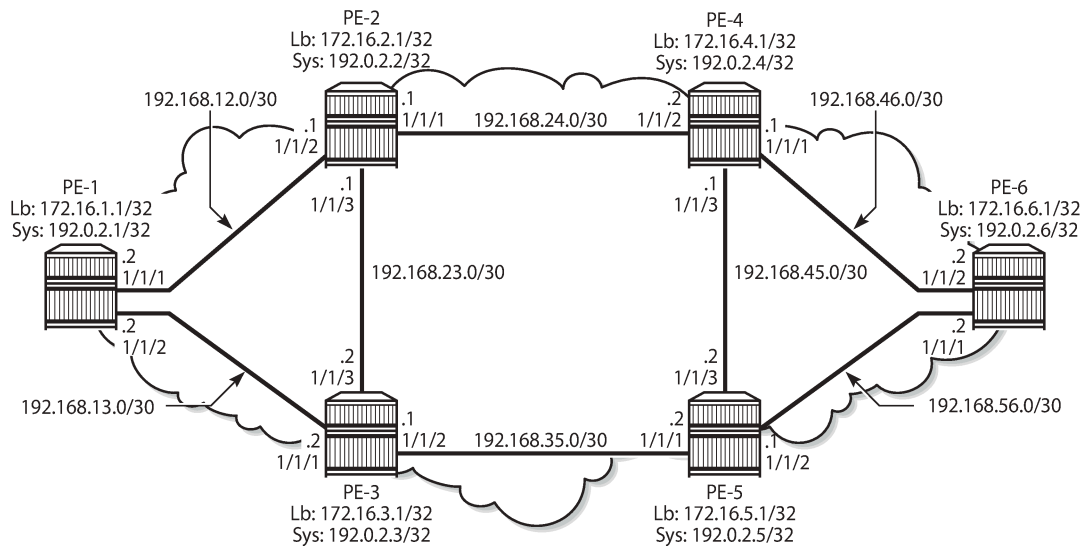
al_0675

IGP shortcuts are enabled on a per router basis; SPF computations are independent and irrelevant to other routers, so there is no need to enable shortcuts on every single router.

The example topology used in this example is shown in [Figure 207: Example Topology](#). The setup consists of six 7750 service routers. There is a single AS and a single IGP area. The following configuration tasks should be completed first:

- IS-IS or OSPF on all interfaces within the AS (configuration has been done using IS-IS but using OSPF shows exactly the same behavior).
- Label Distribution Protocol (LDP) and Resource Reservation Protocol (RSVP) on all interfaces within the AS.

Figure 207: Example Topology



26168

In all figures, **Lb** stands for Loopback and **Sys** stands for the system IP addresses.

Configuration

The first step is to configure the IGP (IS-IS) on all nodes, where IS-IS redistributes route reachability to all routers. To facilitate IS-IS configuration, all routers are L2-L1 capable within the same IS-IS area-id so there is only a single topology area in the network (all routers share the same topology). Traffic engineering (TE) is enabled on the IGP as it is a requirement for RSVP. The metric is using the default values: because no reference bandwidth command is used, the default metric of 10 is applicable on all interfaces. The configuration for PE-2 is as follows.

```
[/]
A:admin@PE-2# configure {
  router "Base" {
    interface "int-PE-2-PE-1" {
      port 1/1/2
      ipv4 {
        primary {
          address 192.168.12.2
          prefix-length 30
        }
      }
    }
    interface "int-PE-2-PE-3" {
      port 1/1/3
      ipv4 {
        primary {
          address 192.168.23.1
          prefix-length 30
        }
      }
    }
  }
  interface "int-PE-2-PE-4" {
```



```

    int-PE-2-PE-1                                0
192.168.13.0/30                                Remote  ISIS    00h02m31s 15
    192.168.12.1                                20
192.168.23.0/30                                Local   Local   00h02m32s  0
    int-PE-2-PE-3                                0
192.168.24.0/30                                Local   Local   00h02m32s  0
    int-PE-2-PE-4                                0
192.168.35.0/30                                Remote  ISIS    00h02m05s 15
    192.168.23.2                                20
192.168.45.0/30                                Remote  ISIS    00h01m51s 15
    192.168.24.2                                20
192.168.46.0/30                                Remote  ISIS    00h01m51s 15
    192.168.24.2                                20
192.168.56.0/30                                Remote  ISIS    00h01m31s 15
    192.168.23.2                                30
-----
No. of Routes: 14
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

LDP and RSVP Shortcuts

Interface Label Distribution Protocol (iLDP) is enabled on all interfaces (except system interfaces, which is not allowed) in all routers. The configuration on all nodes is similar and the IP addresses are derived from [Figure 207: Example Topology](#). The configuration of PE-4 is as follows:

```

[/]
A:admin@PE-4# configure {
  router "Base" {
    ldp {
      interface-parameters {
        interface "int-PE-4-PE-2" {
          ipv4 {
          }
        }
        interface "int-PE-4-PE-5" {
          ipv4 {
          }
        }
        interface "int-PE-4-PE-6" {
          ipv4 {
          }
        }
      }
    }
  }
}

```

With iLDP enabled, PE-4 establishes iLDP sessions with its directly connected neighbors, as follows:

```

[/]
A:admin@PE-4# show router ldp session ipv4

=====
LDP IPv4 Sessions
=====
Peer LDP Id          Adj Type  State          Msg Sent  Msg Recv  Up Time

```

```
-----
192.0.2.2:0      Link      Established  54      54      0d 00:01:54
192.0.2.5:0      Link      Established  49      50      0d 00:01:36
192.0.2.6:0      Link      Established  42      43      0d 00:01:19
-----
No. of IPv4 Sessions: 3
=====
```

The following tunnel table shows that there is a Label Switched Path (LSP) to every other router. The reason is that the LDP label distribution mode is downstream unsolicited (DU) by default, originating label bindings for system addresses only (which are used by iLDP as transport address by default). The command also shows the preference of the LSPs (where the preference is 9 for LDP) and the metric of the LSPs (the metric is inherited from the IGP, each hop counts as a metric of 10), as follows. The metric to destinations PE-1 and PE-3 is 20 because there are two hops in between (PE-4 is two hops away from PE-1 and PE-3).

```
[/]
A:admin@PE-4# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.1/32     ldp       MPLS  65538     9     192.168.24.1  20
192.0.2.2/32     ldp       MPLS  65537     9     192.168.24.1  10
192.0.2.3/32     ldp       MPLS  65539     9     192.168.24.1  20
192.0.2.5/32     ldp       MPLS  65540     9     192.168.45.2  10
192.0.2.6/32     ldp       MPLS  65541     9     192.168.46.2  10
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
```

In order to configure RSVP shortcuts, RSVP must be enabled on all interfaces where traffic engineering is required, but in this example, MPLS and RSVP are enabled on all interfaces of the network. By default, MPLS is enabled on the system interface, therefore, it need not be configured explicitly. When RSVP is in **admin-state enable**, it is automatically configured on the interfaces where MPLS is configured. The configuration for PE-6 is as follows.

```
[/]
A:admin@PE-6# configure {
  router "Base" {
    mpls {
      admin-state enable
      interface "int-PE-6-PE-4" {
      }
      interface "int-PE-6-PE-5" {
      }
    }
    rsvp {
      admin-state enable
      interface "int-PE-6-PE-4" {
      }
      interface "int-PE-6-PE-5" {
      }
    }
  }
}
```

```
}
}
```

The configuration of the other nodes is similar. The IP addresses can be derived from [Figure 207: Example Topology](#). Because there are no RSVP LSPs configured yet, the tunnel table has no RSVP LSPs and only contains LDP LSPs.

LDP Static Route (IP Tunneled in LDP Tunnel)

Using LDP LSP shortcuts for static route resolution enables forwarding of IPv4 packets over LDP LSPs instead of using a regular IP next hop. In other words, the traffic to the resolved static routes is forwarded using MPLS LDP LSP rather than plain IP.

The configuration defines a static route pointing to the destination PE (remote loopback, which is an indirect next hop in the example), and explicitly indicates that it should use LDP rather than IGP. Taking PE-1 and PE-6 as an example, two loopback interfaces are configured (172.16.X.1/32), where X = PE number, and a static route is defined according to the preceding explanation. The following shows the configuration on PE-1.

```
[/]
A:admin@PE-1# configure {
  router "Base" {
    interface "loopback" {
      ipv4 {
        primary {
          address 172.16.1.1
          prefix-length 32
        }
      }
    }
    loopback
  }
  static-routes {
    route 172.16.6.1/32 route-type unicast {
      indirect 192.0.2.6 {
        admin-state enable
        tunnel-next-hop {
          disallow-igp true
          resolution filter
          resolution-filter {
            ldp true
          }
        }
      }
    }
  }
}
}
```

Looking at the GRT or forwarding information base (FIB), there are two new entries corresponding to the two configured loopback interfaces. One entry has the protocol set to LOCAL (the local loopback on the PE), and the other entry has the protocol set to STATIC, where the next hop is reached using an LDP LSP.

```
[/]
A:admin@PE-1# show router fib 1
```

```
=====
FIB Display
=====
```

Prefix [Flags] NextHop	Protocol

172.16.1.1/32	LOCAL
172.16.1.1 (loopback)	
172.16.6.1/32	STATIC
192.0.2.6 (Transport:LDP)	
192.0.2.1/32	LOCAL
192.0.2.1 (system)	
192.0.2.2/32	ISIS
192.168.12.2 (int-PE-1-PE-2)	
192.0.2.3/32	ISIS
192.168.13.2 (int-PE-1-PE-3)	
192.0.2.4/32	ISIS
192.168.12.2 (int-PE-1-PE-2)	
192.0.2.5/32	ISIS
192.168.13.2 (int-PE-1-PE-3)	
192.0.2.6/32	ISIS
192.168.12.2 (int-PE-1-PE-2)	
192.168.12.0/30	LOCAL
192.168.12.0 (int-PE-1-PE-2)	
192.168.13.0/30	LOCAL
192.168.13.0 (int-PE-1-PE-3)	
192.168.23.0/30	ISIS
192.168.12.2 (int-PE-1-PE-2)	
192.168.24.0/30	ISIS
192.168.12.2 (int-PE-1-PE-2)	
192.168.35.0/30	ISIS
192.168.13.2 (int-PE-1-PE-3)	
192.168.45.0/30	ISIS
192.168.12.2 (int-PE-1-PE-2)	
192.168.46.0/30	ISIS
192.168.12.2 (int-PE-1-PE-2)	
192.168.56.0/30	ISIS
192.168.13.2 (int-PE-1-PE-3)	

Total Entries : 16	
=====	

The following output shows that a **ping** sourced by the loopback interface on PE-1 is able to reach the loopback interface on PE-6, and **tracert** demonstrates that the traffic is following the LDP LSP. The **ping** and **tracert** traffic cannot follow the IGP path because the static route command states that the IGP is disallowed when no LDP LSP toward PE-6 is available (also, the loopback interfaces are not enabled on IS-IS).

```
[/]
A:admin@PE-1# ping 172.16.6.1 source ping 172.16.6.1 source-address 172.16.1.1
PING 172.16.6.1 56 data bytes
64 bytes from 172.16.6.1: icmp_seq=1 ttl=64 time=3.18ms.
64 bytes from 172.16.6.1: icmp_seq=2 ttl=64 time=3.30ms.
. 64 bytes from 172.16.6.1: icmp_seq=3 ttl=64 time=3.03ms.
. 64 bytes from 172.16.6.1: icmp_seq=4 ttl=64 time=3.25ms.
. 64 bytes from 172.16.6.1: icmp_seq=5 ttl=64 time=2.72ms.

---- 172.16.6.1 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 2.72ms, avg = 3.10ms, max = 3.30ms, stddev = 0.208ms
```

```
[/]
A:admin@PE-1# traceroute 172.16.6.1 source traceroute 172.16.6.1 source-address 172.16.1.1
traceroute to 172.16.6.1 from 172.16.1.1, 30 hops max, 40 byte packets
```

```

1 0.0.0.0 * * *
2 0.0.0.0 * * *
3 172.16.6.1 (172.16.6.1) 2.79 ms 2.35 ms 2.61 ms

```

With the **traceroute** command, there are three hops from PE-1 to PE-6. There is no information regarding IP for the first two hops because the traffic is encapsulated in an MPLS LSP. The reason why the hops are displayed even when there is an MPLS LSP tunnel is because by default, the SR router propagates (copies) the Time To Live (TTL) from the IP header in the MPLS header. This is known as uniform mode.

However, a service provider might not want to show how many MPLS hops (nodes) there are in their network if a **traceroute** command is executed from outside their network. To prevent internal hops being shown, **no propagate** commands are needed in the LDP configuration, as follows. This is known as pipe mode.

```

[/]
A:admin@PE-1# configure {
  router "Base" {
    ldp {
      shortcut-local-ttl-propagate false
      shortcut-transit-ttl-propagate false
    }
  }
}

```

When TTL propagation is disabled, the hops are not displayed any longer when running the **traceroute** command.

```

[/]
A:admin@PE-1# traceroute 172.16.6.1 source-address 172.16.1.1
traceroute to 172.16.6.1 from 172.16.1.1, 30 hops max, 40 byte packets
1 172.16.6.1 (172.16.6.1) 1.86 ms 2.04 ms 1.96 ms

```

For more information about uniform mode and pipe mode, see the "Tunneling of ICMP Reply Packets over MPLS LSPs" chapter in the MPLS volume of the *7450 ESS, 7750 SR, and 7950 XRS Advanced Configuration Guide — Book I*.

RSVP Static Route (IP Tunneled in RSVP Tunnel)

Using RSVP LSP shortcuts for static route resolution enables forwarding of IPv4 packets over RSVP LSPs instead of using a regular IP next hop. In other words, the traffic to the resolved static routes is forwarded using an MPLS RSVP LSP rather than plain IP.

The configuration defines a static route pointing to a destination PE (remote loopback, which is an indirect next hop in the example), and explicitly indicates that it should use RSVP rather than IGP. Taking PE-6 and PE-1 as an example, two loopback interfaces are configured (172.16.X.1/32), where X = PE number, and a static route is defined according to the preceding explanation. The following shows the configuration on PE-6.

```

[/]
A:admin@PE-6# configure {
  router "Base" {
    static-routes {
      route 172.16.1.1/32 route-type unicast {
        indirect 192.0.2.1 {
          admin-state enable
          tunnel-next-hop {

```


The RSVP LSP is used to resolve the indirect next hop (PE-1 system address) in the static route (the LSP used is identified with the tunnel ID, in this case 1), therefore, the route for prefix 172.16.1.1 in the GRT looks as follows:

```
[/]
A:admin@PE-6# show router route-table 172.16.1.1

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type   Proto   Age      Pref
  Next Hop[Interface Name]                Metric
-----
172.16.1.1/32                    Remote Static 00h00m00s 5
  192.0.2.1 (tunneled:RSVP:1)                1
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

As in the LDP shortcut with static route example, between PE-6 and PE-1, TTL propagation is disabled, as follows.

```
[/]
A:admin@PE-1# configure {
  router "Base" {
    ldp {
      shortcut-local-ttl-propagate false
      shortcut-transit-ttl-propagate false
    }
  }
}
```

The output is the following when running a traceroute:

```
[/]
A:admin@PE-6# traceroute 172.16.1.1 source traceroute 172.16.1.1 source-address 172.16.6.1
traceroute to 172.16.1.1 from 172.16.6.1, 30 hops max, 40 byte packets
 1 172.16.1.1 (172.16.1.1) 2.81 ms 3.24 ms 3.02 ms
```

The two static routes that have been defined to use the LDP and RSVP shortcuts follow the static routes default values and have a preference of 5 and a metric of 1.

LDP Shortcut for IGP Route Resolution

Using LDP shortcuts for IGP route resolution enables forwarding of packets to IGP learned routes over an LDP LSP. The default is to disable the LDP shortcut across all interfaces in the node.

When LDP shortcuts are enabled, LDP populates the Route Table Manager (RTM) with next hop entries corresponding to all prefixes for which it activated an LDP Forwarding Equivalence Class (FEC). For a prefix, two route entries are populated in the RTM. One corresponds to the LDP shortcut next hop and has an owner of LDP. The other one is the regular IP next hop. The LDP shortcut next hop always takes preference over the regular IP next hop for forwarding user packets and specific control packets over an outgoing interface to the route next hop.

When LDP has activated a FEC for a prefix and programmed the RTM, it also programs the ingress tunnel table in the line card with the LDP tunnel information.

When an IPv4 packet is received on an ingress network interface and the preferred RTM entry corresponds to an LDP shortcut, a subscriber Internet Enhanced Service (IES) interface, or a regular IES interface, the lookup of the packet by the ingress line card results in the packet being sent labeled with the label stack corresponding to the Next Hop Label Forwarding Entry (NHLFE) of the LDP LSP. If the preferred RTM entry corresponds to an IP next hop, the IPv4 packet is forwarded unlabeled. The activation of the FEC by LDP is done by performing an exact match with an IGP route prefix in the RTM, but it can also be done by performing a longest prefix match with an IGP route in the RTM if the aggregate-prefix-match option is enabled globally in LDP.

Handling of Control Packets

All control plane packets will not see the LDP shortcut route entry in the RTM with the exception of the following control packets which will be forwarded over an LDP shortcut when enabled:

- A locally generated or in transit ICMP ping and UDP traceroute of an IGP route. The transit message appears as a user packet to the ingress LER node.
- A locally generated response to a received ICMP ping or UDP traceroute message.

All other control plane packets that require an RTM lookup and have knowledge of which destination is reachable over the LDP shortcut will continue to be forwarded over the IP next hop route in the RTM.

Handling of Multicast Packets

LDP shortcuts apply to unicast FEC types and are used for forwarding IP unicast packets in the data path. IP multicast packets forwarded over an multicast Label Distribution Protocol (mLDP) Point-to-Multi-Point (P2MP) LSP make use of a multicast FEC and thus cannot make use of the LDP unicast shortcut.

ECMP Considerations

When Equal Cost Multi-Path (ECMP) is enabled and multiple equal cost next hops exist for the IGP route, the ingress line card will spray the packets for this route based on the hashing routine supported for IPv4 packets. When the preferred RTM entry corresponds to an LDP shortcut route, spraying is performed across the multiple next hops for the LDP FEC. The FEC next hops can either be direct link LDP neighbors, or T-LDP (targeted LDP) neighbors reachable over RSVP LSPs in the case of LDP-over-RSVP, but not both. This is as per ECMP for LDP in the existing implementation. When the preferred RTM entry corresponds to a regular IP route, spraying will be performed across regular IP next hops for the prefix. Spraying across regular IP next hops and LDP shortcut next hops concurrently is not supported.

Configuring IGP LDP shortcuts is straightforward, and only applies to the node where there is interest to provision the LDP shortcut. In this example, only PE-1 is provisioned with LDP shortcuts, as follows:

```
[/]
A:admin@PE-1# configure {
  router "Base" {
    ldp {
      ldp-shortcut {
        ipv4 true
      }
    }
  }
}
```

```
}  
}
```

Now, all tunnel LSPs that resolve an IGP next hop will replace the IP next hops, as shown in the following output:

```
[/]
A:admin@PE-1# show router route-table

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]
Type Proto Age Pref
Metric
-----
192.0.2.1/32 Local Local 00h50m05s 0
system 0
192.0.2.2/32 Remote LDP 00h00m00s 9
192.168.12.2 (tunneled) 10
192.0.2.3/32 Remote LDP 00h00m00s 9
192.168.13.2 (tunneled) 10
192.0.2.4/32 Remote LDP 00h00m00s 9
192.168.12.2 (tunneled) 20
192.0.2.5/32 Remote LDP 00h00m00s 9
192.168.13.2 (tunneled) 20
192.0.2.6/32 Remote LDP 00h00m00s 9
192.168.12.2 (tunneled) 30
192.168.12.0/30 Local Local 00h50m05s 0
int-PE-1-PE-2 0
192.168.13.0/30 Local Local 00h50m05s 0
int-PE-1-PE-3 0
192.168.23.0/30 Remote ISIS 00h49m41s 15
192.168.12.2 20
192.168.24.0/30 Remote ISIS 00h49m41s 15
192.168.12.2 20
192.168.35.0/30 Remote ISIS 00h49m20s 15
192.168.13.2 20
192.168.45.0/30 Remote ISIS 00h49m06s 15
192.168.12.2 30
192.168.46.0/30 Remote ISIS 00h49m06s 15
192.168.12.2 30
192.168.56.0/30 Remote ISIS 00h48m46s 15
192.168.13.2 30
-----
No. of Routes: 14
Flags: n = Number of times nexthop is repeated
B = BGP backup route available
L = LFA nexthop available
S = Sticky ECMP requested
=====
```

```
[/]
A:admin@PE-1# show router fib 1

=====
FIB Display
=====
Prefix [Flags] Protocol
NextHop
-----
192.0.2.1/32 LOCAL
192.0.2.1 (system)
```

```

192.0.2.2/32                                LDP
  192.0.2.2 (Transport:LDP)
192.0.2.3/32                                LDP
  192.0.2.3 (Transport:LDP)
192.0.2.4/32                                LDP
  192.0.2.4 (Transport:LDP)
192.0.2.5/32                                LDP
  192.0.2.5 (Transport:LDP)
192.0.2.6/32                                LDP
  192.0.2.6 (Transport:LDP)
192.168.12.0/30                              LOCAL
  192.168.12.0 (int-PE-1-PE-2)
192.168.13.0/30                              LOCAL
  192.168.13.0 (int-PE-1-PE-3)
192.168.23.0/30                              ISIS
  192.168.12.2 (int-PE-1-PE-2)
192.168.24.0/30                              ISIS
  192.168.12.2 (int-PE-1-PE-2)
192.168.35.0/30                              ISIS
  192.168.13.2 (int-PE-1-PE-3)
192.168.45.0/30                              ISIS
  192.168.12.2 (int-PE-1-PE-2)
192.168.46.0/30                              ISIS
  192.168.12.2 (int-PE-1-PE-2)
192.168.56.0/30                              ISIS
  192.168.13.2 (int-PE-1-PE-3)
-----
Total Entries : 14
-----
=====

```

Applying LDP IGP shortcuts only on PE-1 implies that IP traffic from PE-1 to any of the system addresses of the rest of the nodes will use the LDP shortcut, however, the traffic replied from any PE back to PE-1 will be native IP because IGP shortcuts have not been provisioned in the other nodes.

RSVP Shortcut for IGP Route Resolution

Using RSVP LSP shortcuts when resolving IGP routes enables forwarding of packets to IGP learned routes over an RSVP LSP. The use of RSVP shortcuts for resolving IGP routes is enabled at the IS-IS (or OSPF) routing protocol level or at the LSP level, and instructs IS-IS and OSPF to include RSVP LSPs originating on this node and terminating on the system address (router ID) of a remote node and considers them as direct links. RSVP LSPs with a destination address corresponding to an interface address or any other loopback interface address of a remote node are automatically not considered by IS-IS or OSPF.

By default, RSVP shortcuts are disabled in all IGP instances.

RSVP LSPs are included in the IGP SPF computation with the following characteristics:

- RSVP LSP is modeled as a point-to-point link IP interface and its metric is used in the computation of the shortest path of IGP routes
- Next hop and interface include the NHLFE of the shortcut LSP when the IGP path cost using the RSVP LSP is the best.
- Shortcuts are not used when the destination RSVP LSP is in a different IGP area. In addition, IGP adjacencies across an RSVP LSP are not supported.

RSVP shortcuts are enabled at IGP instance level as follows:

```
[/]
```

```
A:admin@<all PEs># configure {
  router "Base" {
    isis 0 {
      igp-shortcut {
        admin-state enable
        tunnel-next-hop {
          family ipv4 {
            resolution filter
            resolution-filter {
              rsvp true
            }
          }
        }
      }
    }
  }
}
```

The configuration can be done at the IGP level or per LSP level. When RSVP shortcuts are enabled at the IGP instance level, all RSVP LSPs originating on this node are eligible by default. The user can, however, exclude a specific RSVP LSP from being used as a shortcut for resolving IGP routes by entering the command (e.g. on PE-6):

```
[/]
A:admin@PE-6# configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-6-PE-1" {
        delete igp-shortcut
      }
    }
  }
}
```

As RSVP shortcuts can coexist with LDP shortcuts or IP next hops, SPF computation and path selection follows the procedures in RFC 3906:

- SPF picks the RSVP shortcut next hop if there is an RSVP LSP directly to that address regardless of the path cost compared to the IGP next hop.
- SPF picks the RSVP shortcut next hop or the IGP next hop based on path lowest cost if there is an IGP path to the prefix that does not go via the tail-end of the LSP.
- If the IGP next hop is picked, then it can be an LDP shortcut next hop or a regular IP next hop. The LDP shortcut next hop always has preference over the regular IP next hop.

Handling of Control Packets

All control plane packets requiring an RTM lookup and whose destination is reachable over the RSVP shortcut are forwarded over the shortcut. This is because the RTM keeps a single route entry for each prefix, except if there is ECMP over different outgoing interfaces. Interface bound control packets are not impacted by the RSVP shortcut because RSVP LSPs with a destination address different than the router ID are not included by IGP in its SPF calculation.

RSVP shortcuts for IGP shortcut resolution should only be used with CSPF LSPs or with fully explicit path non-CSPF LSPs. RSVP hop-by-hop Path messages will try to use the shortcut and consequently LSPs without CSPF enabled, or that use a loose/empty hop path, will not come up. However, LSPs with CSPF enabled or using a strict hop path will come up. This is because in the former case, the RTM lookup to get

the next hop results in using the shortcut and so the path messages are sent directly to the destination of the LSP, where they are dropped. With CSPF enabled, the next hop (and the entire path) is provided by CSPF and the path messages are sent unlabeled to the directly connected neighbor which corresponds to the next hop of the destination of the LSP. Similar processing occurs if a strict hop path is used, as is the case in the following example.

Handling of Multicast Packets

IP multicast packets cannot be forwarded over an RSVP shortcut, they can only be forwarded over an RSVP P2MP LSP. However, RSVP shortcut routes appear in the RTM and are seen by all applications when they are the best route. When the Reverse Path Forwarding (RPF) check for the source of the multicast packet matches an RSVP shortcut route, the check will pass if both the RSVP shortcut and the multicast-import options are enabled in the IGP, as follows, because the RTM is populated with next hops only and not with tunnels (RPFs will fail for source prefixes resolved to a tunnel NH).

```
[/]
A:admin@PE-6# configure {
  router "Base" {
    isis multicast-import ?

  multicast-import

  ipv4          - Submit IPv4 routes into the multicast RPF of the RTM
  ipv6          - Submit IPv6 routes into the multicast RPF of the RTM
```

The unicast RTM can still use the tunnel next hop for the same prefix. SPF keeps track of both the direct first hop and the tunneled first hop of a node which is added to the Dijkstra tree.

ECMP Considerations

When ECMP is enabled and multiple equal cost paths exist for the route over a set of tunnel next hops based on the hashing routine supported for IPv4 packets, there are two possibilities:

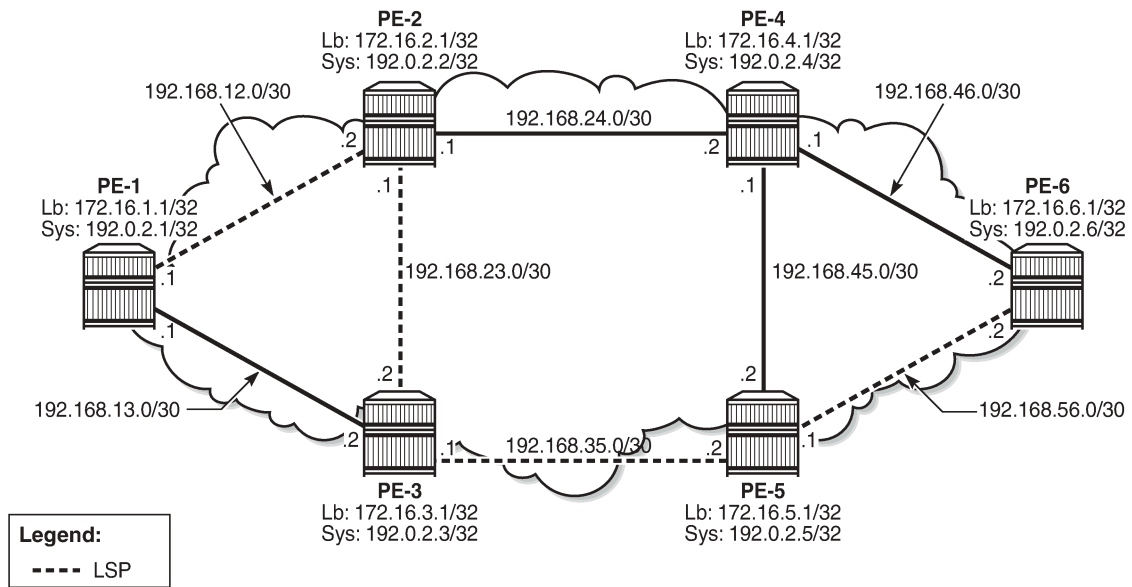
- Destination is tunnel endpoint: the system selects the tunnel with lowest tunnel ID (IP next hop is never used).
- Destination is different from the tunnel endpoint: it selects tunnel endpoints when the LSP metric is not greater than the IGP cost and it prefers tunnel endpoint over IP next hop.

ECMP is not performed across the IP and tunnel next hops simultaneously.

RSVP Shortcuts Configuration

Configuring RSVP LSP shortcuts is straightforward, and only applies to the node where there is interest to provision the RSVP shortcut. Two LSPs, from PE-6 to PE-1 and from PE-1 to PE-6, with strict hops, are provisioned according to [Figure 208: LSPs Between PE-1 and PE-6](#).

Figure 208: LSPs Between PE-1 and PE-6



25826

The configuration on PE-1 and PE-6 is similar (replacing the IP addresses), so only the configuration for PE-6 is shown:

```
[/]
A:admin@PE-6# configure {
  router "Base" {
    isis 0 {
      igp-shortcut {
        admin-state enable
        tunnel-next-hop {
          family ipv4 {
            resolution filter
            resolution-filter {
              rsvp true
            }
          }
        }
      }
    }
  }
}
```

```
[/]
A:admin@PE-6# configure {
  router "Base" {
    mpls {
      path "path-to-PE-1" {
        admin-state enable
        hop 10 {
          ip-address 192.0.2.5
          type strict
        }
        hop 20 {
          ip-address 192.0.2.3
        }
      }
    }
  }
}
```



```
B = BGP backup route available
L = LFA nexthop available
S = Sticky ECMP requested
```

The RSVP LSP in the output has a metric of 16777215, the LSP administrative metric matches the maximum value allowed for an IS-IS link using the wide-metric (24-bit value with a range of [0 — 16777215]). The following metric rules apply:

- A dynamic strict path non-CSPF LSP has the maximum metric (16777215).
- A dynamic CSPF LSP has a metric equal to the cumulative IGP cost.
 - If the user enabled the use of the TE metric on this LSP ([/] A:admin@<a PE># configure { router "Base" { mpls { lsp <LSP Name> { path-computation-method local-cspf metric-type te } } } }), then the metric for the LSP is the maximum (16777215).
 - If the user enabled the use of the TE metric on this LSP and provisioned a specific metric on the lsp ([/] A:admin@<a PE># configure { router "Base" { mpls { lsp <LSP Name> { path-computation-method local-cspf metric-type te metric <value> } } } }), then the metric for the LSP is the one provisioned. When configuring the metric of an LSP, the parameter "metric-type" is not required.
- A static LSP has a maximum metric (16777215).
- Manual and dynamic bypass LSPs have the maximum metric (16777215).

The RSVP shortcuts section detailed the importance of the LSP metric when using CSPF LSPs or when importing RSVP tunnel links into the IGP. The LSP metric can be inherited from the IGP, or can be manually modified by configuring a specific LSP metric or relative metric offset. Because IP and LDP FECs resolve to RSVP LSPs when the metric is equal or lower compared to the regular routing metric, configuring a specific static LSP metric (lower than the IGP metric) or relative metric offset is strongly recommended when using RSVP shortcuts, so that the GRT and LDP FEC resolution will always prefer the RSVP LSP shortcuts when the CSPF path computation is not using the shortest path.

For the preceding example, the first rule applies.

Advertising RSVP LSP Tunnel Links in the IGP: Forwarding Adjacency Feature

If configured, an RSVP LSP can also be advertised into the IGP similar to regular links so that other routers in the network can include that RSVP LSP into their SPF computations. The forwarding adjacency feature can be enabled independently from the RSVP shortcut feature in CLI. If both are configured for an IGP instance, the forwarding adjacency takes precedence. An RSVP LSP must exist in the reverse direction in order for the advertised link to pass the bi-directional link check and be usable by other routers in the network. However, this is not required for the node which originates the LSP. The LSP is advertised as an unnumbered point-to-point link and the Link State Protocol data unit (LSP) and Link State Advertisement (LSA) have no traffic engineering opaque sub-TLVs as per RFC 3906.

Reusing the RSVP IGP shortcuts set up previously (PE-1 and PE-6 RSVP IGP shortcut example according to [Figure 208: LSPs Between PE-1 and PE-6](#)), the outcome is a route linked with an RSVP LSP as next hop, as follows:

```
[/]
A:admin@PE-6# show router route-table 192.0.2.1/32

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type      Proto     Age         Pref
```

Next Hop[Interface Name]	Metric
192.0.2.1/32	Remote ISIS 00h02m25s 15
192.0.2.1 (tunneled:RSVP:2)	16777215

No. of Routes: 1
 Flags: n = Number of times nexthop is repeated
 B = BGP backup route available
 L = LFA nexthop available
 S = Sticky ECMP requested

The route tunneled through RSVP has a metric of 16777215, so it is not used by PE-6 GRT to reach any other routes because the metric is very high. After enabling the forwarding adjacency feature (tunnel links) to use shortcuts in the configuration, PE-1 and PE-6 have a direct connection through the RSVP LSP (as a virtual link). This configuration command must be executed in both routers.

```
[/]
A:admin@PE-1(&6)# configure {
  router "Base" {
    isis 0 {
      advertise-tunnel-link
    }
  }
}
```

When the shortcut is advertised by IS-IS, the route will disappear from the RTM because the metric of the shortcut is greater than the IGP cost.

```
[/]
A:admin@PE-6# show router route-table 192.0.2.1/32
```

Dest Prefix[Flags]	Type	Proto	Age	Pref
Next Hop[Interface Name]			Metric	
192.0.2.1/32	Remote	ISIS	00h01m22s	15
192.168.46.1			30	

No. of Routes: 1
 Flags: n = Number of times nexthop is repeated
 B = BGP backup route available
 L = LFA nexthop available
 S = Sticky ECMP requested

If the LSP is reconfigured to use a metric equal to or smaller than the IGP cost, the router PE-6 will use the RSVP shortcut again. In the example, the LSP is reconfigured with a metric of 30 (e.g. on PE-6):

```
[/]
A:admin@PE-6# configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-6-PE-1-strict" {
        metric 30
      }
    }
  }
}
```

```
}

```

Now the shortcut shows up as the preferred next hop to reach PE-1 from PE-6.

```
[/]
A:admin@PE-6# show router route-table 192.0.2.1/32

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age           Pref
  Next Hop[Interface Name]                Metric
-----
192.0.2.1/32                      Remote ISIS   00h00m53s  15
  192.0.2.1 (tunneled:RSVP:2)          30
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```

As explained earlier, this could be combined with ECMP, so if ECMP is configured to 2, the system shows the two equal cost paths.

```
[/]
A:admin@PE-6# configure {
  router "Base" {
    ecmp 2
  }
}

[/]
A:admin@PE-6# show router route-table 192.0.2.1/32

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age           Pref
  Next Hop[Interface Name]                Metric
-----
192.0.2.1/32                      Remote ISIS   00h00m21s  15
  192.0.2.1 (tunneled:RSVP:2)          30
192.0.2.1/32                      Remote ISIS   00h00m21s  15
  192.168.46.1                          30
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```

The GRT on PE-4 displays the route to reach PE-1 (192.0.2.1/32) with a metric of 20 via PE-2 as next hop. Although PE-6 is announcing the RSVP LSP-PE-6-PE-1 to the other routers, the LSP shortcut is not used by PE-4, because the metric to reach PE-6 (10) plus the metric of the LSP shortcut from PE-6 to PE-1 (metric 30) is greater than 20.

```
[/]
A:admin@PE-4# show router route-table
```

```

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]          Type   Proto   Age           Pref
Metric
-----
192.0.2.1/32                      Remote ISIS   01h06m15s  15
192.168.24.1                       20
192.0.2.2/32                      Remote ISIS   01h06m15s  15
192.168.24.1                       10
192.0.2.3/32                      Remote ISIS   01h06m15s  15
192.168.24.1                       20
192.0.2.4/32                      Local  Local   01h06m16s   0
system                             0
192.0.2.5/32                      Remote ISIS   01h05m50s  15
192.168.45.2                       10
192.0.2.6/32                      Remote ISIS   01h05m33s  15
192.168.46.2                       10
192.168.12.0/30                  Remote ISIS   01h06m15s  15
192.168.24.1                       20
192.168.13.0/30                  Remote ISIS   01h06m15s  15
192.168.24.1                       30
192.168.23.0/30                  Remote ISIS   01h06m15s  15
192.168.24.1                       20
192.168.24.0/30                  Local  Local   01h06m16s   0
int-PE-4-PE-2                     0
192.168.35.0/30                  Remote ISIS   01h05m50s  15
192.168.45.2                       20
192.168.45.0/30                  Local  Local   01h06m16s   0
int-PE-4-PE-5                     0
192.168.46.0/30                  Local  Local   01h06m16s   0
int-PE-4-PE-6                     0
192.168.56.0/30                  Remote ISIS   01h05m33s  15
192.168.45.2                       20
-----
No. of Routes: 14
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====

```

If the metric of the LSP LSP-PE-6-PE-1 is modified to a value between 1 and 9, there is a better metric (less than 20) so that PE-4 will change the next hop via PE-6. First the metric of the LSP is modified to 9:

```

[/]
A:admin@PE-6# configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-6-PE-1-strict" {
        metric 9
      }
    }
  }
}

```

The GRT on PE-4 shows that the next hop to reach PE-1 has changed, from next hop PE-2 to next hop PE-6 (therefore, using the LSP shortcut), and the metric is 19 (10 to reach PE-6 plus metric 9 of the LSP PE-6-PE-1 shortcut):

```

[/]

```

```
A:admin@PE-4# show router route-table 192.0.2.1/32

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age          Pref
  Next Hop[Interface Name]                Metric
-----
192.0.2.1/32                      Remote ISIS   00h00m29s  15
  192.168.46.2                      19
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

Because the metric of the LSP shortcut was modified to a value of 9, the GRT of PE-6 shows that the next hops of several routes have changed and are also using the shortcut LSP PE-6-PE-1 because the metric is better than the regular IS-IS metric. IGP shortcuts will not be used to resolve prefixes downstream of the LSP endpoint when the LSP metric is higher than the underlying IGP cumulative metric.

```
[/]
A:admin@PE-6# show router route-table

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age          Pref
  Next Hop[Interface Name]                Metric
-----
192.0.2.1/32                      Remote ISIS   00h00m54s  15
  192.0.2.1 (tunneled:RSVP:2)          9
192.0.2.2/32                      Remote ISIS   00h00m54s  15
  192.0.2.1 (tunneled:RSVP:2)          19
192.0.2.3/32                      Remote ISIS   00h00m54s  15
  192.0.2.1 (tunneled:RSVP:2)          19
192.0.2.4/32                      Remote ISIS   00h02m06s  15
  192.168.46.1                      10
192.0.2.5/32                      Remote ISIS   00h02m06s  15
  192.168.56.1                      10
192.0.2.6/32                      Local  Local    01h08m12s  0
  system                             0
192.168.12.0/30                   Remote ISIS   00h00m54s  15
  192.0.2.1 (tunneled:RSVP:2)          19
192.168.13.0/30                   Remote ISIS   00h00m54s  15
  192.0.2.1 (tunneled:RSVP:2)          19
192.168.23.0/30                   Remote ISIS   00h00m54s  15
  192.0.2.1 (tunneled:RSVP:2)          29
192.168.24.0/30                   Remote ISIS   00h02m06s  15
  192.168.46.1                      20
192.168.35.0/30                   Remote ISIS   00h02m06s  15
  192.168.56.1                      20
192.168.45.0/30                   Remote ISIS   00h02m06s  15
  192.168.46.1                      20
192.168.46.0/30                   Local  Local    01h08m12s  0
  int-PE-6-PE-4                     0
192.168.56.0/30                   Local  Local    01h08m12s  0
  int-PE-6-PE-5                     0
-----
No. of Routes: 14
Flags: n = Number of times nexthop is repeated
```

B = BGP backup route available
L = LFA nexthop available
S = Sticky ECMP requested

=====

There are also cases where an LDP FEC can resolve to an RSVP LSP, if the user enables the LDP-over-RSVP feature or IGP shortcut feature when **prefer-tunnel-in-tunnel** is set to true in LDP and the endpoint of the RSVP LSP matches the FEC prefix. For those cases, the metric to the prefix is the sum of the RSVP LSP metric and the remaining IGP path cost.

[Table 13: RSVP LSP Role As Outcome of LSP Level and IGP Level Configuration Options](#) provides a summary of the outcome when configuring the forwarding adjacency, LDPoRSVP and RSVP shortcut options at both the IGP instance level and at the LSP level.

Table 13: RSVP LSP Role As Outcome of LSP Level and IGP Level Configuration Options

	IGP Instance Level Configurations					
LSP Level Configuration	advertise-tunnel-link enabled/ rsvp-shortcut enabled/ldp-over-rsvp enabled	advertise-tunnel-link enabled/ rsvp-shortcut enabled/ldp-over-rsvp disabled	advertise-tunnel-link enabled/ rsvp-shortcut disabled/ldp-over-rsvp disabled	advertise-tunnel-link disabled/ rsvp-shortcut disabled/ldp-over-rsvp disabled	advertise-tunnel-link disabled/ rsvp-shortcut enabled/ldp-over-rsvp enabled	advertise-tunnel-link disabled/ rsvp-shortcut disabled/ldp-over-rsvp enabled
igp-shortcut enabled/ldp-over-rsvp enabled	Forwarding Adjacency	Forwarding Adjacency	Forwarding Adjacency	None	IGP Shortcut	LDP-over-RSVP
igp-shortcut enabled/ldp-over-rsvp disabled	Forwarding Adjacency	Forwarding Adjacency	Forwarding Adjacency	None	IGP Shortcut	None
igp-shortcut disabled/ldp-over-rsvp enabled	None	None	None	None	None	LDP-over-RSVP
igp-shortcut disabled/ldp-over-rsvp disabled	None	None	None	None	None	None

LSP Relative Metric

It is possible to use relative metrics for IGP shortcuts as per RFC 3906, *Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels*, with the following command (e.g. on PE-6):

```
[/]
A:admin@PE-6# configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-6-PE-1-strict" {
        igp-shortcut ?
```

```

admin-state      - Administrative state of the LSP to be used as a shortcut while
resolving IGP routes

Choice: igp-choice
lfa-type         :- Specific RSVP LSP by IS-IS and OSPF routing protocols to assign as
shortcut or as forwarding adjacency
relative-metric  :- Shortest IGP cost between the endpoints of the LSP plus the configured
offset

[/]
A:admin@PE-6# configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-6-PE-1-strict" {
        igp-shortcut {
          lfa-type ?

lfa-type <keyword>
<keyword> - (lfa-protect|lfa-only)

        Specific RSVP LSP by IS-IS and OSPF routing protocols to assign as shortcut or as
forwarding adjacency

[/]
A:admin@PE-6# configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-6-PE-1-strict" {
        igp-shortcut {
          relative-metric ?

relative-metric <number>
<number> - <-10..10>

        Shortest IGP cost between the endpoints of the LSP plus the configured offset

```

When this feature is enabled, IGP applies the shortest IGP cost between the endpoints of the LSP, plus the value of a configured offset when computing the cost of the prefix that is resolved to the LSP.

The offset value is optional and can have a value between -10 and 10, and defaults to zero (0). An offset value of zero (0) is used when the relative metric option is enabled without specifying the offset parameter value. The minimum net cost for the prefix is capped to the value of one (1) after applying the offset:

$$\text{Prefix cost} = \max(1, \text{IGP Cost} + \text{relative metric offset})$$

The **relative-metric** option is ignored when **advertise-tunnel-link** is enabled in IS-IS or OSPF. In that case, the IGP advertises the LSP as a P2P unnumbered link using the LSP operational metric.

The **relative-metric** option is mutually exclusive with the **lfa-protect** (Loop-Free Alternate (LFA)) or the **lfa-only** options. An LSP with **relative-metric** option enabled cannot be included in the LFA SPF and vice versa when RSVP shortcuts are enabled in the IGP (see chapter [LDP/IP FRR LFA for IGP Shortcut Using IS-IS/OSPF](#) for more information).

The offset can be used to enforce the preference of the shortcut path over the other paths for the prefix. Using an example, a new CSPF LSP with empty path and relative metric of -10 is created between PE-6 and PE-1. Whereas the operational or absolute metric is 30 (IGP cost and populated in the Tunnel Table Manager, TTM), the metric that the RTM shows is 20 after applying the offset:

```

[/]
A:admin@PE-6# configure {
  router "Base" {

```



```

mpls {
  admin-state enable
  lsp "LSP-PE-6-PE-1-loose" {
    to 192.0.2.1
    type p2p-rsvp
    primary "loose_path" { }
    path-computation-method local-cspf
    igp-shortcut {
      relative-metric -10
    }
  }
}

```

```

[/]
A:admin@PE-6# show router tunnel-table 192.0.2.1

```

=====
IPv4 Tunnel Table (Router: Base)
=====

Destination Color	Owner	Encap	TunnelId	Pref	Nexthop	Metric
192.0.2.1/32	rsvp	MPLS	1	7	192.168.46.1	30
192.0.2.1/32	rsvp	MPLS	2	7	192.168.56.1	16777215
192.0.2.1/32	ldp	MPLS	65546	9	192.168.46.1	30

Flags: B = BGP or MPLS backup hop available
L = Loop-Free Alternate (LFA) hop available
E = Inactive best-external BGP route
k = RIB-API or Forwarding Policy backup hop
=====

```

[/]
A:admin@PE-6# show router route-table 192.0.2.1

```

=====
Route Table (Router: Base)
=====

Dest Prefix[Flags] Next Hop[Interface Name]	Type	Proto	Age	Pref Metric
192.0.2.1/32 192.168.46.1	Remote	ISIS	00h03m11s	15 20

No. of Routes: 1
Flags: n = Number of times nexthop is repeated
B = BGP backup route available
L = LFA nexthop available
S = Sticky ECMP requested
=====

LDP/IP FRR LFA for IGP Shortcut Using IS-IS/OSPF

MPLS LDP/IP FRR LFA for IGP shortcuts allows for the use of RSVP LSP-based IGP shortcuts as Loop-Free Alternate (LFA) backups, this way expanding the coverage of the IP Fast-Reroute (FRR) and the LDP FRR capabilities for IS-IS and OSPF prefixes. For a detailed description about IP and LDP FRR, see chapter [MPLS LDP FRR using ISIS as IGP](#).

When an RSVP LSP is used as a shortcut by IS-IS or OSPF, it is included by the SPF as a P2P link and it can also be optionally advertised into the rest of the network by the IGP.

Two LSP-level configuration options are provided:

- The **lfa-protect** option includes the RSVP LSP in both the main SPF and the LFA SPFs. If the prefix primary next hop (NH) is tunneled, no LFA NH is computed. The protection in this case is provided by RSVP FRR. If the prefix primary NH is direct, then an LFA NH is computed. A direct LFA NH is preferred over a tunneled LFA NH. Within each LFA NH type, node protection is preferred over link protection. The configuration command is:

```
[/]
A:admin@a PE># configure {
  router "Base" {
    mpls {
      lsp "<LSP Name>" {
        igp-shortcut {
          lfa-type lfa-protect
        }
      }
    }
  }
}
```

- The **lfa-only** option includes the LSP in the LFA SPFs only so that the introduction of IGP shortcuts does not impact the main SPF decision. The prefix primary NH is always direct and the prefix LFA NH is computed. A direct LFA NH is preferred over a tunneled LFA NH. Within each LFA NH type, node protection is preferred over link protection. The configuration command is:

```
[/]
A:admin@a PE># configure {
  router "Base" {
    mpls {
      lsp "<LSP Name>" {
        igp-shortcut {
          lfa-type lfa-only
        }
      }
    }
  }
}
```

LDP/IP FRR is a local decision, so it can be enabled per node and there are no interoperability issues with other nodes. In the topology, PE-2 is provisioned with IS-IS LFA (OSPF configuration for the rest of this section is similar):

```
[/]
A:admin@PE-2# configure {
  router "Base" {
    isis 0 {
      loopfree-alternate { }
    }
  }
}
```

The second item to configure is whether LDP or IP FRR is provisioned. To configure IP FRR, the command is:

```
[/]
```

```
A:admin@PE-2# configure {
  routing-options {
    ip-fast-reroute true
  }
}
```

To configure LDP FRR, the following command is used:

```
[/]
A:admin@PE-2# configure {
  router "Base" {
    ldp {
      fast-reroute
    }
  }
}
```

Although not shown, it is recommended to enable IGP-LDP synchronization per interface to avoid possible traffic black-holes.

LFA is enabled in all routers of the topology. The following command shows the LFA coverage on PE-2 where four nodes out of five are protected (80%) and seven of the ten prefixes are protected (70%). IPv4 prefixes are protected (IPv6 is not configured). The following output shows L1 and L2 because this node is provisioned as an L1-L2 IS-IS router.

```
[/]
A:admin@PE-2# show router isis lfa-coverage

=====
Rtr Base ISIS Instance 0 LFA Coverage
=====
Topology          Level  Node      IPv4          IPv6
-----
IPV4 Unicast      L1     4/5(80%)   7/10(70%)    0/0(0%)
IPV6 Unicast      L1     0/0(0%)    0/0(0%)      0/0(0%)
IPV4 Multicast    L1     0/0(0%)    0/0(0%)      0/0(0%)
IPV6 Multicast    L1     0/0(0%)    0/0(0%)      0/0(0%)
IPV4 Unicast      L2     4/5(80%)   7/10(70%)    0/0(0%)
IPV6 Unicast      L2     0/0(0%)    0/0(0%)      0/0(0%)
IPV4 Multicast    L2     0/0(0%)    0/0(0%)      0/0(0%)
IPV6 Multicast    L2     0/0(0%)    0/0(0%)      0/0(0%)
=====
```

PE-2, PE-3, PE-4, and PE-5 share the same results, whereas only PE-1 and PE-6 have a coverage of 100% as shown in the following output.

```
[/]
A:admin@PE-1# show router isis lfa-coverage

=====
Rtr Base ISIS Instance 0 LFA Coverage
=====
Topology          Level  Node      IPv4          IPv6
-----
IPV4 Unicast      L1     5/5(100%)  11/11(100%)  0/0(0%)
IPV6 Unicast      L1     0/0(0%)    0/0(0%)      0/0(0%)
IPV4 Multicast    L1     0/0(0%)    0/0(0%)      0/0(0%)
IPV6 Multicast    L1     0/0(0%)    0/0(0%)      0/0(0%)
IPV4 Unicast      L2     5/5(100%)  11/11(100%)  0/0(0%)
IPV6 Unicast      L2     0/0(0%)    0/0(0%)      0/0(0%)
IPV4 Multicast    L2     0/0(0%)    0/0(0%)      0/0(0%)
=====
```

```
IPV6 Multicast L2 0/0(0%) 0/0(0%) 0/0(0%)
=====
```

Taking a deeper look into the IS-IS LFA on PE-2, it can be seen that the node which is not protected is PE-4 (system address 192.0.2.4, because it is the one missing):

```
[/]
A:admin@PE-2# show router route-table alternative | match LFA pre-lines 2
192.0.2.1/32 Remote ISIS 01h21m18s 15
    192.168.12.1 10
    192.168.23.2 (LFA) 20
192.0.2.3/32 Remote ISIS 01h20m52s 15
    192.168.23.2 10
    192.168.12.1 (LFA) 20
192.0.2.5/32 Remote ISIS 01h20m18s 15
    192.168.23.2 20
    192.168.24.2 (LFA) 20
192.0.2.6/32 Remote ISIS 01h19m56s 15
    192.168.24.2 20
    192.168.23.2 (LFA) 30
192.168.13.0/30 Remote ISIS 01h21m18s 15
    192.168.12.1 20
    192.168.23.2 (LFA) 30
192.168.35.0/30 Remote ISIS 01h20m52s 15
    192.168.23.2 20
    192.168.12.1 (LFA) 30
192.168.56.0/30 Remote ISIS 01h20m18s 15
    192.168.23.2 30
    192.168.24.2 (LFA) 30
Flags: n = Number of times nexthop is repeated
Backup = BGP backup route
LFA = Loop-Free Alternate nexthop
```

LFA is improved by taking advantage of RSVP shortcuts when it is properly provisioned. The reason why PE-4 cannot be protected with an LFA path is because the direct NH is using the direct link between PE-2 and PE-4 (the shortest IGP) and the intended LFA path through PE-3 is not valid (when LFA tries to find an alternate path via PE-3, the IGP cost from PE-3 to PE-4 is the same going via PE-5 as the path back via PE-2, invalidating that LFA calculation because there is a loop). This is normal because PE-2, PE-3, PE-4 and PE-5 are forming a ring. LFA coverage is increased by adding a link between PE-2 and PE-5, which can be done using a physical link or a virtual link with an RSVP shortcut. From the two possible options (**lfa-only** and **lfa-protect**), a new LSP "LSP-PE-2-PE-5" is configured with **igp-shortcut lfa-only**.

```
[/]
A:admin@PE-2# configure {
  router "Base" {
    mpls {
      admin-state enable
      path "path-to-PE-5" {
        admin-state enable
        hop 10 {
          ip-address 192.0.2.3
          type strict
        }
        hop 20 {
          ip-address 192.0.2.5
          type strict
        }
      }
    }
    lsp "LSP-PE-2-PE-5" {
      admin-state enable
      to 192.0.2.5
    }
  }
}
```



```

192.168.23.2 10
192.168.12.1 (LFA) 20
192.0.2.4/32 Remote ISIS 01h22m47s 15
192.168.24.2 10
192.0.2.5 (LFA) (tunneled:RSVP:1) 16777225
192.0.2.5/32 Remote ISIS 01h22m27s 15
192.168.23.2 20
192.168.24.2 (LFA) 20
192.0.2.6/32 Remote ISIS 01h22m05s 15
192.168.24.2 20
192.168.23.2 (LFA) 30
192.168.13.0/30 Remote ISIS 01h23m27s 15
192.168.12.1 20
192.168.23.2 (LFA) 30
192.168.35.0/30 Remote ISIS 01h23m01s 15
192.168.23.2 20
192.168.12.1 (LFA) 30
192.168.45.0/30 Remote ISIS 01h22m47s 15
192.168.24.2 20
192.0.2.5 (LFA) (tunneled:RSVP:1) 16777235
192.168.46.0/30 Remote ISIS 01h22m47s 15
192.168.24.2 20
192.0.2.5 (LFA) (tunneled:RSVP:1) 16777235
192.168.56.0/30 Remote ISIS 01h22m27s 15
192.168.23.2 30
192.168.24.2 (LFA) 30
Flags: n = Number of times nexthop is repeated
Backup = BGP backup route
LFA = Loop-Free Alternate nexthop

```

The tunnel table shows the RSVP LSP used as a shortcut and its operational metric.

```

[/]
A:admin@PE-2# show router tunnel-table 192.0.2.5

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.5/32      rsvp      MPLS 1          7    192.168.23.2 16777215
192.0.2.5/32 [L]  ldp      MPLS 65540        9    192.168.23.2 20
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====

```

If the LSP "LSP-PE-2-PE-5" is provisioned with **lfa-protect** instead of **lfa-only**, the result is that the LSP "LSP-PE-2-PE-5" is used by normal SPF to define the primary NH and it is not used by LFA SPF anymore.

```

[/]
A:admin@PE-2# configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-2-PE-5" {
        igp-shortcut {
          lfa-type lfa-protect
        }
      }
    }
  }
}

```

```
)
)
}
```

The coverage when lfa-protect is used also shows a 100% for nodes and 100% for prefixes, as follows.

```
[/]
A:admin@PE-2# show router isis lfa-coverage

=====
Rtr Base ISIS Instance 0 LFA Coverage
=====
Topology          Level   Node          IPv4          IPv6
-----
IPV4 Unicast      L1      5/5(100%)     9/9(100%)     0/0(0%)
IPV6 Unicast      L1      0/0(0%)       0/0(0%)       0/0(0%)
IPV4 Multicast    L1      0/0(0%)       0/0(0%)       0/0(0%)
IPV6 Multicast    L1      0/0(0%)       0/0(0%)       0/0(0%)
IPV4 Unicast      L2      5/5(100%)     9/9(100%)     0/0(0%)
IPV6 Unicast      L2      0/0(0%)       0/0(0%)       0/0(0%)
IPV4 Multicast    L2      0/0(0%)       0/0(0%)       0/0(0%)
IPV6 Multicast    L2      0/0(0%)       0/0(0%)       0/0(0%)
=====
```

In this case, the GRT looks as follows, the main difference being that now PE-5 (192.0.2.5) has a direct shortcut from PE-2:

```
[/]
A:admin@PE-2# show router route-table alternative

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type   Proto   Age           Pref
  Next Hop[Interface Name]  Metric
  Alt-NextHop               Alt-
                             Metric
-----
192.0.2.1/32                 Remote ISIS    01h24m50s  15
  192.168.12.1                10
  192.168.23.2 (LFA)          20
192.0.2.2/32                 Local  Local    01h24m51s   0
  system                       0
192.0.2.3/32                 Remote ISIS    01h24m24s  15
  192.168.23.2                10
  192.168.12.1 (LFA)          20
192.0.2.4/32                 Remote ISIS    01h24m10s  15
  192.168.24.2                10
  192.0.2.5 (LFA) (tunneled:RSVP:1) 16777225
192.0.2.5/32                 Remote ISIS 00h00m45s 15
  192.0.2.5 (tunneled:RSVP:1) 16777215
192.0.2.6/32                 Remote ISIS    01h23m28s  15
  192.168.24.2                20
  192.168.23.2 (LFA)          30
192.168.12.0/30              Local  Local    01h24m51s   0
  int-PE-2-PE-1                0
192.168.13.0/30              Remote ISIS    01h24m50s  15
  192.168.12.1                20
  192.168.23.2 (LFA)          30
192.168.23.0/30              Local  Local    01h24m51s   0
  int-PE-2-PE-3                0
192.168.24.0/30              Local  Local    01h24m51s   0
```

```

int-PE-2-PE-4
192.168.35.0/30 Remote ISIS 01h24m24s 15
192.168.23.2 Remote ISIS 20
192.168.12.1 (LFA) Remote ISIS 30
192.168.45.0/30 Remote ISIS 01h24m10s 15
192.168.24.2 Remote ISIS 20
192.0.2.5 (LFA) (tunneled:RSVP:1) Remote ISIS 16777235
192.168.46.0/30 Remote ISIS 01h24m10s 15
192.168.24.2 Remote ISIS 20
192.0.2.5 (LFA) (tunneled:RSVP:1) Remote ISIS 16777235
192.168.56.0/30 Remote ISIS 00h00m45s 15
192.168.24.2 Remote ISIS 30
192.168.23.2 (LFA) Remote ISIS 40
-----
No. of Routes: 14
Flags: n = Number of times nexthop is repeated
Backup = BGP backup route
LFA = Loop-Free Alternate nexthop
S = Sticky ECMP requested
=====

```

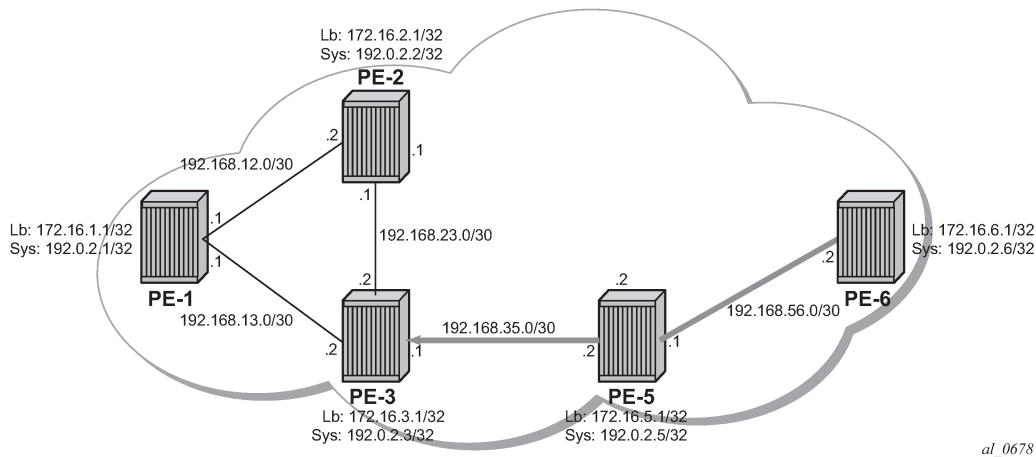
Rules Determining the Installation of Shortcuts into the RTM

Although it was already mentioned in the RSVP-TE LSP shortcut for IGP route resolution section, the rules determining how shortcuts are installed into the RTM are (sorted by higher priority):

- RSVP shortcut.
- LDP shortcut.
- IGP route with regular IP next hop.
- The implementation is compliant with RFC 3906.

To check the rules, the network configuration is iLDP in all interfaces with LDP shortcuts enabled, there is also an RSVP LSP from PE-6 to PE-3 available but RSVP shortcuts are disabled. The topology is shown in [Figure 210: Network Topology to Verify Installation of Shortcuts into the RTM](#).

Figure 210: Network Topology to Verify Installation of Shortcuts into the RTM



The following RSVP LSP is needed between PE-6 and PE-3.

```
[/]
A:admin@PE-6# configure {
  router "Base" {
    ldp {
      ldp-shortcut {
        ipv4 true
      }
    }
  }
}
```

```
[/]
A:admin@PE-6# configure {
  router "Base" {
    isis 0 {
      delete loopfree-alternate
      igp-shortcut {
        tunnel-next-hop {
          family ipv4 {
            resolution-filter {
              rsvp false
            }
            resolution none
          }
        }
      }
      admin-state disable
    }
  }
}
```

```
[/]
A:admin@PE-6# configure {
  router "Base" {
    mpls {
      path "loose_path" {
        admin-state enable
      }
      lsp "LSP-PE-6-PE-3" {
        admin-state enable
        to 192.0.2.3
        type p2p-rsvp
        primary "loose_path" { }
        path-computation-method local-cspf
      }
    }
  }
}
```

The routes in the routing table on PE-6 are the following:

```
[/]
A:admin@PE-6# show router route-table

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                                Type  Proto  Age      Pref
Next Hop[Interface Name]                          Metric

=====
```

```
-----
192.0.2.1/32                               Remote LDP      00h00m00s 9
    192.168.56.1 (tunneled)                30
192.0.2.2/32                               Remote LDP      00h00m00s 9
    192.168.56.1 (tunneled)                30
192.0.2.3/32                               Remote LDP      00h00m00s 9
    192.168.56.1 (tunneled)                20
192.0.2.5/32                               Remote LDP      00h00m00s 9
    192.168.56.1 (tunneled)                10
192.0.2.6/32                               Local  Local      01h28m21s 0
    system                                  0
192.168.12.0/30                            Remote ISIS     00h02m08s 15
    192.168.56.1                            40
192.168.13.0/30                            Remote ISIS     00h19m24s 15
    192.168.56.1                            30
192.168.23.0/30                            Remote ISIS     00h02m08s 15
    192.168.56.1                            30
192.168.35.0/30                            Remote ISIS     00h22m14s 15
    192.168.56.1                            20
192.168.56.0/30                            Local  Local      01h28m21s 0
    int-PE-6-PE-5                            0
-----
No. of Routes: 10
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

The tunnel table shows the LSPs available for the shortcuts, and therefore, these are used in the GRT for LDP (but not for RSVP):

```
[/]
A:admin@PE-6# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.1/32     ldp       MPLS  65547    9    192.168.56.1  30
192.0.2.2/32     ldp       MPLS  65544    9    192.168.56.1  30
192.0.2.3/32     rsvp      MPLS   4        7    192.168.56.1  20
192.0.2.3/32     ldp       MPLS  65545    9    192.168.56.1  20
192.0.2.5/32     ldp       MPLS  65540    9    192.168.56.1  10
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
```

So far, LDP shortcuts are preferred over the IGP next hops for the system addresses (router ID). After enabling RSVP shortcuts in the IS-IS context, the changes in the GRT are:

```
[/]
A:admin@PE-6# configure {
  router "Base" {
    isis 0 {
      igp-shortcut {
        admin-state enable
      }
    }
  }
}
```



```

=====
Dest Prefix[Flags]                                Type  Proto  Age      Pref
  Next Hop[Interface Name]                        Metric
-----
192.0.2.3/32                                     Remote ISIS  00h00m53s 15
  192.0.2.3 (tunneled:RSVP:4)                    100
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

2. SPF also picks the RSVP LSP shortcut if both the LSP path and the IGP path to the prefix are via the tail-end of the LSP. This is regardless of the path cost compared to the IGP next hop. When paths over multiple RSVP shortcuts have the same lowest cost, if ECMP is enabled on the system, the LSP with the lowest tunnel ID is chosen. In this example, 192.168.13.0 and 192.168.23.0 are using the shortcut but 192.168.12.0 is not.

```

[/]
A:admin@PE-6# show router route-table

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                                Type  Proto  Age      Pref
  Next Hop[Interface Name]                        Metric
-----
192.0.2.1/32                                     Remote LDP    00h00m53s 9
  192.168.56.1 (tunneled)                        30
192.0.2.2/32                                     Remote LDP    00h00m53s 9
  192.168.56.1 (tunneled)                        30
192.0.2.3/32                                     Remote ISIS  00h00m53s 15
  192.0.2.3 (tunneled:RSVP:4)                    100
192.0.2.5/32                                     Remote LDP    00h01m25s 9
  192.168.56.1 (tunneled)                        10
192.0.2.6/32                                     Local  Local    01h29m45s 0
  system
192.168.12.0/30                                Remote ISIS  00h00m53s 15
  192.168.56.1                                40
192.168.13.0/30                                Remote ISIS  00h00m53s 15
  192.0.2.3 (tunneled:RSVP:4)                  110
192.168.23.0/30                                Remote ISIS  00h00m53s 15
  192.0.2.3 (tunneled:RSVP:4)                  110
192.168.35.0/30                                  Remote ISIS  00h23m39s 15
  192.168.56.1                                  20
192.168.56.0/30                                  Local  Local    01h29m45s 0
  int-PE-6-PE-5                                  0
-----
No. of Routes: 10
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

LDP/RSVP LSP Shortcut for BGP NH Resolution

Using LDP/RSVP LSP shortcuts for resolving BGP next hops allows IPv4 packet forwarding to routes resolved via a BGP next hop using an LDP/RSVP LSP instead of using a regular IP next hop. In the network topology of [Figure 207: Example Topology](#), both PE-3 and PE-6 have a single peer configured, initially without any shortcuts enabled under the **bgp** context. Also, one static route is configured in PE-3 and PE-6 and that is redistributed into BGP. The relevant configuration on PE-3 is the following:

```
[/]
A:admin@PE-3# configure {
  router "Base" {
    interface "int-static-route" {
      ipv4 {
        primary {
          address 172.16.33.1
          prefix-length 30
        }
      }
      port 1/1/4:33
    }
    autonomous-system 65536
    static-routes {
      route 10.10.10.0/24 route-type unicast {
        next-hop "172.16.33.2" {
          admin-state enable
        }
      }
    }
  }
}

[/]
A:admin@PE-3# configure {
  policy-options {
    policy-statement "static-routes-policy" {
      description "export static-routes for I-BGP"
      entry 10 {
        from {
          protocol {
            name [static]
          }
        }
        to {
          protocol {
            name [bgp]
          }
        }
        action {
          action-type accept
          next-hop self
        }
      }
      default-action {
        action-type reject
      }
    }
  }
}

[/]
A:admin@PE-3# configure {
  router "Base" {
    bgp {
```

```

    export {
      policy ["static-routes-policy"]
    }
    group "ibgp" {
      peer-as 65536
    }
    neighbor 192.0.2.6 {
      group "ibgp"
    }
  }
}

```

Checking the static route received on PE-6 via BGP, the next hop is the PE-3 system address:

```

[/]
A:admin@PE-6# show router bgp routes 10.10.10.0/24 detail
=====
BGP Router ID:192.0.2.6      AS:65536      Local AS:65536
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Original Attributes

Network       : 10.10.10.0/24
Nexthop      : 192.0.2.3
Path Id      : None
From         : 192.0.2.3
Res. Protocol : ISIS                      Res. Metric   : 20
Res. Nexthop : 192.168.56.1
Local Pref.  : 100                        Interface Name : int-PE-6-PE-5
Aggregator AS : None                      Aggregator    : None
Atomic Aggr. : Not Atomic                  MED           : None
AIGP Metric  : None                       IGP Cost      : 20
Connector    : None
Community    : No Community Members
Cluster      : No Cluster Members
Originator Id : None                       Peer Router Id : 192.0.2.3
Fwd Class    : None                       Priority       : None
Flags        : Used Valid Best Incomplete In-RTM
Route Source : Internal
AS-Path      : No As-Path
Route Tag    : 0
Neighbor-AS  : n/a
Orig Validation: NotFound
Source Class : 0                          Dest Class    : 0
Add Paths Send : Default
RIB Priority  : Normal
Last Modified : 00h01m32s

Modified Attributes

---snip---

-----
Routes : 1
=====

```

Three of the BGP peering configuration possibilities are LDP, RSVP, or BGP. The other resolution filter options are related to segment routing and are beyond the scope of this chapter. In case both LDP and RSVP are included in the filter, RSVP is preferred. Disabling the IGP is also allowed (meaning that unless there is a shortcut, the BGP peering will not fall back to IGP):

```
[/]
A:admin@PE-6# configure {
  router "Base" {
    bgp {
      next-hop-resolution {
        shortcut-tunnel {
          family ipv4 {
            resolution ?

resolution <keyword>
<keyword> - (none|filter|any)
Default   - none

Resolution mode for binding BGP routes to tunnel types

[/]
A:admin@PE-6# configure {
  router "Base" {
    bgp {
      next-hop-resolution {
        shortcut-tunnel {
          family ipv4 {
            resolution-filter ?

resolution-filter

bgp          - Use BGP tunneling for next-hop resolution
ldp          - Use LDP tunneling for next-hop resolution
mpls-fwd-policy - Use MPLS forwarding policy for next-hop resolution
rib-api      - Use RIB API gRPC service for next-hop resolution
rsvp        - Use RSVP tunneling for next-hop resolution
sr-isis     - Use IS-IS SR tunneling for next-hop resolution
sr-ospf     - Use OSPF SR tunneling for next-hop resolution
sr-ospfv3   - Use OSPFv3 SR tunneling for next-hop resolution
sr-policy   - Use SR policies for next-hop resolution
sr-te       - Use SR-TE tunneling for next-hop resolution
```

When enabling LDP shortcuts on PE-6, the output changes showing the detail of the received BGP route indicating that the next hop is resolved using LDP:

```
configure {
  router "Base" {
    bgp {
      next-hop-resolution {
        shortcut-tunnel {
          family ipv4 {
            resolution-filter {
              ldp true
            }
            resolution filter
          }
        }
      }
    }
  }
}
```

```

}

[/]
A:admin@PE-6# show router bgp routes 10.10.10.0/24 detail
=====
BGP Router ID:192.0.2.6      AS:65536      Local AS:65536
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Original Attributes

Network       : 10.10.10.0/24
Nexthop       : 192.0.2.3
Path Id       : None
From          : 192.0.2.3
Res. Protocol : LDP                      Res. Metric   : 20
Res. Nexthop  : 192.0.2.3 (LDP)
Local Pref.   : 100                      Interface Name : NotAvailable
Aggregator AS : None                      Aggregator    : None
Atomic Aggr.  : Not Atomic                MED           : None
AIGP Metric   : None                      IGP Cost      : 20
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None                      Peer Router Id : 192.0.2.3
Fwd Class     : None                      Priority       : None
Flags         : Used Valid Best Incomplete In-RTM
Route Source  : Internal
AS-Path       : No As-Path
Route Tag     : 0
Neighbor-AS   : n/a
Orig Validation: NotFound
Source Class  : 0                          Dest Class    : 0
Add Paths Send : Default
RIB Priority   : Normal
Last Modified : 00h02m32s

Modified Attributes

---snip---

-----
Routes : 1
=====

```

The GRT output command also shows that the route is reachable using LDP (indicated as tunneled):

```

[/]
A:admin@PE-6# show router route-table next-hop-type tunneled
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age      Pref
  Next Hop[Interface Name]                Metric
-----
10.10.10.0/24              Remote BGP    00h00m30s 170

```



```

192.0.2.3 (tunneled)                                20
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

The previously created LSP LSP-PE-6-PE-3 is up and running:

```

[/]
A:admin@PE-6# show router mpls lsp "LSP-PE-6-PE-3" path detail

=====
MPLS LSP LSP-PE-6-PE-3 Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use
  b - Bandwidth Protected      n - Node Protected
  s - Soft Preemption
  S - Strict                    L - Loose
  A - ABR                      + - Inherited
=====

LSP LSP-PE-6-PE-3
Path loose_path
-----
LSP Name      : LSP-PE-6-PE-3
From          : 192.0.2.6
To            : 192.0.2.3
Admin State   : Up                Oper State    : Up
Path Name     : loose_path
Path LSP ID   : 9216              Path Type     : Primary
Path Admin    : Up                Path Oper     : Up
Out Interface : 1/1/1             Out Label     : 524283
Path Up Time  : 0d 00:13:27      Path Down Time : 0d 00:00:00
Retry Limit   : 0                 Retry Timer    : 30 sec
Retry Attempt : 0                 Next Retry In  : 0 sec

---snip---

Adspec        : Disabled          Oper Adspec    : Disabled
PathCompMethod : local-cspf       OperPathCompMethod: local-cspf
MetricType     : igp              Oper MetricType : igp
Least Fill     : Disabled         Oper LeastFill  : Disabled
FRR            : Disabled         Oper FRR        : Disabled
Propagate Adm Grp: Disabled       Oper Prop Adm Grp : Disabled
Inter-area     : False

---snip---

Adaptive      : Enabled           Oper Metric    : 100
Preference    : n/a
Path Trans    : 1                 CSPF Queries   : 1
Failure Code   : noError
Failure Node  : n/a
Explicit Hops  :
  No Hops Specified
Actual Hops    :
  192.168.56.2(192.0.2.6)         Record Label   : N/A
  -> 192.168.56.1(192.0.2.5)     Record Label   : 524283
  -> 192.168.35.1(192.0.2.3)    Record Label   : 524284

```

```

Computed Hops      :
  192.168.56.2(S)
-> 192.168.56.1(S)
-> 192.168.35.1(S)
Resignal Eligible: False
Last Resignal     : n/a                CSPF Metric      : 20
=====

```

After adding **resolution-filter rsvp** to the shortcut-tunnel configuration in the **bgp** context, the output shows that the BGP peer is reachable using an RSVP LSP (switched from LDP to RSVP because RSVP is preferred):

```

[/]
A:admin@PE-6# configure {
  router "Base" {
    bgp {
      next-hop-resolution {
        shortcut-tunnel {
          family ipv4 {
            resolution-filter {
              ldp true
              rsvp true
            }
            resolution filter
          }
        }
      }
    }
  }
}

[/]
A:admin@PE-6# show router bgp routes 10.10.10.0/24 detail
=====
BGP Router ID:192.0.2.6      AS:65536      Local AS:65536
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Original Attributes

Network       : 10.10.10.0/24
Nexthop      : 192.0.2.3
Path Id      : None
From         : 192.0.2.3
Res. Protocol : RSVP                      Res. Metric   : 100
Res. Nexthop  : 192.0.2.3 (RSVP LSP: 4)
Local Pref.   : 100                       Interface Name : NotAvailable
Aggregator AS : None                       Aggregator    : None
Atomic Aggr.  : Not Atomic                 MED           : None
AIGP Metric   : None                       IGP Cost      : 100
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None                       Peer Router Id : 192.0.2.3
Fwd Class     : None                       Priority       : None
Flags         : Used Valid Best Incomplete In-RTM
Route Source  : Internal

```

```
AS-Path      : No As-Path
Route Tag    : 0
Neighbor-AS  : n/a
Orig Validation: NotFound
Source Class : 0                      Dest Class : 0
Add Paths Send : Default
RIB Priority  : Normal
Last Modified : 00h03m59s
```

Modified Attributes

---snip---

Routes : 1
=====

The GRT output command also shows that the route is reachable using RSVP (indicated as tunneled:RSVP:4):

```
[/]
A:admin@PE-6# show router route-table next-hop-type tunneled

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age           Pref
  Next Hop[Interface Name]                Metric
-----
10.10.10.0/24                      Remote BGP    00h00m33s  170
  192.0.2.3 (tunneled:RSVP:4)                100
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

If the RSVP LSP is **disabled**, the system reverts back to the LDP LSP:

```
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-6-PE-3" {
        admin-state disable
      }
    }
  }
}
```

```
[/]
A:admin@PE-6# show router bgp routes 10.10.10.0/24 detail

=====
BGP Router ID:192.0.2.6      AS:65536      Local AS:65536
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
```

```

=====
BGP IPv4 Routes
=====
Original Attributes

Network       : 10.10.10.0/24
Nexthop      : 192.0.2.3
Path Id      : None
From         : 192.0.2.3
Res. Protocol : LDP                      Res. Metric   : 20
Res. Nexthop : 192.0.2.3 (LDP)
Local Pref.  : 100                      Interface Name : NotAvailable
Aggregator AS : None                   Aggregator   : None
Atomic Aggr. : Not Atomic              MED          : None
AIGP Metric  : None                    IGP Cost     : 20
Connector    : None
Community    : No Community Members
Cluster      : No Cluster Members
Originator Id : None                   Peer Router Id : 192.0.2.3
Fwd Class    : None                    Priority      : None
Flags        : Used Valid Best Incomplete In-RTM
Route Source : Internal
AS-Path      : No As-Path
Route Tag    : 0
Neighbor-AS  : n/a
Orig Validation: NotFound
Source Class : 0                       Dest Class    : 0
Add Paths Send : Default
RIB Priority  : Normal
Last Modified : 00h04m21s

Modified Attributes

---snip---

-----
Routes : 1
=====

```

When the shortcut tunnel with **resolution-filter rsvp** is enabled at the BGP level, all RSVP LSPs originating on this node are eligible to be used by default as long as the destination address of the LSP corresponds to that of the BGP next hop for that prefix. It is also possible to exclude a specific RSVP LSP from BGP next hop resolution, similar to the exclusion of a specific RSVP LSP being used as a shortcut for resolving IGP routes. In this example, if the RSVP LSP LSP-PE-6-PE-3 is excluded to be eligible for BGP next hop resolution, it reverts back to LDP.

```

[/]
A:admin@PE-6# configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-6-PE-3" {
        admin-state enable
        delete bgp-shortcut
      }
    }
  }
}

[/]
A:admin@PE-6# show router route-table 10.10.10.0

```

```

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                                Type  Proto  Age           Pref
  Next Hop[Interface Name]                        Metric
-----
10.10.10.0/24                                     Remote BGP      00h00m29s    170
      192.0.2.3 (tunneled:RSVP:4)                  100
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

If the configuration is using **disallow-igp**, and neither LDP nor RSVP LSPs are available, the remote route received via BGP is removed from the GRT although the BGP peer session remains up. A field in the detailed show BGP route output indicates that the next hop is "Unresolved":

```

[/]
A:admin@PE-6# configure {
  router "Base" {
    bgp {
      next-hop-resolution {
        shortcut-tunnel {
          family ipv4 {
            disallow-igp true
          }
        }
      }
    }
  }
}

```

```

[/]
A:admin@PE-6# configure {
  router "Base" {
    ldp {
      admin-state disable
    }
  }
}

```

```

[/]
A:admin@PE-6# show router bgp routes 10.10.10.0/24 detail
=====
BGP Router ID:192.0.2.6      AS:65536      Local AS:65536
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Original Attributes
Network       : 10.10.10.0/24

```

```

Nexthop      : 192.0.2.3
Path Id      : None
From         : 192.0.2.3
Res. Protocol : INVALID           Res. Metric : 0
Res. Nexthop : Unresolved
Local Pref.  : 100                    Interface Name : NotAvailable
Aggregator AS : None                  Aggregator    : None
Atomic Aggr. : Not Atomic             MED           : None
AIGP Metric  : None                   IGP Cost      : 0
Connector    : None
Community    : No Community Members
Cluster      : No Cluster Members
Originator Id : None                  Peer Router Id : 192.0.2.3
Fwd Class    : None                   Priority       : None
Flags       : Invalid Incomplete Nexthop-Unresolved
Route Source : Internal
AS-Path      : No As-Path
Route Tag    : 0
Neighbor-AS  : n/a
Orig Validation: NotFound
Source Class : 0                       Dest Class    : 0
Add Paths Send : Default
RIB Priority  : Normal
Last Modified : 00h10m33s

Modified Attributes

---snip---

-----
Routes : 1
=====

```

Because the route is unresolved, it does not appear in the GRT:

```

[/]
A:admin@PE-6# show router route-table 10.10.10.0

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type   Proto   Age      Pref
Next Hop[Interface Name]          Metric
-----
No. of Routes: 0
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====

```

MPLS/GRE Shortcut for BGP NH Resolution within a VRF

Using RSVP/LDP or GRE shortcuts for resolving BGP next hops within a Virtual Private Routed Network (VPRN), also known as auto-bind-tunnel, allows a VPRN service to automatically resolve the BGP next hop for VPRN routes to an MPLS LSP or a GRE tunnel. Three possible mechanisms exist to provide transport tunnels for forwarding traffic between PE routers within an RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*, network:

- RSVP-TE protocol to create tunnel LSPs between PE routers.
- LDP protocol to create tunnel LSPs between PE routers.
- GRE tunnels between PE routers.

These transport tunneling mechanisms provide the flexibility to use dynamically created LSPs where the service tunnels are automatically bound (the **auto-bind-tunnel** feature), and the ability to provide certain VPN services with their own transport tunnels by explicitly binding SDPs if desired. All services using the auto-bind-tunnel feature use the same set of LSPs, which does not allow for alternate tunneling mechanisms (like GRE) or the ability to craft sets of LSPs with bandwidth reservations for specific customers, as is available with explicit SDPs for the service.

The auto-bind-tunnel configuration is as follows:

```
[/]
A:admin@PE-2# configure {
  service {
    vprn "NON_EXISTING_VPRN" {
      bgp-ipvpn {
        mpls {
          auto-bind-tunnel
          ?

allow-flex-algo-fallback      - Enable flexible algorithm fallback
apply-groups                  - Apply a configuration group at this level
apply-groups-exclude         - Exclude a configuration group at this level
ecmp                          - Maximum ECMP routes allowed
enforce-strict-tunnel-tagging - Allow enforcement of strict tunnel tagging
resolution                    - Resolution method for tunnel selection
resolution-filter             + Enter the resolution-filter context
weighted-ecmp                 - Allow weighted load-balancing

[/]
A:admin@PE-2# configure {
  service {
    vprn "NON_EXISTING_VPRN" {
      bgp-ipvpn {
        mpls {
          auto-bind-tunnel {
            resolution ?

resolution <keyword>
<keyword> - (none|filter|any)
Default   - none

Resolution method for tunnel selection

[/]
A:admin@PE-2# configure {
  service {
    vprn "NON_EXISTING_VPRN" {
      bgp-ipvpn {
        mpls {
          auto-bind-tunnel {
            resolution-filter ?

resolution-filter

bgp - Use BGP tunneling for next-hop resolution
```

gre	- Use GRE tunneling for next-hop resolution
ldp	- Use LDP tunneling for next-hop resolution
mpls-fw-policy	- Use MPLS forwarding policy for next-hop resolution
rib-api	- Use RIB API gRPC service for next-hop resolution
rsvp	- Use RSVP tunneling for next-hop resolution
sr-isis	- Use IS-IS SR tunneling for next-hop resolution
sr-ospf	- Use OSPF SR tunneling for next-hop resolution
sr-ospf3	- Use OSPFv3 SR tunneling for next-hop resolution
sr-policy	- Use SR policies for next-hop resolution
sr-te	- Use SR-TE tunneling for next-hop resolution
udp	- Use MPLS over UDP tunneling for next-hop resolution

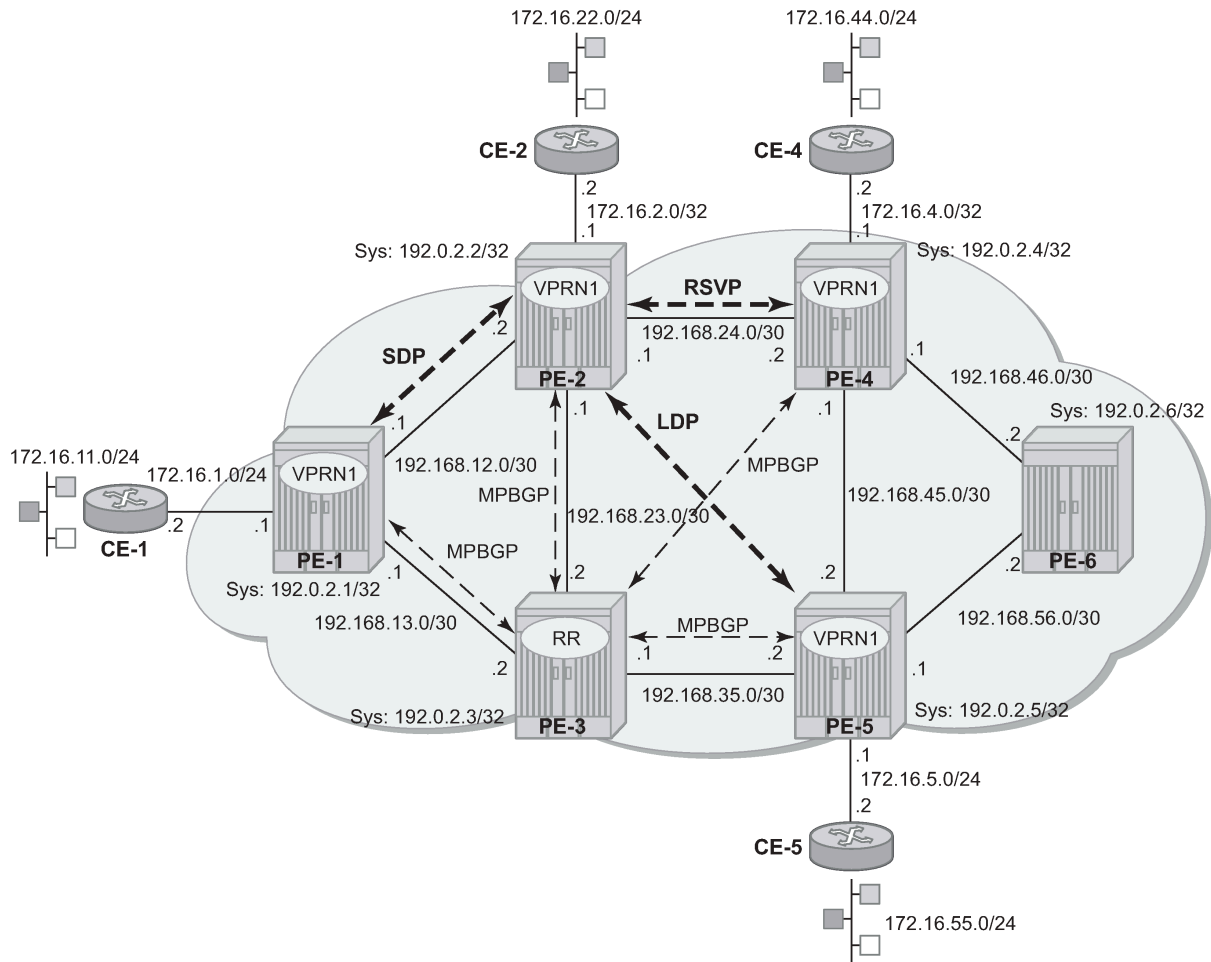
Parameter descriptions:

- **ldp** — Specifies LDP-based LSPs should be used to resolve the BGP next hop for VPRN routes in an associated VPRN instance.
- **gre** — Specifies GRE-based tunnels to be used to resolve the BGP next hop for VPRN routes in an associated VPRN instance. GRE is out of the scope regarding shortcuts, refer to SR OS documentation for further details.
- **rsvp** — Specifies RSVP-TE LSPs should be used to resolve the BGP next hop for VPRN routes in an associated VPRN instance.
- the remaining parameters are beyond the scope of this chapter.

In all cases, if an explicit spoke-SDP is specified in the VPRN, it is always preferred over automatically selected tunnels (even if the SDP is down, the route becomes inactive; there is no fallback to the automatic selection).

The network is configured according to the topology shown in [Figure 211: Shortcuts Within a VRF Topology Network](#). Four PEs (PE-1, PE-2, PE-4, and PE-5) are connected forming a meshed IP-VPN (named VPRN_1), using a route reflector on PE-3 for MP-BGP peering. All PEs have LDP tunnels enabled so at a minimum all can establish LDP shortcut tunnels to the others. In order to have not only LDP but also RSVP-TE LSPs and static SDPs (using an RSVP LSP) in the network, a mix of tunneling methods is configured. For brevity, the configuration of PE-2 only is given, providing the details about the shortcuts created by auto-bind-tunnel. PE-2 has a static SDP (RSVP-based) with PE-1, an RSVP LSP with PE-4, and an LDP LSP with PE-5. Every PE has a CE connected, so each PE has an interface connected to the CE as well as a static route to a CE LAN (although redistribution routing policies are needed, they are not shown for brevity).

Figure 211: Shortcuts Within a VRF Topology Network



OSSG627

On PE-2, VPRN1 is configured as follows:

```

configure {
  service {
    sdp 1 {
      admin-state enable
      far-end {
        ip-address 192.0.2.1
      }
      delivery-type mpls
      lsp "LSP-PE-2-PE-1"
    }
  }
  vprn "VPRN_1" {
    admin-state enable
    service-id 1
    customer "1"
    bgp-ipvpn {
      mpls {
        admin-state enable
        route-distinguisher "65536:1"
        vrf-import {
          policy ["VPRN1-import"]
        }
      }
    }
  }
}

```

```

    }
    vrf-export {
        policy ["VPN1-export"]
    }
    auto-bind-tunnel {
        resolution filter
        resolution-filter {
            gre true
            ldp true
            rsvp true
        }
    }
}
interface "to-CE-2" {
    ipv4 {
        primary {
            address 172.16.2.1
            prefix-length 24
        }
    }
    sap 1/1/4:1 { }
}
static-routes {
    route 172.16.22.0/24 route-type unicast {
        next-hop "172.16.2.2" {
            admin-state enable
        }
    }
}
spoke-sdp 1:1
}
}
}

```

As previously mentioned, regarding IP-VPN meshed connectivity, the configuration shows that there is a static SDP 1 (pointing to PE-1), and the rest of the configuration is just **auto-bind-tunnel**. On PE-2, the connectivity toward the other PEs in the network can be verified by checking VPRN_1:

```

[/]
A:admin@PE-2# show router 1 route-table
=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]                               Type   Proto   Age           Pref
  Next Hop[Interface Name]                       Metric
-----
172.16.1.0/24                                     Remote BGP VPN  00h01m14s  170
    192.0.2.1 (tunneled)                          0
172.16.2.0/24                                     Local  Local   00h09m06s   0
    to-CE-2                                         0
172.16.4.0/24                                     Remote BGP VPN  00h01m14s  170
    192.0.2.4 (tunneled:RSVP:3)                    10
172.16.5.0/24                                     Remote BGP VPN  00h01m14s  170
    192.0.2.5 (tunneled)                          20
172.16.11.0/24                                    Remote BGP VPN  00h01m14s  170
    192.0.2.1 (tunneled)                          0
172.16.22.0/24                                    Remote Static  00h09m07s   5
    172.16.2.2                                       1
172.16.44.0/24                                    Remote BGP VPN  00h01m14s  170
    192.0.2.4 (tunneled:RSVP:3)                    10
172.16.55.0/24                                    Remote BGP VPN  00h01m14s  170

```

```

192.0.2.5 (tunneled)                                     20
-----
No. of Routes: 8
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

As can be seen, there are eight routes because every PE has two routes (one direct PE-CE interface and one static route), so six routes are received from other PEs via MP-BGP. The VPRN_1 routing table can be understood by looking at the tunnel table (active LSPs for remote system IDs):

```

[/]
A:admin@PE-2# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner      Encap TunnelId  Pref  Nexthop          Metric
  Color
-----
192.0.2.1/32         sdp        MPLS    1         5     192.0.2.1         0
192.0.2.1/32         rsvp       MPLS    2         7     192.168.12.1     10
192.0.2.1/32         ldp        MPLS   65537      9     192.168.12.1     10
192.0.2.3/32         ldp        MPLS   65538      9     192.168.23.2     10
192.0.2.4/32         rsvp       MPLS    3         7     192.168.24.2     10
192.0.2.4/32         rsvp       MPLS    4         7     192.168.24.2    16777215
192.0.2.4/32         ldp        MPLS   65545      9     192.168.24.2     10
192.0.2.5/32         ldp        MPLS   65542      9     192.168.23.2     20
192.0.2.6/32         ldp        MPLS   65546      9     192.168.24.2     20
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====

```

The tunnel table shows one entry per LSP per remote PE. The following tunnel selection rules apply:

- SDP has the lowest (best) preference, followed by RSVP and then by LDP.
- If the preference is the same, the lowest metric is selected (ECMP is possible with LDP).

PE-2 has three possibilities to reach PE-1 (192.0.2.1): an SDP tunnel ID 1 with preference 5, an RSVP tunnel ID 2 with preference 7, and an LDP LSP with preference 9. Because SDP tunnel ID 1 has the lowest preference, it is the chosen option. PE-2 has three possibilities to reach PE-4 (192.0.2.4): an RSVP tunnel ID 3 with preference 7 and metric 10, an RSVP tunnel ID 4 with preference 7 and metric 16777215, and an LDP LSP with preference 9; so RSVP tunnel ID 3 is selected. PE-2 only has one option to reach PE-5 and PE-6 (192.0.2.5 and 192.0.2.6) using an LDP LSP.

The following FIB for router VPRN_1 on PE-2 provides more detailed information on the tunneling:

```

[/]
A:admin@PE-2# show router 1 fib 1

=====
FIB Display
=====
Prefix [Flags]                                Protocol
  NextHop
-----

```

```

172.16.1.0/24                                BGP_VPN
  192.0.2.1 (VPRN Label:524281 Transport:SDP:1)
172.16.2.0/24                                LOCAL
  172.16.2.0 (to-CE-2)
172.16.4.0/24                                BGP_VPN
  192.0.2.4 (VPRN Label:524280 Transport:RSVP LSP:3)
172.16.5.0/24                                BGP_VPN
  192.0.2.5 (VPRN Label:524282 Transport:LDP)
172.16.11.0/24                               BGP_VPN
  192.0.2.1 (VPRN Label:524281 Transport:SDP:1)
172.16.22.0/24                               STATIC
  172.16.2.2 (to-CE-2)
172.16.44.0/24                               BGP_VPN
  192.0.2.4 (VPRN Label:524280 Transport:RSVP LSP:3)
172.16.55.0/24                               BGP_VPN
  192.0.2.5 (VPRN Label:524282 Transport:LDP)
-----
Total Entries : 8
=====

```

The FIB shows the chosen transport tunnel, specifying SDP ID, RSVP Tunnel ID, and LDP, as well as service label information linked to the routes.

Static SDP tunnels are preferred over dynamic tunnels (RSVP or LDP auto-bind-tunnel). When the static SDP 1 is shut down or the LSP goes down (there is no fallback to dynamic tunneling), the associated routes are removed:

```

[/]
A:admin@PE-2# configure {
  service {
    sdp 1 {
      admin-state disable
    }
  }
}

```

```

[/]
A:admin@PE-2# show router 1 fib 1

```

```

=====
FIB Display
=====
Prefix [Flags]                                Protocol
NextHop
-----
172.16.2.0/24                                LOCAL
  172.16.2.0 (to-CE-2)
172.16.4.0/24                                BGP_VPN
  192.0.2.4 (VPRN Label:524280 Transport:RSVP LSP:3)
172.16.5.0/24                                BGP_VPN
  192.0.2.5 (VPRN Label:524282 Transport:LDP)
172.16.22.0/24                               STATIC
  172.16.2.2 (to-CE-2)
172.16.44.0/24                               BGP_VPN
  192.0.2.4 (VPRN Label:524280 Transport:RSVP LSP:3)
172.16.55.0/24                               BGP_VPN
  192.0.2.5 (VPRN Label:524282 Transport:LDP)
-----
Total Entries : 6
-----

```

To avoid this fallback issue, the configuration is modified and the manual spoke-SDPs are removed from the configuration of PE-1 and PE-2; the rest of the configuration remains the same. Now the connectivity between PE-1 and PE-2 is using an RSVP LSP, as shown in the PE-1 following output (RSVP LSP which was used by SDP 1 has disappeared):

```
[/]
A:admin@PE-1# configure {
  service {
    vprn "VPRN_1" {
      delete spoke-sdp 1:1
    }
  }
}
```

```
[/]
A:admin@PE-2# configure {
  service {
    vprn "VPRN_1" {
      delete spoke-sdp 1:1
    }
  }
}
```

```
*A:PE-1# show router 1 route-table
```

```
=====
Route Table (Service: 1)
=====
```

Dest Prefix[Flags] Next Hop[Interface Name]	Type	Proto	Age Metric	Pref
172.16.1.0/24 to-CE-1	Local	Local	00h14m16s 0	0
172.16.2.0/24 192.0.2.2 (tunneled:RSVP:2)	Remote	BGP VPN	00h00m23s 10	170
172.16.4.0/24 192.0.2.4 (tunneled)	Remote	BGP VPN	00h00m23s 20	170
172.16.5.0/24 192.0.2.5 (tunneled)	Remote	BGP VPN	00h00m23s 20	170
172.16.11.0/24 172.16.1.2	Remote	Static	00h14m16s 1	5
172.16.22.0/24 192.0.2.2 (tunneled:RSVP:2)	Remote	BGP VPN	00h00m23s 10	170
172.16.44.0/24 192.0.2.4 (tunneled)	Remote	BGP VPN	00h00m23s 20	170
172.16.55.0/24 192.0.2.5 (tunneled)	Remote	BGP VPN	00h00m23s 20	170

```
-----
No. of Routes: 8
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```

If RSVP is disabled, the connectivity falls back to LDP as the output shows:

```
configure {
```

```

router "Base" {
  mpls {
    admin-state disable
  }
}

```

```

[/]
A:admin@PE-1# show router 1 fib 1

```

```

=====
FIB Display
=====
Prefix [Flags]                                Protocol
NextHop
-----
172.16.1.0/24                                  LOCAL
  172.16.1.0 (to-CE-1)
172.16.2.0/24                                  BGP_VPN
  192.0.2.2 (VPRN Label:524281 Transport:LDP)
172.16.4.0/24                                  BGP_VPN
  192.0.2.4 (VPRN Label:524280 Transport:LDP)
172.16.5.0/24                                  BGP_VPN
  192.0.2.5 (VPRN Label:524282 Transport:LDP)
172.16.11.0/24                                 STATIC
  172.16.1.2 (to-CE-1)
172.16.22.0/24                                 BGP_VPN
  192.0.2.2 (VPRN Label:524281 Transport:LDP)
172.16.44.0/24                                 BGP_VPN
  192.0.2.4 (VPRN Label:524280 Transport:LDP)
172.16.55.0/24                                 BGP_VPN
  192.0.2.5 (VPRN Label:524282 Transport:LDP)
-----
Total Entries : 8
=====

```

If LDP is disabled, the connectivity falls back to GRE as the output shows:

```

[/]
A:admin@PE-1# configure {
  router "Base" {
    ldp {
      admin-state disable
    }
  }
}

```

```

*A:PE-1# show router 1 fib 1

```

```

=====
FIB Display
=====
Prefix [Flags]                                Protocol
NextHop
-----
172.16.1.0/24                                  LOCAL
  172.16.1.0 (to-CE-1)
172.16.2.0/24                                  BGP_VPN
  192.0.2.2 (VPRN Label:524280 Transport:GRE)
172.16.4.0/24                                  BGP_VPN
  192.0.2.4 (VPRN Label:524280 Transport:GRE)

```

```
172.16.5.0/24 BGP_VPN
  192.0.2.5 (VPRN Label:524286 Transport:GRE)
172.16.11.0/24 STATIC
  172.16.1.2 (to-CE-1)
172.16.22.0/24 BGP_VPN
  192.0.2.2 (VPRN Label:524280 Transport:GRE)
172.16.44.0/24 BGP_VPN
  192.0.2.4 (VPRN Label:524280 Transport:GRE)
172.16.55.0/24 BGP_VPN
  192.0.2.5 (VPRN Label:524286 Transport:GRE)
-----
Total Entries : 8
-----
=====
```

Conclusion

IGP shortcuts provide a variety of shortcuts in IP, MPLS, and IP-VPN scenarios to customers who want to use new options for building routing topologies. Because IGP shortcuts are enabled on a per router basis, SPF computations are independent and irrelevant to other routers, so there is no need to enable shortcuts globally. This network example shows the configuration of IGP shortcuts together with the associated show outputs which can be used for verification and troubleshooting.

Inter-Area TE Point-to-Point LSPs

This chapter describes inter-area Traffic Engineering (TE) Point-to-Point (P2P) Label Switched Paths (LSPs) configurations.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 11.0.R4, but the MD-CLI in the current edition corresponds to SR OS Release 21.2.R1.

Overview

Multi-Protocol Label Switching with Traffic Engineering (MPLS TE) is implemented on a wide scale in current Internet Service Provider (ISP) networks to steer traffic across their backbones to facilitate efficient use of available bandwidth between the routers and to guarantee fast convergence in case a link or node fails.

Regular TE LSPs in MPLS networks are confined to only a single Interior Gateway Protocol (IGP) area or level. This is because the head-end has information in the TE database of only the local area for Open Shortest Path First (OSPF) or level for Intermediate System to Intermediate System (IS-IS). As the name implies, inter-area TE LSPs can cross the area or level borders of the IGP.

Inter-Area TE LSP based on Explicit Route expansion

Inter-area TE LSP using Explicit Route Object (ERO) expansion enables the head-end to calculate the ERO path within its own area or level and keep the remaining Area Border Routers (ABRs) of other areas/levels as loose hops in the ERO path. On receiving a PATH message with a loose hop ERO and based on local configuration, each ABR does a partial Constrained Shortest Path First (CSPF) calculation to the next ABR or a full CSPF calculation to reach the destination.

Automatic selection of ABRs is supported so that the head-end node can work with an empty primary path. When the **to** field of an LSP definition is in an area/level different from the head-end node, CSPF will automatically compute the segment to the exit ABR router which advertised the prefix and which currently is the best path for resolving the prefix in the Route Table Manager (RTM).

ABR protection

Link and node protection within the respective areas are supported through the TE capabilities of the IGP and Resource Reservation Protocol (RSVP) in each area. To support ABR node protection, a bypass is

required from the Point of Local Repair (PLR; node prior to ABR) to the Merge Point (MP; next-hop node to ABR).

Two methods are possible: static ABR protection and dynamic ABR protection. Static ABR protection uses Manual Bypass Tunnels (MBTs), statically configured by the operator between the PLR and the MP. For dynamic ABR protection, node ID propagation and signaling of an eXclude Route Object (XRO) in RSVP PATH messages must both be supported.

Because the Record Route Object (RRO) Node ID sub-object description in RFC 4561 (*Definition of a Record Route Object (RRO) Node-Id Sub-Object*) is not clear about the format of the included node address (S), interface address (I) and label (L), the system supports multiple formats: IL, SL, ISL, SIL, SLI, ILSL and SLIL. The system uses the SLIL (node-address, label, interface-address, label) format to include the node ID itself.

The exclude route object (XRO) inclusion (RFC 4874, *Exclude Routes - Extension to Resource ReserVation Protocol-Traffic Engineering*) in bypass RSVP PATH messages is required to exclude the protected ABR from the bypass path. The XRO object contains the ABR system IP address.

Example topology

The example topology in this chapter contains ten nodes in three areas, as shown in [Figure 212: Inter-area TE LSP setup](#).

Figure 212: Inter-area TE LSP setup

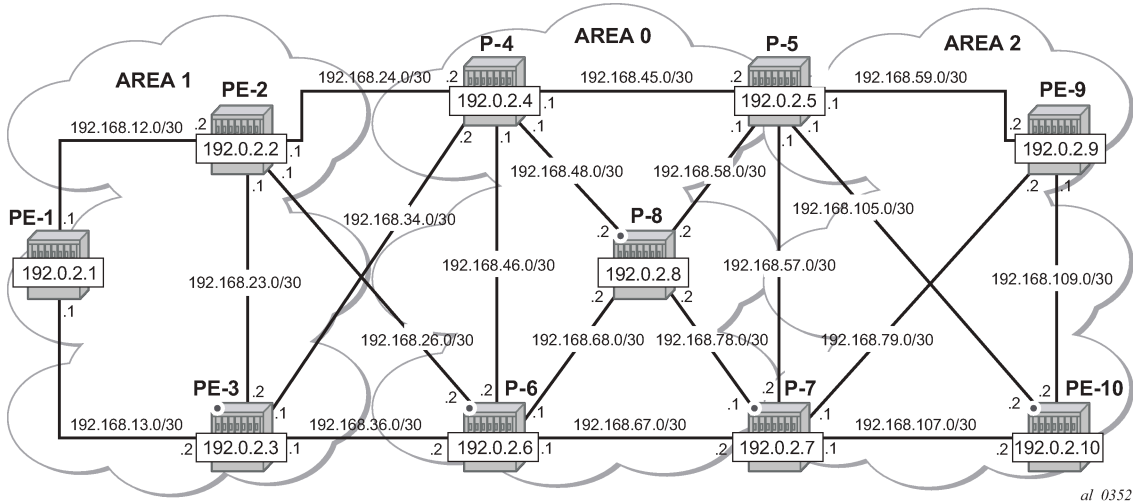
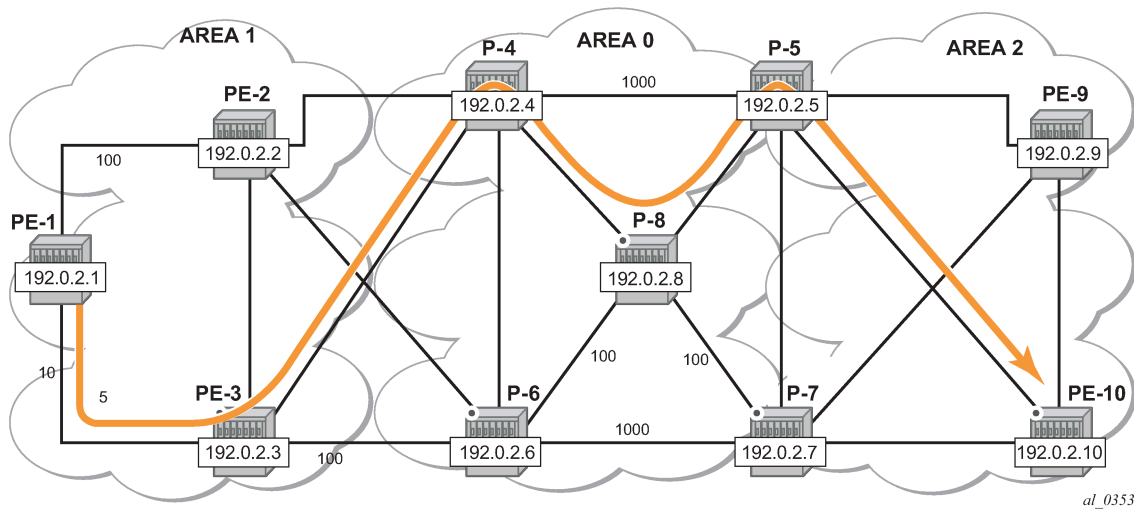


Figure 213: Inter-area TE LSP path shows the LSP path intended to be set up through the network. An empty MPLS path is used. At the head-end node PE-1, the destination address PE-10 is learned via ABR node P-4 and ABR node P-5.

Figure 213: Inter-area TE LSP path



Configuration

The following base configuration has been implemented on the nodes:

- Cards, MDAs, and ports configured
- Interfaces configured
- IGP areas configured and converged
- Traffic Engineering configured for the IGP
- MPLS and RSVP configured on all links in the network

OSPF or IS-IS can be configured as the IGP; OSPF is used in this chapter.

The following output shows the opaque database of PE-1:

```
[/]
A:admin@PE-1# show router ospf opaque-database

=====
Rtr Base OSPFv2 Instance 0 Opaque Link State Database (type: All)
=====
Type  Id                Link State Id      Adv Rtr Id         Age  Sequence           Cksum
-----
Area  0.0.0.1             1.0.0.1            192.0.2.1          269  0x80000002         0x9234
Area  0.0.0.1             1.0.0.2            192.0.2.1          236  0x80000001         0xa53c
Area  0.0.0.1             1.0.0.3            192.0.2.1          215  0x80000001         0xf3e9
Area  0.0.0.1             1.0.0.1            192.0.2.2          230  0x80000002         0x962e
Area  0.0.0.1             1.0.0.2            192.0.2.2          199  0x80000001         0xe6df
Area  0.0.0.1             1.0.0.3            192.0.2.2          165  0x80000001         0x8d32
Area  0.0.0.1             1.0.0.4            192.0.2.2          237  0x80000001         0x6d72
Area  0.0.0.1             1.0.0.5            192.0.2.2          215  0x80000001         0x7056
Area  0.0.0.1             1.0.0.1            192.0.2.3          209  0x80000002         0x9a28
Area  0.0.0.1             1.0.0.2            192.0.2.3          200  0x80000001         0x773a
Area  0.0.0.1             1.0.0.3            192.0.2.3          165  0x80000001         0x1e8c
Area  0.0.0.1             1.0.0.4            192.0.2.3          216  0x80000001         0xa933
Area  0.0.0.1             1.0.0.5            192.0.2.3          216  0x80000001         0x4c7a
```

```

Area 0.0.0.1      1.0.0.1      192.0.2.4      194  0x80000002  0x9e22
Area 0.0.0.1      1.0.0.6      192.0.2.4      201  0x80000001  0x7e44
Area 0.0.0.1      1.0.0.7      192.0.2.4      201  0x80000001  0x218b
Area 0.0.0.1      1.0.0.1      192.0.2.6      160  0x80000002  0xa616
Area 0.0.0.1      1.0.0.6      192.0.2.6      166  0x80000001  0xf6c5
Area 0.0.0.1      1.0.0.7      192.0.2.6      166  0x80000001  0x990d
-----
No. of Opaque LSAs: 19
=====

```

The information is only about routers that are part of area 0.0.0.1. PE-1 cannot calculate an end-to-end CSPF path to node PE-10 because this would require TE topology information from area 0.0.0.0 and area 0.0.0.2.

Each node announces its router ID and each attached link that is part of that area, resulting in 19 opaque LSAs in area 0.0.0.1. The system interfaces of P-4 and P-6 are configured in backbone area 0.0.0.0, not in area 0.0.0.1.

In [Figure 213: Inter-area TE LSP path](#), the LSP passes through node PE-3 and node P-8. To prefer a dynamic path from PE-1 to P-4 via PE-3 rather than via PE-2, it is necessary to configure on PE-1 a lower IGP metric on the interface to PE-3 (the default metric is derived from the interface speed; in this case the metric is 10 by default).

```

# on PE-1:
configure {
  router "Base" {
    ospf 0 {
      area 0.0.0.1 {
        interface "int-PE-1-PE-3" {
          metric 5
        }
      }
    }
  }
}

```

Similarly, in the core, the IGP metric between P-4 and P-5, and between P-6 and P-7 is increased to force the LSP to pass through the core P-8 node.

```

# on P-4:
configure {
  router "Base" {
    ospf 0 {
      area 0.0.0.0 {
        interface "int-P-4-P-5" {
          metric 1000
        }
      }
    }
  }
}

```

```

# on P-6:
configure {
  router "Base" {
    ospf 0 {
      area 0.0.0.0 {
        interface "int-P-6-P-7" {
          metric 1000
        }
      }
    }
  }
}

```

Other metrics have also been manipulated as shown on [Figure 213: Inter-area TE LSP path](#).

MPLS path configuration

An empty MPLS path is sufficient on the head-end node PE-1 because automatic ABR selection is performed. Using an empty MPLS path will ease the provisioning process and brings consistency because this empty MPLS path can be used for both intra and inter-area/level type LSPs.

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      path "empty_path" {
        admin-state enable
      }
    }
  }
}
```

MPLS LSP configuration

On PE-1, the following LSP to PE-10 is configured with the previously created MPLS path as primary path. CSPF and fast reroute (FRR) facility are enabled on the LSP.

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-10" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.10
        path-computation-method local-cspf
        fast-reroute {
          frr-method facility
        }
        primary "empty_path" {
        }
      }
    }
  }
}
```

At this stage, the LSP is in an operational Down state with a failure code of badNode at failure node 192.168.34.2 (ABR P-4), as follows.

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-10" path

=====
MPLS LSP LSP-PE-1-PE-10 Path
=====
-----
LSP Name   : LSP-PE-1-PE-10
From       : 0.0.0.0
To         : 192.0.2.10
Adm State  : Up                               Oper State   : Down
-----
Path Name           Next Hop           Type           Out I/F      Adm  Opr
-----
empty_path                n/a              Primary        n/a          Up   Dwn
=====
```

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-10" path detail
```

```

=====
MPLS LSP LSP-PE-1-PE-10 Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use
  b - Bandwidth Protected      n - Node Protected
  s - Soft Preemption
  S - Strict                    L - Loose
  A - ABR                       + - Inherited
=====
-----
LSP LSP-PE-1-PE-10
Path empty_path
-----
LSP Name      : LSP-PE-1-PE-10
From          : 192.0.2.1
To           : 192.0.2.10
Admin State   : Up                Oper State      : Down
Path Name     : empty_path
Path LSP ID   : 24596             Path Type       : Primary
Path Admin    : Up                Path Oper       : Down
Out Interface : n/a              Out Label       : n/a
Path Up Time  : 0d 00:00:00      Path Down Time  : 0d 00:05:22
Retry Limit   : 0                 Retry Timer     : 30 sec
Retry Attempt : 11               Next Retry In   : 20 sec

BFD Configuration and State
Template      : None                Ping Interval   : 60
Enable       : False              State           : notApplicable
WaitForUpTm  : 4 sec              OperWaitForUpTm: N/A
WaitForUpTmLeft : 0 sec

Adspec       : Disabled            Oper Adspec     : N/A
PathCompMethod : local-cspf        OperPathCompMethod: N/A
MetricType    : igp                Oper MetricType : N/A
Least Fill    : Disabled           Oper LeastFill  : N/A
FRR           : Enabled            Oper FRR        : N/A
FRR NodeProtect : Enabled          Oper FRR NP     : N/A
FR Hop Limit  : 16                 Oper FRHopLimit : N/A
FR Prop Admin Gr*: Disabled        Oper FRPropAdmGrp : N/A
Propagate Adm Grp: Disabled        Oper Prop Adm Grp : N/A
Inter-area    : N/A

PCE Report    : Disabled+          Oper PCE Report : Disabled
PCE Control   : Disabled          Oper PCE Control : Disabled
PCE Update ID : 0

Neg MTU       : 0                  Oper MTU        : N/A
Bandwidth     : No Reservation     Oper Bandwidth  : N/A
Hop Limit     : 255                Oper HopLimit   : N/A
Record Route  : Record             Oper Record Route : N/A
Record Label  : Record             Oper Record Label : N/A
Setup Priority : 7                  Oper SetupPriority: N/A
Hold Priority  : 0                  Oper HoldPriority : N/A
Class Type    : 0                  Oper CT         : N/A
Backup CT     : None
MainCT Retry  : Infinite
  Rem         :
MainCT Retry  : 0
  Limit      :
Include Groups :                    Oper IncludeGroups:
None          :                    N/A
Exclude Groups :                    Oper ExcludeGroups:

```

```

None                                     N/A
Adaptive           : Enabled             Oper Metric      : N/A
Preference        : n/a
Path Trans        : 0                   CSPF Queries     : 11
Failure Code     : badNode
Failure Node    : 192.168.34.2
Explicit Hops     :
  No Hops Specified
Actual Hops       :
  No Hops Specified
Computed Hops     :
  No Hops Specified
Resignal Eligible: False
Last Resignal    : n/a                 CSPF Metric      : N/A
=====
* indicates that the corresponding row element may have been truncated.

```

To get around the intra-area CSPF confinement, the ERO-expansion feature is enabled on all ABR nodes.

```

# on P-4, P-5, P-6, P-7:
configure {
  router "Base" {
    mpls {
      cspf-on-loose-hop true
    }
  }
}

```

cspf-on-loose-hop is only required if FRR or TE parameters are configured on the LSP. If any of these parameters is configured on the LSP while one of the ABRs along the path is not configured with **cspf-on-loose-hop true**, the LSP will stay operationally down with failure code: badNode and an indication of the interface address of the failure node.

The LSP path can also contain other strict and/or loose hops. However, **cspf-on-loose-hop** must be enabled in the **mpls** context whenever loose hops are configured in the MPLS path. This command enables ERO expansion and is required for inter-area LSPs on all possible ABR nodes and all nodes not belonging to the area where the iLER is located, which have a loose hop reference in the MPLS path.



Note: The LSP may fail to set up if **cspf-on-loose-hop** is enabled on an LSR that is not an ABR and that receives a PATH message without a proper next loose hop in the ERO.

On all nodes, debugging is enabled—in classic CLI—for RSVP PATH messages, as follows:

```

# on all nodes:
debug
  router "Base"
    rsvp
      packet
        path detail
      exit
    exit
  exit
exit

```

The following RSVP PATH message on PE-1 shows the ERO calculation on the head-end to the first ABR.

```

# on PE-1:
2 2021/05/12 08:13:36.522 UTC MINOR: DEBUG #2001 Base RSVP
"RSVP: PATH Msg
Send PATH From:192.0.2.1, To:192.0.2.10
      TTL:255, Checksum:0x5b2d, Flags:0x0
Session - EndPt:192.0.2.10, TunnId:1, ExtTunnId:192.0.2.1

```

```
SessAttr - Name:LSP-PE-1-PE-10::empty_path
          SetupPri:7, HoldPri:0, Flags:0x17
RSVPHop - Ctype:1, Addr:192.168.13.1, LIH:3
TimeValue - RefreshPeriod:30
SendTempl - Sender:192.0.2.1, LspId:24648
SendTSpec - Ctype:QOS, CDR:0.000 bps, PBS:0.000 bps, PDR:infinity
          MPU:20, MTU:1564
LabelReq - IfType:General, L3ProtID:2048
RRO - IpAddr:192.168.13.1, Flags:0x0
ERO - IPv4Prefix 192.168.13.2/32, Strict
          IPv4Prefix 192.168.34.2/32, Strict
          IPv4Prefix 192.0.2.10/32, Loose
FRRObj - SetupPri:7, HoldPri:0, HopLimit:16, BW:0.000 bps, Flags:0x2
ExcAny:0x0, IncAny:0x0, IncAll:0x0
"
```

On the ABR P-4, the ERO is expanded to include the nodes of area 0.0.0.0 of which P-4 is also part. The RRO contains all the hops the PATH message has passed so far.

```
# on P-4:
2 2021/05/12 08:13:24.842 UTC MINOR: DEBUG #2001 Base RSVP
"RSVP: PATH Msg
Send PATH From:192.0.2.1, To:192.0.2.10
          TTL:253, Checksum:0xe9a8, Flags:0x0
Session - EndPt:192.0.2.10, TunnId:1, ExtTunnId:192.0.2.1
SessAttr - Name:LSP-PE-1-PE-10::empty_path
          SetupPri:7, HoldPri:0, Flags:0x17
RSVPHop - Ctype:1, Addr:192.168.48.1, LIH:4
TimeValue - RefreshPeriod:30
SendTempl - Sender:192.0.2.1, LspId:24648
SendTSpec - Ctype:QOS, CDR:0.000 bps, PBS:0.000 bps, PDR:infinity
          MPU:20, MTU:1564
LabelReq - IfType:General, L3ProtID:2048
RRO - IpAddr:192.168.48.1, Flags:0x0
          IpAddr:192.168.34.1, Flags:0x0
          IpAddr:192.168.13.1, Flags:0x0
ERO - IPv4Prefix 192.168.48.2/32, Strict
          IPv4Prefix 192.168.58.1/32, Strict
          IPv4Prefix 192.0.2.10/32, Loose
FRRObj - SetupPri:7, HoldPri:0, HopLimit:16, BW:0.000 bps, Flags:0x2
ExcAny:0x0, IncAny:0x0, IncAll:0x0
"
```

Finally, the P-5 ABR will expand the ERO to the final destination PE-10:

```
# on P-5:
7 2021/05/12 08:13:38.230 UTC MINOR: DEBUG #2001 Base RSVP
"RSVP: PATH Msg
Send PATH From:192.0.2.1, To:192.0.2.10
          TTL:251, Checksum:0x9207, Flags:0x0
Session - EndPt:192.0.2.10, TunnId:1, ExtTunnId:192.0.2.1
SessAttr - Name:LSP-PE-1-PE-10::empty_path
          SetupPri:7, HoldPri:0, Flags:0x17
RSVPHop - Ctype:1, Addr:192.168.105.1, LIH:5
TimeValue - RefreshPeriod:30
SendTempl - Sender:192.0.2.1, LspId:24648
SendTSpec - Ctype:QOS, CDR:0.000 bps, PBS:0.000 bps, PDR:infinity
          MPU:20, MTU:1564
LabelReq - IfType:General, L3ProtID:2048
RRO - IpAddr:192.168.105.1, Flags:0x0
          IpAddr:192.168.58.2, Flags:0x0
          IpAddr:192.168.48.1, Flags:0x0
```

```

IpAddr:192.168.34.1, Flags:0x0
IpAddr:192.168.13.1, Flags:0x0
ERO - IPv4Prefix 192.168.105.2/32, Strict
FRR0bj - SetupPri:7, HoldPri:0, HopLimit:16, BW:0.000 bps, Flags:0x2
"      - ExcAny:0x0, IncAny:0x0, IncAll:0x0

```

The MPLS LSP is now operational Up and the LSP path can be shown in detail on the head-end, PE-1:

```

[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-10" path

=====
MPLS LSP LSP-PE-1-PE-10 Path
=====
-----
LSP Name   : LSP-PE-1-PE-10
From       : 0.0.0.0
To         : 192.0.2.10
Adm State  : Up
Oper State : Up
-----
Path Name      Next Hop      Type      Out I/F  Adm  Opr
-----
empty_path     192.168.13.2 Primary    1/1/2    Up   Up
=====

```

```

[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-10" path detail

=====
MPLS LSP LSP-PE-1-PE-10 Path (Detail)
=====
Legend :
@ - Detour Available      # - Detour In Use
b - Bandwidth Protected  n - Node Protected
s - Soft Preemption
S - Strict                L - Loose
A - ABR                   + - Inherited
=====
-----
LSP LSP-PE-1-PE-10
Path empty_path
-----
LSP Name   : LSP-PE-1-PE-10
From       : 192.0.2.1
To         : 192.0.2.10
Admin State : Up
Oper State : Up
Path Name  : empty_path
Path LSP ID : 24648
Path Type  : Primary
Path Admin : Up
Path Oper  : Up
Out Interface : 1/1/2
Out Label  : 524287
Path Up Time : 0d 00:02:07
Path Down Time : 0d 00:00:00
Retry Limit : 0
Retry Timer : 30 sec
Retry Attempt : 0
Next Retry In : 0 sec
-----
---snip---
Adspec      : Disabled
Oper Adspec : Disabled
PathCompMethod : local-cspf
OperPathCompMethod : local-cspf
MetricType  : igp
Oper MetricType : igp
Least Fill  : Disabled
Oper LeastFill : Disabled

```



```

FRR           : Enabled           Oper FRR       : Enabled
FRR NodeProtect : Enabled         Oper FRR NP    : Enabled
FR Hop Limit   : 16               Oper FRHopLimit : 16
FR Prop Admin Gr*: Disabled       Oper FRPropAdmGrp : Disabled
Propagate Adm Grp: Disabled       Oper Prop Adm Grp : Disabled
Inter-area     : True

---snip---

Adaptive       : Enabled           Oper Metric    : 15
Preference     : n/a              CSPF Queries   : 37
Path Trans     : 1
Failure Code   : noError
Failure Node   : n/a
Explicit Hops  :
  No Hops Specified
Actual Hops    :
  192.168.13.1(192.0.2.1) @ n      Record Label   : N/A
-> 192.168.13.2(192.0.2.3) @      Record Label   : 524287
-> 192.168.34.2(192.0.2.4) @ n    Record Label   : 524287
-> 192.168.48.2 @                  Record Label   : 524287
-> 192.168.58.1 @                  Record Label   : 524287
-> 192.168.105.2 @                 Record Label   : 524287
Computed Hops  :
  192.168.13.1(S)
-> 192.168.13.2(S)
-> 192.168.34.2(SA)
-> 192.0.2.10(L)
Resignal Eligible: False
Last Resignal  : n/a              CSPF Metric    : 15
=====
* indicates that the corresponding row element may have been truncated.

```

ABR node protection

The LSP is configured with facility FRR protection; link and node protection are established within each area, as shown in the preceding output. Node protection is available for nodes PE-3 in area 1 (bypass originating in PE-1), and P-8 in area 0 (bypass originating in P-4), but not for the ABRs P-4 and P-5. No bypass tunnels for node protection originate in PLRs PE-3 (for ABR P-4) or P-8 (for ABR P-5). The bypass tunnels originating in PE-3 and P-8 only offer link protection. Dynamic ABR node protection requires the setup of a bypass tunnel from the PLR (node just upstream of the ABR) to the MP (node just downstream of the ABR). The following two things are required to establish a bypass tunnel for an ABR:

- The PLR node (part of area x) needs to know the system IP address of the MP node (part of area y) to set up the bypass. For this reason, the node ID of the MP must be included in the RESV message so that the PLR can link the manual bypass tunnel to the primary path to protect the ABR. By default, the node ID is not included in the RESV message, but it can be configured on the MPs with the **include-node-id-in-rro** command.
- The other ABR node receiving the RSVP bypass PATH message for the protected ABR needs to do an ERO expansion toward the MP node. For this reason, the XRO object is included in the RSVP bypass PATH message, containing the node ID of the protected ABR. As an example, the following bypass PATH message is shown on node PE-3.

The XRO object includes the system IP address of the protected ABR node P-4 and the ERO object has MP node P-8 as loose destination:

```
# on PE-3:
```

```

46 2021/05/12 08:20:46.609 UTC MINOR: DEBUG #2001 Base RSVP
"RSVP: PATH Msg
Send PATH From:192.0.2.3, To:192.0.2.8
      TTL:17, Checksum:0xfddd, Flags:0x0
Session   - EndPt:192.0.2.8, TunnId:61442, ExtTunnId:192.0.2.3
SessAttr  - Name:bypass-node192.0.2.4-61442
          - SetupPri:7, HoldPri:0, Flags:0x2
RSVPHop   - Ctype:1, Addr:192.168.36.1, LIH:3
TimeValue - RefreshPeriod:30
SendTempl - Sender:192.0.2.3, LspId:4
SendTSpec - Ctype:QOS, CDR:0.000 bps, PBS:0.000 bps, PDR:infinity
          - MPU:20, MTU:1564
LabelReq  - IfType:General, L3ProtID:2048
RR0       - IpAddr:192.168.36.1, Flags:0x0
ER0       - IPv4Prefix 192.168.36.2/32, Strict
IPv4Prefix 192.0.2.8/32, Loose
XR0       - IPv4Prefix: 192.0.2.4/32, Attribute: Node, LBit: Exclude
AdSpec    - General BreakBit:0, NumISHops:0, PathBwEstimate:0
          - MinPathLatency:4294967295, CompPathMTU:1564
          - Controlled BreakBit:0
"

```

Node-ID inclusion in the RESV message

P-8 will be the MP for the bypass of ABR P-4 and PE-10 will be the MP for the bypass of ABR P-5. So P-8 and PE-10 need to include their node ID in the RESV message, inside the Record Route Object (RRO).

```

# on P-8 and PE-10:
configure {
  router "Base" {
    rsvp {
      include-node-id-in-rro true
    }
  }
}

```

The default is **include-node-id-in-rro false**.

On PE-3, debugging is enabled—in classic CLI—for RSVP RESV messages, as follows:

```

# on PE-3:
debug
  router "Base"
    rsvp
      packet
        resv detail
      exit
    exit

```

The following RESV message is received on PLR node PE-3. The RRO contains the MP node P-8 information in SLIL format:

```

52 2021/05/12 08:21:19.609 UTC MINOR: DEBUG #2001 Base RSVP
"RSVP: RESV Msg
Send RESV From:192.168.13.2, To:192.168.13.1
      TTL:255, Checksum:0x7e4b, Flags:0x0
Session   - EndPt:192.0.2.10, TunnId:1, ExtTunnId:192.0.2.1
RSVPHop   - Ctype:1, Addr:192.168.13.2, LIH:3
TimeValue - RefreshPeriod:30
Style     - SE
FlowSpec  - Ctype:QOS, CDR:0.000 bps, PBS:0.000 bps, PDR:infinity
          - MPU:20, MTU:1560, RSpecRate:0, RSpecSlack:0
"

```

```

FilterSpec - Sender:192.0.2.1, LspId:24648, Label:524287
RR0
- InterfaceIp:192.168.13.2, Flags:0x9
  Label:524287, Flags:0x1
  InterfaceIp:192.168.34.2, Flags:0x9
  Label:524287, Flags:0x1
  SystemIp:192.0.2.8, Flags:0x29
  Label:524287, Flags:0x1
  InterfaceIp:192.168.48.2, Flags:0x9
  Label:524287, Flags:0x1
  SystemIp:192.0.2.5, Flags:0x21
  Label:524287, Flags:0x1
  InterfaceIp:192.168.58.1, Flags:0x1
  Label:524287, Flags:0x1
  SystemIp:192.0.2.10, Flags:0x20
  Label:524287, Flags:0x1
  InterfaceIp:192.168.105.2, Flags:0x0
  Label:524287, Flags:0x1
"
    
```

Bypass configuration for ABR protection

Because dynamic ABR protection is supported and used in this example, no explicit Manual Bypass Tunnels (MBTs) are configured to protect the ABRs. Each PLR first checks if an MBT tunnel exists between the PLR and the MP matching the constraints and protecting the ABR. If no MBT is available, the PLR will signal a bypass tunnel in a dynamic way toward the MP node.

Figure 214: ABR protection shows the two dynamic ABR node protections that are signaled for this LSP.

Figure 214: ABR protection

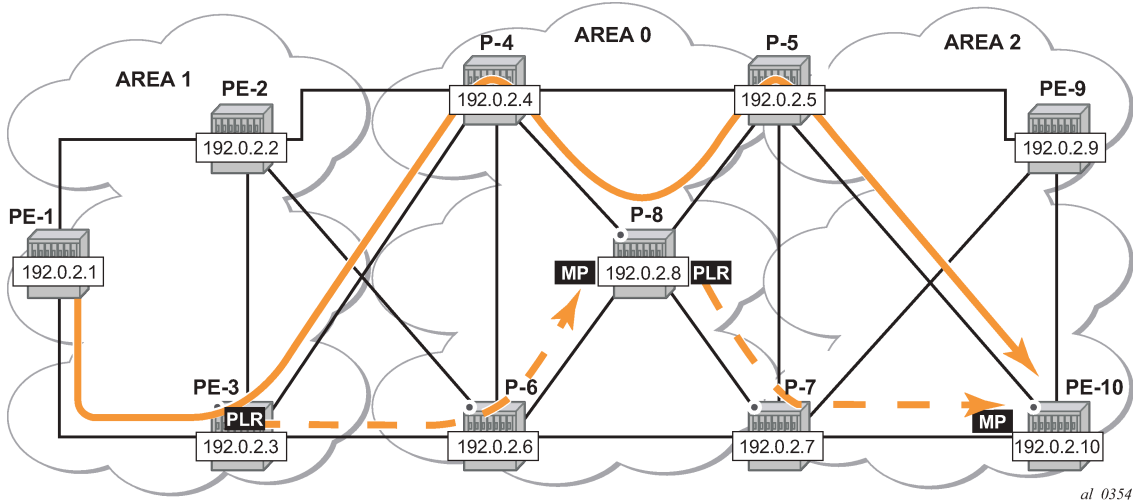
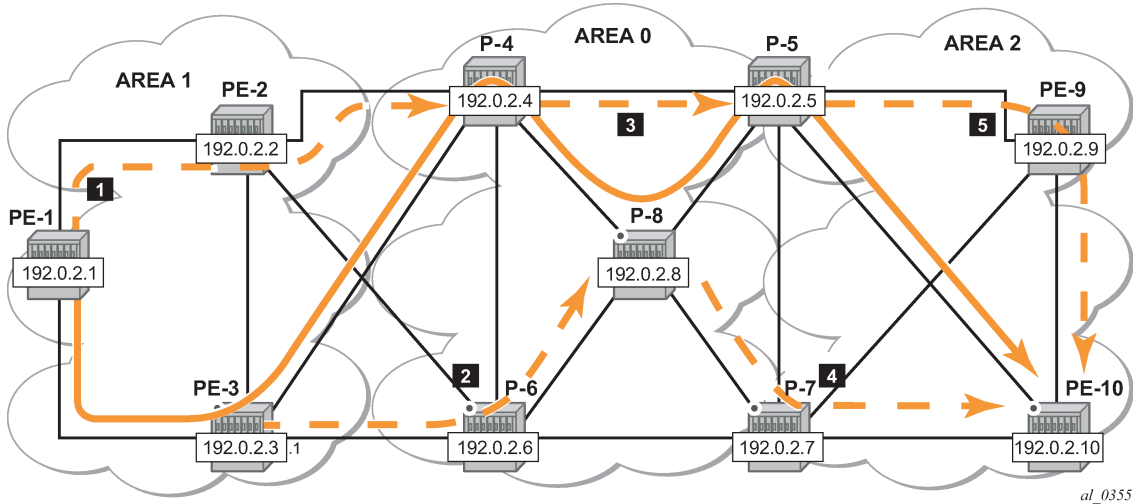


Figure 215: Protection of all nodes/links along the LSP path shows the complete picture of all the FRR protections and indicates each node/link protection in the setup.

Figure 215: Protection of all nodes/links along the LSP path



This can be seen in the detailed show output of the LSP path:

```
[/]  
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-10" path detail  
  
=====
```

MPLS LSP LSP-PE-1-PE-10 Path (Detail)

```
=====
```

Legend :

@ - Detour Available	# - Detour In Use
b - Bandwidth Protected	n - Node Protected
s - Soft Preemption	L - Loose
S - Strict	+ - Inherited
A - ABR	

```
=====
```

LSP LSP-PE-1-PE-10
Path empty_path

```
-----
```

LSP Name	: LSP-PE-1-PE-10	Oper State	: Up
From	: 192.0.2.1	Path Type	: Primary
To	: 192.0.2.10	Path Oper	: Up
Admin State	: Up	Out Interface	: 1/1/2
Path Name	: empty_path	Out Label	: 524287
Path LSP ID	: 24648	Path Up Time	: 0d 00:19:36
Path Admin	: Up	Path Down Time	: 0d 00:00:00
Retry Limit	: 0	Retry Timer	: 30 sec
Retry Attempt	: 0	Next Retry In	: 0 sec

```
-----
```

---snip---

Adspec	: Disabled	Oper Adspec	: Disabled
PathCompMethod	: local-cspf	OperPathCompMethod	: local-cspf
MetricType	: igp	Oper MetricType	: igp
Least Fill	: Disabled	Oper LeastFill	: Disabled
FRR	: Enabled	Oper FRR	: Enabled
FRR NodeProtect	: Enabled	Oper FRR NP	: Enabled

```

FR Hop Limit      : 16                Oper FRHopLimit   : 16
FR Prop Admin Gr*: Disabled          Oper FRPropAdmGrp : Disabled
Propagate Adm Grp: Disabled          Oper Prop Adm Grp : Disabled
Inter-area       : True

---snip---

Adaptive         : Enabled            Oper Metric       : 15
Preference       : n/a
Path Trans       : 1                  CSPF Queries      : 37
Failure Code     : noError
Failure Node    : n/a
Explicit Hops    :
  No Hops Specified
Actual Hops      :
  192.168.13.1(192.0.2.1) @ n        Record Label     : N/A
-> 192.168.13.2(192.0.2.3) @ n        Record Label     : 524287
-> 192.168.34.2(192.0.2.4) @ n        Record Label     : 524287
-> 192.0.2.8(192.0.2.8) @ n          Record Label     : 524287
-> 192.168.48.2 @ n                  Record Label     : 524287
-> 192.0.2.5(192.0.2.5) @           Record Label     : 524287
-> 192.168.58.1 @                   Record Label     : 524287
-> 192.0.2.10(192.0.2.10)            Record Label     : 524287
-> 192.168.105.2                     Record Label     : 524287
Computed Hops    :
  192.168.13.1(S)
-> 192.168.13.2(S)
-> 192.168.34.2(SA)
-> 192.0.2.10(L)
Resignal Eligible: False
Last Resignal   : n/a                CSPF Metric       : 15
=====
* indicates that the corresponding row element may have been truncated.

```

- The first bypass originates in PE-1 and protects node PE-3.
- The second bypass originates in PE-3 and protects node P-4.
- The third bypass originates in P-4 and protects node P-8.
- The fourth bypass originates in P-8 and protects node P-5. There are two entries for P-8: hop 192.0.2.8 and hop 192.168.48.2.
- The fifth bypass originates in P-5 and protects the link between P-5 and PE-10. There are two entries for P-5: hop 192.0.2.5 and hop 192.168.58.1.

There are two entries for P-8, P-5 and PE-10 in the 'Actual Hops' section in the previous output: one for the interface IP address and one for the system IP address. This is a consequence of configuring **include-node-id-in-rro true** on P-8, P-5, and PE-10.

The **include-node-id-in-rro true** command is not mandatory for this example on ABR node P-5, but to be able to cover cases where a new LSP is established in the network and P-5 acts as an MP node while the corresponding PLR node for that new LSP is in another area. This RSVP command can be executed on all possible MP nodes in the network.

The following command shows the details of the bypass tunnel from PE-3 to PE-8, protecting PE-4:

```

[/]
A:admin@PE-3# show router mpls bypass-tunnel protected-lsp detail

=====
MPLS Bypass Tunnels (Detail)
=====

```

```

-----
bypass-node192.0.2.4-61442
-----
To           : 192.0.2.8           State           : Up
Out I/F      : 1/1/2              Out Label      : 524287
Up Time     : 0d 00:08:18        Active Time    : n/a
Reserved BW  : 0 Kbps            Protected LSP Count : 1
Type        : Dynamic           Bypass Path Cost : 100
Setup Priority : 7                Hold Priority   : 0
Class Type   : 0
Exclude Node : 192.0.2.4         Inter-Area     : True
Computed Hops :
  192.168.36.1(S)              Egress Admin Groups : None
-> 192.168.36.2(SA)            Egress Admin Groups : None
-> 192.0.2.8(L)                Egress Admin Groups : None
Actual Hops  :
  192.168.36.1(192.0.2.3)      Record Label     : N/A
-> 192.168.36.2(192.0.2.6)    Record Label     : 524287
-> 192.0.2.8(192.0.2.8)      Record Label     : 524286
-> 192.168.68.2              Record Label     : 524286
Last Resignal :
Attempted At : n/a              Resignal Reason  : n/a
Resignal Status: n/a           Reason           : n/a

Protected LSPs -
LSP Name     : LSP-PE-1-PE-10::empty_path
From         : 192.0.2.1         To               : 192.0.2.10
Avoid Node/Hop : 192.0.2.4     Downstream Label : 524287
Bandwidth    : 0 Kbps
=====

```

The LSP could be protected with one or more additional secondary paths, pre-sigaled or not, but this is outside the scope of this chapter.

When a link or node failure occurs along the LSP path, FRR protection kicks in and end-to-end path re-optimization is executed: a PATHERR message is forwarded to the head-end. Upon receiving the PATHERR message, the head-end calculates a new path.

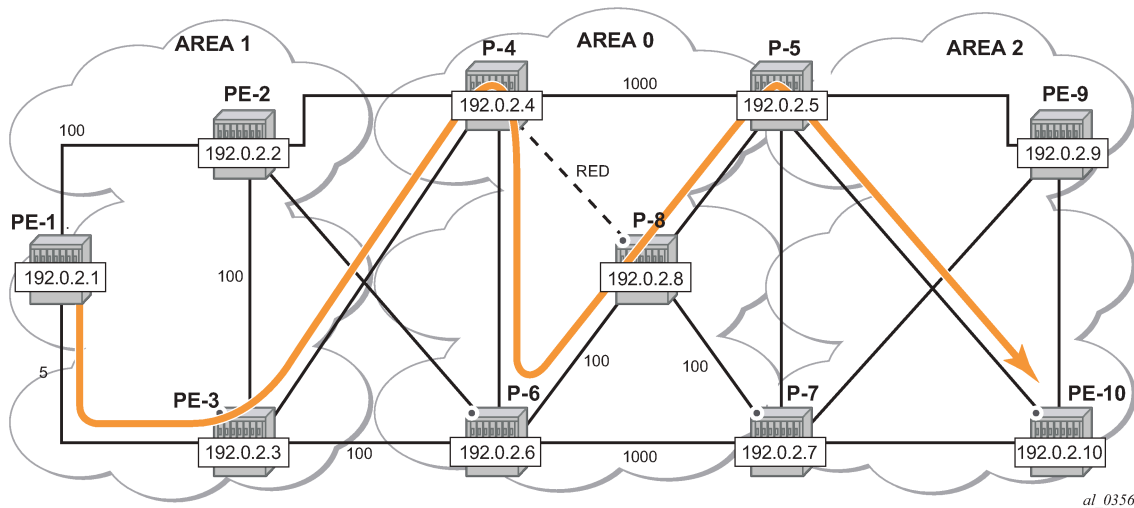
Admin groups

The use of administrative groups is described in the [RSVP Point-to-Point LSPs](#) chapter.

To support admin groups for inter-area LSPs, the ingress node PE-1 must propagate the admin groups within the Session Attribute object (SA) of the PATH message so that the ABRs along the path receive the admin group restrictions they have to take into account when further expanding the ERO in the PATH message.

In [Figure 215: Protection of all nodes/links along the LSP path](#) the LSP path avoids the link between P-4 and P-8. This is implemented by assigning admin group "red" to the link between P-4 and P-8 and then configuring the LSP to exclude the admin group "red".

Figure 216: Admin group example



Admin group configuration

On P-4, configure admin group "red" and assign a group value. In this example, group value 11 is used, but this can be any value between 0 and 31. Assign admin group "red" to the link to P-8.

This admin group configuration is required on P-4 and on iLER PE-1. However, it is good practice to configure the admin group on all the nodes.

```
# on all nodes:
configure {
  routing-options {
    if-attribute {
      admin-group "red" {
        value 11
      }
    }
  }
}
```

```
# on P-4:
configure {
  router "Base" {
    mpls {
      interface "int-P-4-P-8" {
        admin-group ["red"]
      }
    }
  }
}
```

On PE-1, change the LSP configuration as follows:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-10" {
        exclude-admin-group "red"
        propagate-admin-group true
      }
    }
  }
}
```

It is possible to have the same admin group constraint applied to the FRR bypass tunnels in the PLRs, but that is not the case here. The bypass tunnels ignore any admin group constraint. The **propagate-admin-group** command is required to include the admin group properties in the SA object of the PATH message. The admin group value is mapped to a 32-bitmap. In this example, value 11 means that the 12th bit is set, which means in binary 100000000000 or hex 0x800.

```
# on PE-1:
68 2021/05/12 08:29:57.687 UTC MINOR: DEBUG #2001 Base RSVP
"RSVP: PATH Msg
Send PATH From:192.0.2.1, To:192.0.2.10
      TTL:255, Checksum:0x5319, Flags:0x0
Session   - EndPt:192.0.2.10, TunnId:1, ExtTunnId:192.0.2.1
SessAttr  - Name:LSP-PE-1-PE-10::empty_path
          SetupPri:7, HoldPri:0, Flags:0x17
          CType:RA, ExcAny:0x800, IncAny:0x0, IncAll:0x0
RSVPHop   - CType:1, Addr:192.168.13.1, LIH:3
TimeValue - RefreshPeriod:30
SendTempl - Sender:192.0.2.1, LspId:24650
SendTSpec - CType:QOS, CDR:0.000 bps, PBS:0.000 bps, PDR:infinity
          MPU:20, MTU:1564
LabelReq  - IfType:General, L3ProtID:2048
RR0       - IpAddr:192.168.13.1, Flags:0x0
ERO       - IPv4Prefix 192.168.13.2/32, Strict
          IPv4Prefix 192.168.34.2/32, Strict
          IPv4Prefix 192.0.2.10/32, Loose
FRR0bj    - SetupPri:7, HoldPri:0, HopLimit:16, BW:0.000 bps, Flags:0x2
          ExcAny:0x0, IncAny:0x0, IncAll:0x0
"
```

The following two sets of output show that when P-4 expands the ERO it now excludes the link to node P-8 for the path calculation and the path is set up through P-6, P-8 and P-5.

```
# on P-4:
147 2021/05/12 08:29:57.693 UTC MINOR: DEBUG #2001 Base RSVP
"RSVP: PATH Msg
Send PATH From:192.0.2.1, To:192.0.2.10
      TTL:253, Checksum:0xc1d2, Flags:0x0
Session   - EndPt:192.0.2.10, TunnId:1, ExtTunnId:192.0.2.1
SessAttr  - Name:LSP-PE-1-PE-10::empty_path
          SetupPri:7, HoldPri:0, Flags:0x17
          CType:RA, ExcAny:0x800, IncAny:0x0, IncAll:0x0
RSVPHop   - CType:1, Addr:192.168.46.1, LIH:3
TimeValue - RefreshPeriod:30
SendTempl - Sender:192.0.2.1, LspId:24650
SendTSpec - CType:QOS, CDR:0.000 bps, PBS:0.000 bps, PDR:infinity
          MPU:20, MTU:1564
LabelReq  - IfType:General, L3ProtID:2048
RR0       - IpAddr:192.168.46.1, Flags:0x0
          IpAddr:192.168.34.1, Flags:0x0
          IpAddr:192.168.13.1, Flags:0x0
ERO       - IPv4Prefix 192.168.46.2/32, Strict
          IPv4Prefix 192.168.68.2/32, Strict
          IPv4Prefix 192.168.58.1/32, Strict
          IPv4Prefix 192.0.2.10/32, Loose
FRR0bj    - SetupPri:7, HoldPri:0, HopLimit:16, BW:0.000 bps, Flags:0x2
          ExcAny:0x0, IncAny:0x0, IncAll:0x0
"
```

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-10" path detail
```



```

=====
MPLS LSP LSP-PE-1-PE-10 Path (Detail)
=====
Legend :
@ - Detour Available          # - Detour In Use
b - Bandwidth Protected      n - Node Protected
s - Soft Preemption
S - Strict                   L - Loose
A - ABR                      + - Inherited
=====

LSP LSP-PE-1-PE-10
Path empty_path
-----
LSP Name      : LSP-PE-1-PE-10
From          : 192.0.2.1
To            : 192.0.2.10
Admin State   : Up
Oper State    : Up
Path Name     : empty_path
Path LSP ID   : 24650
Path Type     : Primary
Path Admin    : Up
Path Oper     : Up
Out Interface : 1/1/2
Out Label     : 524286
Path Up Time  : 0d 00:26:29
Path Down Time : 0d 00:00:00
Retry Limit   : 0
Retry Timer   : 30 sec
Retry Attempt : 0
Next Retry In : 0 sec

---snip---

Adspec        : Disabled
Oper Adspec    : Disabled
PathCompMethod : local-cspf
OperPathCompMethod : local-cspf
MetricType    : igp
Oper MetricType : igp
Least Fill    : Disabled
Oper LeastFill : Disabled
FRR           : Enabled
Oper FRR      : Enabled
FRR NodeProtect : Enabled
Oper FRR NP   : Enabled
FR Hop Limit  : 16
Oper FRHopLimit : 16
FR Prop Admin Gr* : Disabled
Oper FRPropAdmGrp : Disabled
Propagate Adm Grp : Enabled
Oper Prop Adm Grp : Enabled
Inter-area    : True

---snip---

Include Groups :
None
Oper IncludeGroups:
None
Exclude Groups :
red
Oper ExcludeGroups:
red

Adaptive      : Enabled
Oper Metric   : 15
Preference    : n/a
Path Trans    : 2
Failure Code   : noError
Failure Node  : n/a
Explicit Hops :
No Hops Specified
Actual Hops   :
192.168.13.1(192.0.2.1) @ n
-> 192.168.13.2(192.0.2.3) @ n
-> 192.168.34.2(192.0.2.4) @ n
-> 192.168.46.2 @ n
-> 192.0.2.8(192.0.2.8) @ n
-> 192.168.68.2 @ n
-> 192.0.2.5(192.0.2.5) @
-> 192.168.58.1 @
-> 192.0.2.10(192.0.2.10)
-> 192.168.105.2
Record Label   : N/A
Record Label   : 524286
Record Label   : 524285
Record Label   : 524286
Record Label   : 524285
Record Label   : 524285
Record Label   : 524285
Record Label   : 524285
Record Label   : 524284
Record Label   : 524284

```

```
Computed Hops      :
  192.168.13.1(S)
-> 192.168.13.2(S)
-> 192.168.34.2(SA)
-> 192.0.2.10(L)
Resignal Eligible: False
Last Resignal     : n/a                CSPF Metric      : 15
Last MBB         :
MBB Type          : ConfigChange        MBB State        : Success
Ended At          : 05/12/2021 08:29:59 Old Metric       : 15
Signaled BW       : 0 Mbps
Fail Code         : noError
=====
* indicates that the corresponding row element may have been truncated.
```

Shared Risk Link Groups (SRLG)

Shared risk link groups are described in chapter [Shared Risk Link Groups for RSVP-Based LSPs](#).

SRLGs are also supported in the context of inter-area TE LSPs. SRLGs refer to situations where links in a network share a common fiber (or a common physical attribute). If one link fails, other links in the group may fail as well. Links in the group have fate sharing.

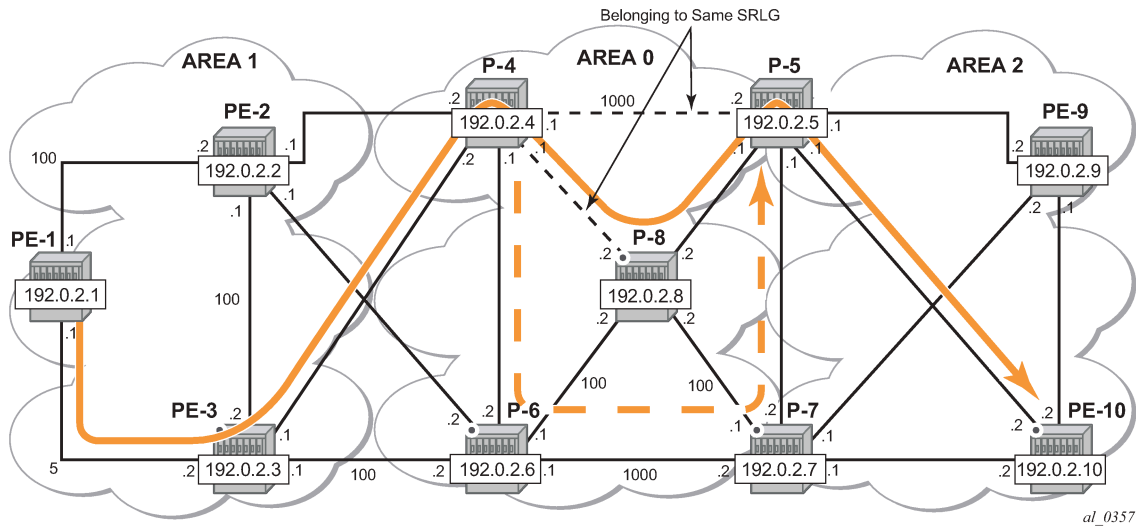
The MPLS TE SRLG feature enhances backup tunnel path selection so that a backup tunnel avoids using links that are in the same SRLG.

Consider the setup in [Figure 217: Share Risk Link Groups](#), where an inter-area LSP is set up from PE-1 to PE-10 and the path goes through P-8 because of a lower IGP metric. To protect against a node failure of P-8, P-4 (PLR) would normally set up an FRR backup directly to P-5 (MP), because of the lower IGP metric (P-4 to P-5:1000) compared to the IGP traffic via P-6 (P-4 to P-6 to P-7 to P-5:1020).

However, imagine that in this setup the link between P-4 and P-5 and the link between P-4 and P-8 are part of the same transmission bundle. In this case, a cut of that fiber bundle will bring down both the primary and the backup path.

This can be avoided by configuring these two links in the same SRLG group and enabling **srlg-frr strict** on P-4. In that case, the backup will be set up via P-6 as indicated by the dashed line in [Figure 217: Share Risk Link Groups](#).

Figure 217: Share Risk Link Groups



SRLG configuration

On P-4, an SRLG group is configured, `srlg-frr strict` is enabled and the links to P-5 and to P-8 are added to this SRLG group.

The SRLG group configuration is required on all nodes that use SRLG groups and on the ABR used by the inter-area TE LSP. In this example, it is configured on all nodes.



Note:

Enabling or disabling `srlg-frr` command only takes effect when the LSP primary path or the bypass path is resignaled. The user can either wait for the resignal timer to expire or cause the paths to be resignaled immediately by executing, at the ingress LER, the manual resignal command for the LSP primary path or for the bypass LSP path.

```
# on all nodes:
configure {
  routing-options {
    if-attribute {
      srlg-group "bundle-red" {
        value 1
      }
    }
  }
}
```

```
# on P-4:
configure {
  router "Base" {
    mpls {
      srlg-frr strict
      interface "int-P-4-P-5" {
        srlg-group "bundle-red" { }
      }
      interface "int-P-4-P-8" {
        srlg-group "bundle-red" { }
      }
    }
  }
}
```

Bounce RSVP to ensure that the srlg-frr command takes effect:

```
# on PE-1:
configure exclusive
router "Base" {
    rsvp {
        admin-state disable
        commit
        admin-state enable
        commit
    }
}
```

LSP configuration

Remove the admin group restriction from the LSP.

```
# on PE-1:
configure {
    router "Base" {
        mpls {
            lsp "LSP-PE-1-PE-10" {
                delete exclude-admin-group "red"
                propagate-admin-group false
            }
        }
    }
}
```

Now check the LSP path on PE-1 and verify that FRR protection is in place.

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-10" path detail

=====
MPLS LSP LSP-PE-1-PE-10 Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use
  b - Bandwidth Protected      n - Node Protected
  s - Soft Preemption
  S - Strict                    L - Loose
  A - ABR                       + - Inherited
=====
-----
LSP LSP-PE-1-PE-10
Path empty_path
-----
LSP Name      : LSP-PE-1-PE-10
From          : 192.0.2.1
To            : 192.0.2.10
Admin State   : Up
Path Name     : empty_path
Path LSP ID   : 24652
Path Admin    : Up
Out Interface : 1/1/2
Path Up Time  : 0d 00:31:05
Retry Limit   : 0
Retry Attempt : 0
Oper State    : Up
Path Type     : Primary
Path Oper     : Up
Out Label     : 524287
Path Down Time : 0d 00:00:00
Retry Timer   : 30 sec
Next Retry In : 0 sec

---snip---

Adspec        : Disabled
PathCompMethod : local-cspf
MetricType    : igp
Oper Adspec    : Disabled
OperPathCompMethod : local-cspf
Oper MetricType : igp
```

```

Least Fill      : Disabled          Oper LeastFill   : Disabled
FRR             : Enabled           Oper FRR         : Enabled
FRR NodeProtect : Enabled           Oper FRR NP      : Enabled
FR Hop Limit    : 16                Oper FRHopLimit  : 16
FR Prop Admin Gr*: Disabled         Oper FRPropAdmGrp : Disabled
Propagate Adm Grp: Disabled         Oper Prop Adm Grp : Disabled
Inter-area      : True

---snip---

Include Groups  :                   Oper IncludeGroups:
None                                                    None
Exclude Groups  :                   Oper ExcludeGroups:
None                                                    None

Adaptive        : Enabled           Oper Metric       : 15
Preference      : n/a
Path Trans      : 3                 CSPF Queries      : 39
Failure Code    : noError
Failure Node    : n/a
Explicit Hops   :
  No Hops Specified
Actual Hops     :
  192.168.13.1(192.0.2.1) @ n      Record Label      : N/A
-> 192.168.13.2(192.0.2.3) @ n      Record Label      : 524287
-> 192.168.34.2(192.0.2.4) @ n      Record Label      : 524287
-> 192.0.2.8(192.0.2.8) @ n         Record Label      : 524287
-> 192.168.48.2 @ n                 Record Label      : 524287
-> 192.0.2.5(192.0.2.5) @          Record Label      : 524287
-> 192.168.58.1 @                   Record Label      : 524287
-> 192.0.2.10(192.0.2.10)          Record Label      : 524287
-> 192.168.105.2                    Record Label      : 524287
Computed Hops   :
  192.168.13.1(S)
-> 192.168.13.2(S)
-> 192.168.34.2(SA)
-> 192.0.2.10(L)
Resignal Eligible: False
Last Resignal   : n/a              CSPF Metric       : 15
Last MBB        :
  MBB Type      : ConfigChange      MBB State         : Success
  Ended At      : 05/12/2021 08:37:49 Old Metric        : 15
  Signaled BW   : 0 Mbps
  Fail Code     : noError
=====
* indicates that the corresponding row element may have been truncated.

```

On P-4, the SRLG configuration is checked as follows:

```

[/]
A:admin@P-4# show router if-attribute srlg-group

=====
Interface Srlg Groups
=====
Group Name          Group Value    Penalty Weight
-----
bundle-red          1              0
-----
No. of Groups: 1
=====

[/]

```

```
A:admin@P-4# show router mpls interface

=====
MPLS Interfaces
=====
Interface                               Port-id           Adm  Opr(V4/V6)  TE-
metric
-----
system                                  system            Up   Up/Down     None
  Admin Groups                          None
  SRLG Groups                            None
int-P-4-P-5                             1/1/1            Up   Up/Down     None
  Admin Groups                            None
  SRLG Groups                           bundle-red
int-P-4-P-6                             1/1/3            Up   Up/Down     None
  Admin Groups                            None
  SRLG Groups                            None
int-P-4-P-8                             1/2/1            Up   Up/Down     None
  Admin Groups                            red
  SRLG Groups                           bundle-red
int-P-4-PE-2                             1/1/2            Up   Up/Down     None
  Admin Groups                            None
  SRLG Groups                            None
int-P-4-PE-3                             1/1/4            Up   Up/Down     None
  Admin Groups                            None
  SRLG Groups                            None
-----
Interfaces : 6
=====
```

On PE-4, it is verified that the bypass tunnel is set up via P-6 rather than via P-5, as follows:

```
[/]
A:admin@P-4# show router mpls bypass-tunnel protected-lsp detail

=====
MPLS Bypass Tunnels (Detail)
=====
-----
bypass-node192.0.2.8-61443
-----
To          : 192.168.57.1          State          : Up
Out I/F     : 1/1/3                   Out Label     : 524285
Up Time     : 0d 00:03:12         Active Time    : n/a
Reserved BW : 0 Kbps              Protected LSP Count : 1
Type        : Dynamic             Bypass Path Cost : 1020
Setup Priority : 7                 Hold Priority   : 0
Class Type  : 0
Exclude Node : None                Inter-Area     : False
Computed Hops :
  192.168.46.1(S)                 Egress Admin Groups : None
  -> 192.168.46.2(S)              Egress Admin Groups : None
  -> 192.168.67.2(S)              Egress Admin Groups : None
  -> 192.168.57.1(S)              Egress Admin Groups : None
Actual Hops :
  192.168.46.1(192.0.2.4)         Record Label    : N/A
  -> 192.168.46.2(192.0.2.6)     Record Label    : 524285
  -> 192.168.67.2(192.0.2.7)     Record Label    : 524287
  -> 192.168.57.1(192.0.2.5)     Record Label    : 524286
Last Resignal :
Attempted At : n/a                Resignal Reason : n/a
Resignal Status: n/a              Reason          : n/a
```

```
Protected LSPs -  
LSP Name      : LSP-PE-1-PE-10::empty_path  
From          : 192.0.2.1                To          : 192.0.2.10  
Avoid Node/Hop : 192.0.2.8              Downstream Label : 524287  
Bandwidth     : 0 Kbps
```

=====

Conclusion

Inter-area TE P2P LSPs can be set up based on ERO expansion. With this feature, the head-end does a partial CSPF calculation to its local ABR. On receiving a PATH message with a loose hop ERO, this ABR does a partial or full CSPF calculation to the next ABR to reach the final destination.

FRR protection within the area is available. FRR node protection of the ABR is possible through an MBT on the PLR (node just upstream of the ABR) to the MP (node just downstream of the ABR) or through a dynamically signaled bypass tunnel on the PLR. Dynamic ABR node protection requires that the node ID of the MP node is propagated in the RESV message and that an XRO object is included in the bypass PATH message which makes it possible for the ABR to calculate a path to an MP node.

TE features such as BW, path prioritization, path pre-emption, and graceful shutdown are supported, as well as propagation of the session attribute with affinity along the LSP path (admin groups) and SRLG.

LDP FEC to BGP Label Route Stitching

This chapter provides information about LDP FEC to BGP label route stitching.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter is applicable to SR OS routers and was initially written for SR OS Release 13.0.R7. The MD-CLI in the current edition corresponds to SR OS Release 21.2.R1. Label Distribution Protocol (LDP) Forwarding Equivalence Class (FEC) to Border Gateway Protocol (BGP) label route stitching was first implemented in SR OS Release 8.0.

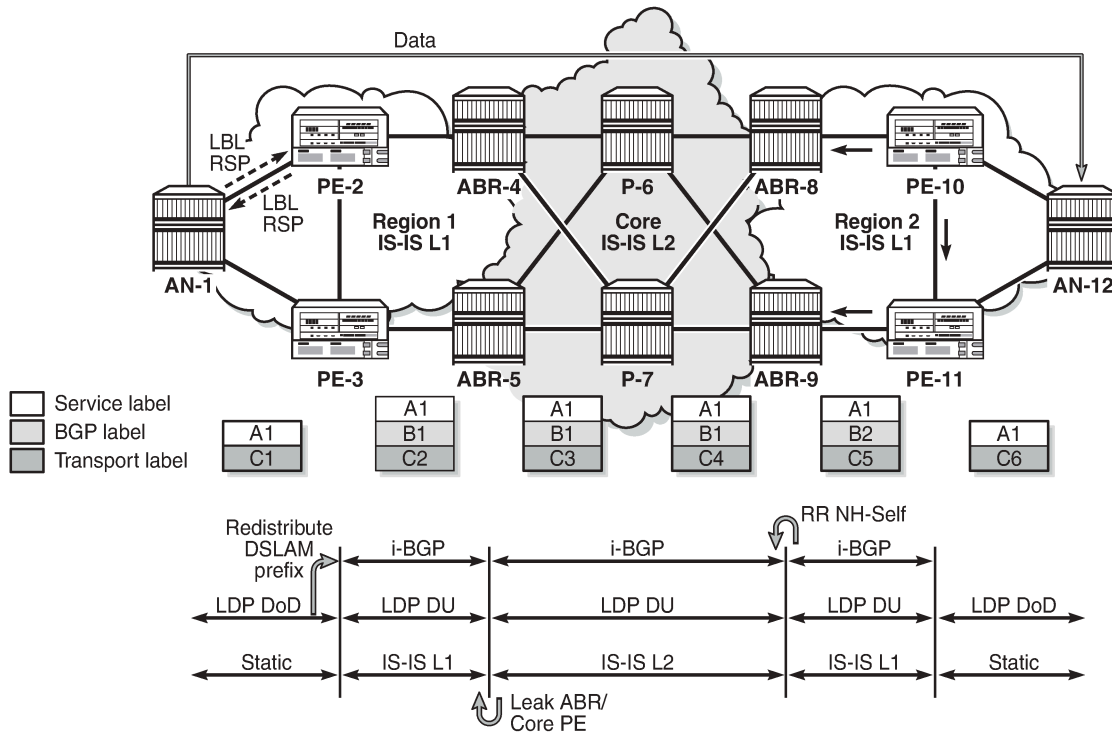
Overview

Stitching of an LDP FEC to a BGP labeled route allows LDP-capable PE devices, such as Digital Subscriber Line Access Multiplexers (DSLAMs), to offer services to LDP-capable PE devices in other areas or domains without the need to support BGP labeled routes. This feature is used in a large network to provide services across multiple areas or Autonomous Systems (ASs).

When BGP is used to distribute a particular route, it can at the same time be used to distribute a Multi-Protocol Label Switching (MPLS) label that is mapped to that route. The label mapping information for a particular route is appended to the same BGP update message that is used to distribute the route. This is described in RFC 3701, *Carrying Label Information in BGPv4*.

[Figure 218: LDP FEC to BGP label route stitching](#) shows a network with a core area and regional areas. The components of the network are defined in the paragraphs that follow. For simplification, the control plane is displayed from right to left and the data plane from left to right.

Figure 218: LDP FEC to BGP label route stitching



25614

The Access Nodes (ANs) are DSLAMs that support LDP. In seamless MPLS networks, LDP Downstream-on-Demand (DoD) label advertisement can be used between the ANs and their next-hop PEs. Usually, MPLS routers implement LDP Downstream Unsolicited (DU) label distribution, advertising MPLS labels for all routes in their Routing Information Base (RIB). The ANs do not need to have LDP bindings for all prefixes in the network. The ANs will request the LDP labels they need. LDP DoD improves scalability in large networks.

BGP Route Reflectors (RRs) can be used to improve scalability. The RR can be any node; it does not need to be an Area Border Router (ABR) as in [Figure 218: LDP FEC to BGP label route stitching](#). If the RR is not in the forwarding path, it does not need to be capable of forwarding MPLS packets.

There are different areas for IS-IS: routers in the core network have level 2 (L2) capability, whereas the routers in the regional areas have level 1 (L1) capability and the ABRs have both. In each ABR, an IS-IS export policy is defined to leak the routes from the core to the regional networks.

Passing L1 routes (regional) into L2 (core) is inherent to IS-IS and cannot be controlled through policy. Passing L2 routes to L1 can be controlled through policy.

Only nodes within a regional area, and the ABR nodes in the same area, exchange LDP FECs. PE routers in a regional area learn the reachability of PE routers in other regional areas by way of RFC 3107 BGP labeled routes redistributed by the remote nodes.

The label stack contains three labels for packets sent in an Epipe service between the access nodes:

- The DSLAMs push a service label to the packets sent in the Epipe service. The service label remains unchanged end-to-end between the DSLAMs. The service label is popped by the remote DSLAM and is the inner label of the label stack.

- The BGP label is the middle label of the label stack and should be regarded as a transport label. The transport label stack contains two labels: BGP and LDP transport label. BGP labeled routes are not supported on the DSLAMs. The BGP label is pushed by the PE nearest to the local DSLAM and is swapped at the BGP next hop, which can be a BGP peer configured with next-hop-self or the PE that is the remote endpoint of the BGP tunnel. The BGP label is popped by the PE at the end of the BGP tunnel.
- The DSLAMs push an LDP transport label to the packets sent to the remote DSLAM. At the PE nearest to the local DSLAM, the LDP transport label is stitched to the BGP label. At the same time, that same PE pushes the LDP transport label to reach the BGP next hop. The LDP transport label is swapped in every Label Switching Router (LSR) and popped by the PE nearest to the remote DSLAM. That PE also pops the BGP label, which is stitched to the LDP transport label that is pushed to the packets sent to the remote DSLAM. This LDP label is the top label of the label stack.

When PE-2 is an ingress Label Edge Router (iLER) sending a service packet to the remote PE, PE-2 inserts the BGP route label to reach the remote PE and an LDP label to reach the next-hop router. In [Figure 218: LDP FEC to BGP label route stitching](#), this is the remote ABR because it has set next-hop-self (NH-Self).

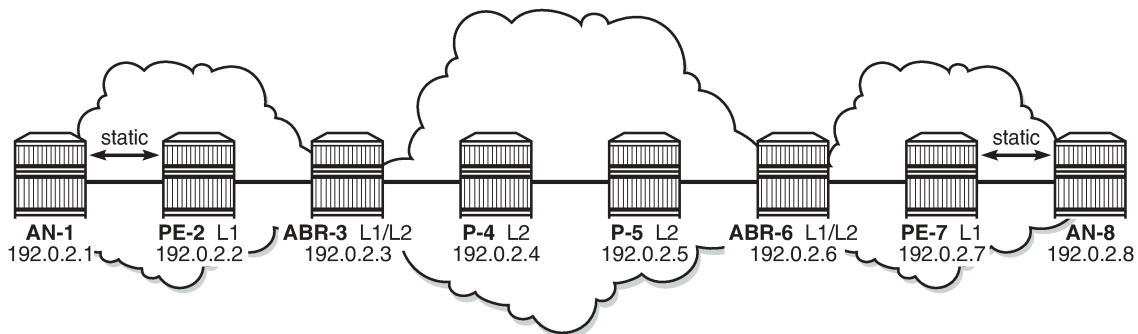
The access node AN-1, which is a DSLAM, can behave as a PE router for Epipe services. It will need to establish a pseudowire (PW) to a PE in a different regional area via LSR PE-2. In this case, PE-2 performs the following actions:

- Translates the LDP FEC it learned from AN-1 into a BGP labeled route and redistributes it using iBGP within its area. This is in addition to redistributing the FEC to its LDP neighbors in the same area.
- Translates the BGP labeled routes it learns through iBGP into an LDP FEC and redistributes it to its LDP neighbors in the same area. AN-1 requests the LDP FEC of the remote DSLAM (AN-12) using LDP DoD.
- When a data packet is received from AN-1 with destination AN-12, PE-2 swaps the LDP label into a BGP label and pushes the LDP label to reach the BGP next hop. When a data packet with destination AN-1 is received on PE-2 from the local ABR (ABR-4), the top transport label (LDP) is removed and the BGP label is swapped for the LDP label corresponding to AN-1.

Configuration

[Figure 219: Example topology](#) shows the example topology that is used in this section. An Epipe will be established between the access nodes AN-1 and AN-8. PE-2 and PE-7 will stitch the LDP FECs to BGP label routes. In the regional areas, IS-IS L1 capability is used whereas in the core area, IS-IS L2 is used. The ABR nodes support both IS-IS L1 and L2 and export routes from L2 to L1. Static routes are configured between the access nodes and the next-hop PEs.

Figure 219: Example topology



25615

Initial configuration



Note:

In the example topology, all nodes are 7750 SRs, while the ANs should be access devices, such as DSLAMs. The limitation of this approach is that the ANs (SRs) in this setup can only request a label for the directly connected PE and not for their remote peer AN; however, DSLAMs do not have this limitation. Consequently, the Epipe service in this configuration will be operationally down because the transport tunnel is down.

All nodes have the following initial configuration:

- Cards, media dependent adapters (MDAs), ports
- Router interfaces



Note:

The IP addresses for the link between node A and node B are in subnet 192.168.AB.0/0. The node with the lowest ID has IP address 192.168.AB.1/30 and the node with the highest ID has IP address 192.168.AB.2/30.

```
# on PE-2:
configure {
  router "Base" {
    interface "int-PE-2-ABR-3" {
      port 1/1/1
      ipv4 {
        primary {
          address 192.168.23.1
          prefix-length 30
        }
      }
    }
  }
  interface "int-PE-2-AN-1" {
    port 1/1/2
    ipv4 {
      primary {
        address 192.168.12.2
        prefix-length 30
      }
    }
  }
}
```

```
    }
    interface "system" {
      ipv4 {
        primary {
          address 192.0.2.2
          prefix-length 32
        }
      }
    }
  }
}
```

- Static routes are configured between AN-1 and PE-2 and between PE-7 and AN-8:

```
# on AN-1:
configure {
  router "Base" {
    static-routes {
      route 0.0.0.0/0 route-type unicast {
        next-hop "192.168.12.2" {
          admin-state enable
        }
      }
    }
  }
}
```

```
# on PE-2:
configure {
  router "Base" {
    static-routes {
      route 192.0.2.1/32 route-type unicast {
        next-hop "192.168.12.1" {
          admin-state enable
        }
      }
    }
  }
}
```

- IS-IS (alternatively, OSPF could have been used)
 - PE-2 and PE-7 have L1 capability.

```
# on PE-2:
configure {
  router "Base" {
    isis 0 {
      admin-state enable
      level-capability 1
      area-address [49.0001]
      interface "int-PE-2-ABR-3" {
        interface-type point-to-point
      }
      interface "system" {
      }
    }
  }
}
```

- P-4 and P-5 have L2 capability.

- ABR-3 and ABR-6 have L1 capability on the interfaces toward the PE routers in the regional areas and L2 capability on the interfaces toward the P routers in the core area. A policy is applied to export the system IP addresses from L2 to L1:

```
# on ABR-3:
configure {
  router "Base" {
    isis 0 {
      admin-state enable
      export-policy ["export_L2_to_L1_policy"]
      area-address [49.0001]
      interface "int-ABR-3-P-4" {
        interface-type point-to-point
        level-capability 2
      }
      interface "int-ABR-3-PE-2" {
        interface-type point-to-point
        level-capability 1
      }
      interface "system" {
      }
    }
  }
}

configure {
  policy-options {
    prefix-list "system_IP_prefixes" {
      prefix 192.0.2.0/24 type longer {
      }
    }
  }
  policy-statement "export_L2_to_L1_policy" {
    entry 10 {
      from {
        level 2
        prefix-list ["system_IP_prefixes"]
        protocol {
          name [isis]
        }
      }
      action {
        action-type accept
      }
    }
  }
}
}
```

- LDP
 - Link LDP is enabled on all router interfaces on all nodes, including the ANs.
 - On PE-2 and PE-7, DoD is enabled in the session parameters for the peering sessions with the ANs:

```
# on PE-2:
configure {
  router "Base" {
    ldp {
      session-parameters {
        peer 192.0.2.1 {
          dod-label-distribution true
        }
      }
    }
  }
}
```

```

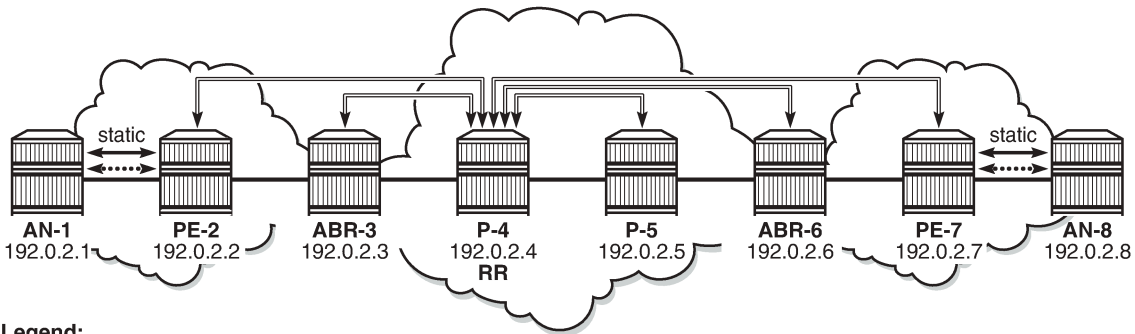
    }
    interface-parameters {
      interface "int-PE-2-ABR-3" {
        ipv4 {
        }
      }
      interface "int-PE-2-AN-1" {
        ipv4 {
        }
      }
    }
  }
}

```

Configure BGP

BGP is configured on all nodes except the ANs. [Figure 220: BGP enabled with P-4 as RR](#) shows that P-4 is the RR.

Figure 220: BGP enabled with P-4 as RR



Legend:
 ↔ iBGP
 ↔ LDP DoD

25616

The initial BGP configuration on PE-2 is the following:

```

# on PE-2:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      ebgp-default-reject-policy {
        import false
        export false
      }
      group "internal_group" {
        peer-as 64496
      }
      neighbor "192.0.2.4" {
        group "internal_group"
      }
    }
  }
}

```

```
}

```

The configuration is identical for ABR-3, P-5, ABR-6, and PE-7. The initial BGP configuration on the RR P-4 is:

```
# on P-4:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      cluster {
        cluster-id 1.1.1.1
      }
      ebgp-default-reject-policy {
        import false
        export false
      }
      group "internal_group" {
        peer-as 64496
      }
      neighbor "192.0.2.2" {
        group "internal_group"
      }
      neighbor "192.0.2.3" {
        group "internal_group"
      }
      neighbor "192.0.2.5" {
        group "internal_group"
      }
      neighbor "192.0.2.6" {
        group "internal_group"
      }
      neighbor "192.0.2.7" {
        group "internal_group"
      }
    }
  }
}
```

This BGP configuration is incomplete: for labeled IPv4 BGP peering sessions, an additional address family will be configured on PE-2 and PE-7, as well as on RR P-4 for neighbors PE-2 and PE-7. The configuration is shown in the following section. The prefixes for AN-1 and AN-8 will be advertised in the labeled IPv4 BGP sessions only, not in IPv4 BGP sessions.

Export policies for BGP and LDP

LDP FEC to BGP label route stitching is established by configuring separate tunnel table route export policies in both protocols. At the local next-hop PE, the LDP FEC of the local AN must be translated into a BGP label and at the remote PE, the BGP label must be translated into an LDP FEC.

An export policy for the export from LDP to BGP must be defined on the PE nodes.

```
# on PE-2:
configure {
  policy-options {
    prefix-list "local_AN_prefixes" {
      prefix 192.0.2.1/32 type exact {
      }
    }
  }
}
```

```
prefix-list "remote_AN_prefixes" {
  prefix 192.0.2.8/32 type exact {
  }
}
policy-statement "export_BGP_policy" {
  entry 10 {
    from {
      prefix-list ["local_AN_prefixes"]
      protocol {
        name [ldp]
      }
    }
    action {
      action-type accept
    }
  }
}
}
```

On PE-7, the policy statement is identical, but the prefix list is different.

This export policy must be applied in the **bgp** context: either in the general settings or per **group** or per **neighbor**.

```
# on PE-2:
configure {
  router "Base" {
    bgp {
      group "internal_group" {
        export {
          policy ["export_BGP_policy"]
        }
      }
    }
  }
}
```

In a similar way, BGP labels must be exported to LDP on the PE routers. The export policy is configured as follows, with a different prefix list:

```
# on PE-2:
configure {
  policy-options {
    prefix-list "remote_AN_prefixes" {
      prefix 192.0.2.8/32 type exact {
      }
    }
  }
  policy-statement "export_LDP_policy" {
    entry 10 {
      from {
        prefix-list ["remote_AN_prefixes"]
        protocol {
          name [bgp-label]
        }
      }
      action {
        action-type accept
      }
    }
  }
}
```



```
}
```

This export policy is applied in the **ldp** context, as follows:

```
# on PE-2:
configure {
  router "Base" {
    ldp {
      export-tunnel-table ["export_LDP_policy"]
    }
  }
}
```

Advertise labels in BGP updates

BGP should evaluate the activated /32 LDP prefixes in the export policy. This needs to be configured on the endpoints of the BGP tunnel on PE-2 and PE-7, as follows:

```
# on PE-2 and PE-7:
configure {
  router "Base" {
    bgp {
      neighbor "192.0.2.4" {
        advertise-ldp-prefix true
        family {
          label-ipv4 true
        }
      }
    }
  }
}
```

On RR P-4, the family **label-ipv4** is enabled and the LDP prefix is advertised toward the clients PE-2 and PE-7, as follows.

```
# on RR P-4:
configure {
  router "Base" {
    bgp {
      neighbor "192.0.2.2" {
        advertise-ldp-prefix true
        family {
          label-ipv4 true
        }
      }
      neighbor "192.0.2.7" {
        advertise-ldp-prefix true
        family {
          label-ipv4 true
        }
      }
    }
  }
}
```

Configuring address family **label-ipv4 true** and **advertise-ldp-prefix true** implies that all activated /32 LDP FEC prefixes will be sent to the remote BGP peer as an RFC 3107 formatted label.

Configuring address family **label-ipv4 true** and **advertise-ldp-prefix false** or without `advertise-ldp-prefix` implies that only core IPv4 routes learned from the Route Table Manager (RTM) are advertised as RFC 3107 BGP labeled routes to this neighbor. No stitching of LDP FEC to the BGP labeled route will be performed for this neighbor, even if the same prefix was learned from LDP.

The BGP open messages contain address family AFI=1 and SAFI=1 between the RR and peers for address family IPv4, that is used for IPv4 unicast. See `Cap_Code MP-BGP`. Bytes 0x0 0x1 (AFI=1) 0x0 0x1 (SAFI=1).

```
# on ABR-3 (debugging is enabled in classic CLI):
A:ABR-3# show debug
debug
  router "Base"
  bgp
    open
    update
  exit
exit
exit
```

```
[/]
A:admin@ABR-3# show log log-id 2

=====
Event Log 2 log-name 2
=====
---snip---
2 2021/08/09 13:47:46.475 CEST MINOR: DEBUG #2001 Base BGP
"BGP: OPEN
Peer 1: 192.0.2.4 - Received BGP OPEN: Version 4
AS Num 64496: Holdtime 90: BGP_ID 192.0.2.4: Opt Length 20 (ExtOpt F)
Opt Para: Type CAPABILITY: Length = 18: Data:
  Cap_Code GRACEFUL-RESTART: Length 2
  Bytes: 0x0 0x78
  Cap_Code MP-BGP: Length 4
  Bytes: 0x0 0x1 0x0 0x1
  Cap_Code ROUTE-REFRESH: Length 0
  Cap_Code 4-OCTET-ASN: Length 4
  Bytes: 0x0 0x0 0xfb 0xf0
"
---snip---
```

Between peers that advertise the labels, AFI=1 and SAFI=4, the address family is labeled IPv4 unicast. The following BGP open message is seen on PE-2:

```
[/]
A:admin@PE-2# show log log-id 2

=====
Event Log 2 log-name 2
=====
Description : (Not Specified)
Memory Log contents [size=100 next event=11 (not wrapped)]

10 2021/08/09 13:51:13.041 CEST MINOR: DEBUG #2001 Base BGP
"BGP: OPEN
Peer 1: 192.0.2.4 - Received BGP OPEN: Version 4
AS Num 64496: Holdtime 90: BGP_ID 192.0.2.4: Opt Length 20 (ExtOpt F)
Opt Para: Type CAPABILITY: Length = 18: Data:
  Cap_Code GRACEFUL-RESTART: Length 2
  Bytes: 0x0 0x78
```

```

Cap_Code MP-BGP: Length 4
Bytes: 0x0 0x1 0x0 0x4
Cap_Code ROUTE-REFRESH: Length 0
Cap_Code 4-OCTET-ASN: Length 4
  Bytes: 0x0 0x0 0xfb 0xf0
"
---snip---

```

No BGP update messages are sent to ABR-3. Prefix 192.0.2.8 is advertised as a labeled IPv4 route from PE-7 to P-4 and forwarded by P-4 to its other labeled IPv4 client, PE-2, but it is not sent to BGP IPv4 clients, such as ABR-3.

The BGP update messages between labeled IPv4 peers contain label information, for example, for prefix 192.0.2.8/32. The address family is LBL-IPV4 (IPV4-Labeled) and the label is 524280. The following BGP update for prefix 192.0.2.8/32 is received on PE-2:

```

[/]
A:admin@PE-2# show log log-id 2

=====
Event Log 2 log-name 2
=====
Description : (Not Specified)
Memory Log contents [size=100  next event=14  (not wrapped)]

12 2021/08/09 13:52:10.511 CEST MINOR: DEBUG #2001 Base Peer 1: 192.0.2.4
"Peer 1: 192.0.2.4: UPDATE
Peer 1: 192.0.2.4 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 56
  Flag: 0x90 Type: 14 Len: 17 Multiprotocol Reachable NLRI:
    Address Family LBL-IPV4
    NextHop len 4 NextHop 192.0.2.7
    192.0.2.8/32 Label 524280
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 0 AS Path:
  Flag: 0x80 Type: 4 Len: 4 MED: 1
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
  Flag: 0x80 Type: 9 Len: 4 Originator ID: 192.0.2.7
  Flag: 0x80 Type: 10 Len: 4 Cluster ID:
    1.1.1.1
"
---snip---

```

After applying the export policy from BGP to LDP, enabling the address family labeled IPv4 in BGP, and advertising labels for the LDP FEC prefixes, LDP will look for BGP route entries in the tunnel table. If a /32 BGP labeled route matches a prefix entry in the export policy, LDP originates an LDP FEC for this prefix, stitches it to the BGP labeled route, and redistributes the LDP FEC to its BGP neighbors. This can be shown on PE-7, as follows.

```

[/]
A:admin@PE-7# show router bgp inter-as-label

=====
BGP Inter-AS labels
Flags: B - entry has backup, P - entry is promoted
=====
NextHop                Received      Advertised    Label
                        Label         Label         Origin
-----
192.0.2.8              524287       524280       InternalLdp

```

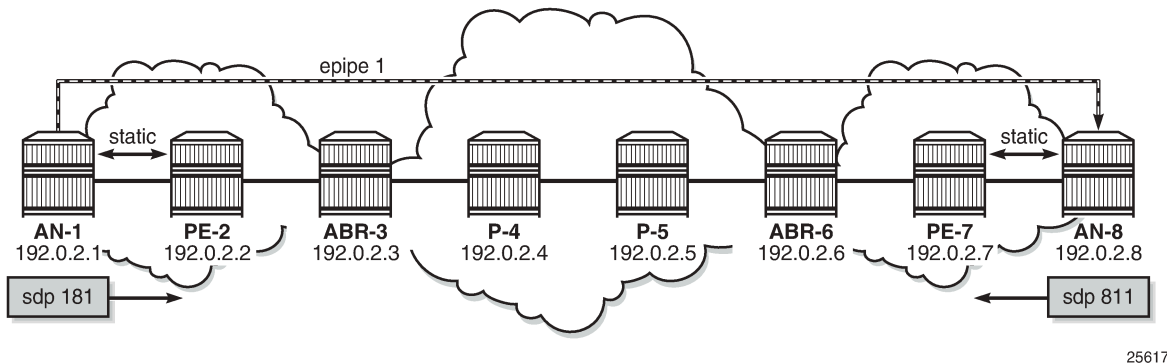
```
-----
Total Labels allocated: 1
-----
```

The label received from AN-8 is 524287. The label origin is **InternalLdp**. This LDP label is stitched to BGP label 524280 that will be advertised by PE-7 to its BGP labeled IPv4 peers: PE-7 advertises to RR P-4 and P-4 advertises this route to PE-2. Traffic sent from AN-1 toward AN-8 will be forwarded from PE-2 to its BGP NH PE-7 using BGP label 524280. In PE-7, the BGP label is stitched to LDP label 524287 that will be used to forward the packet to AN-8.

Configure SDP and Epipe

An end-to-end Epipe service is established between AN-1 and AN-8, as shown in [Figure 221: End-to-end Epipe service](#).

Figure 221: End-to-end Epipe service



Note:

In this setup, ANs are simulated by 7750 SRs. Due to this limitation, the SDP used by the Epipe service will not become operational. 7750 SR only supports single-hop DoD, which implies that AN-1 can only request a label for the LSR ID of the directly connected router, PE-2, not of remote nodes, such as AN-8. Similarly, AN-8 cannot request a label for AN-1. Therefore, it is not possible to have an LDP LSP between the ANs and the SDP will be down because there is no transport tunnel.

The SDP is configured on AN-1, as follows:

```
# on AN-1:
configure {
  service {
    sdp 181 {
      admin-state enable
      delivery-type mpls
      ldp true
      far-end {
        ip-address 192.0.2.8
      }
    }
  }
}
```

An Epipe is configured on AN-1, as follows:

```
# on AN-1:
configure {
  service {
    epipe "Epipe_1_name" {
      admin-state enable
      service-id 1
      customer "1"
      spoke-sdp 181:1 {
      }
      sap 1/2/1:1 {
      }
    }
  }
}
```

The configuration of the SDP and the Epipe on AN-8 is similar.

The SDP is down because there is no transport tunnel, which can be shown as follows:

```
[/]
A:admin@AN-1# show service sdp detail

=====
Services: Service Destination Points Details
=====
-----
Sdp Id 181 -192.0.2.8
-----
Description          : (Not Specified)
SDP Id               : 181
Admin Path MTU       : 0
Delivery             : MPLS
Far End              : 192.0.2.8
Oper Tunnel Far End  : 192.0.2.8
LSP Types           : LDP

SDP Source           : manual
Oper Path MTU       : 0
Tunnel Far End      :

Admin State          : Up
Signaling            : TLDP
Acct. Pol            : None
Oper State           : Down
Metric               : 0
Collect Stats        : Disabled
---snip---
Flags                : TranspTunnDown
---snip---
-----
Number of SDPs : 1
-----
=====
```

A targeted LDP session is established between AN-1 and AN-8, which can be shown as follows:

```
[/]
A:admin@AN-1# show router ldp session ipv4

=====
LDP IPv4 Sessions
=====
-----
Peer LDP Id      Adj Type  State      Msg Sent  Msg Recv  Up Time
-----
192.0.2.2:0     Link     Established 478       484       0d 00:20:59
192.0.2.8:0     Targeted Established 10        11        0d 00:00:23
-----
```

No. of IPv4 Sessions: 2
=====

LDP FEC resolution at PE-2 for traffic from AN-8 to AN-1

The following steps occur at PE-2 for the LDP FEC resolution for traffic from AN-1 toward AN-8. The situation is similar for PE-7.

1. After receiving an LDP label binding message for LDP FEC for the system address of AN-1 (192.0.2.1/32), PE-2 installs this prefix in the Label Forwarding Information Base (LFIB). PE-2 programs a push and a swap Next Hop Label Forwarding Entry (NHLFE) in the egress data path to forward packets to prefix 192.0.2.1/32.



Note:

PE-2 installs this LDP FEC in the LFIB only if there is an exact match of the prefix 192.0.2.1/32 in the routing table or a longest match of the prefix in the routing table, in case aggregate-prefix-match is configured on PE-2. The advertising LDP neighbor (AN-1) must be the next hop to reach the FEC prefix.

```
[/]
A:admin@PE-2# show router ldp bindings active prefixes prefix 192.0.2.1/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
              (IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static           (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop    (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix          Op
IngLbl          EgrLbl
EgrNextHop      EgrIf/LspId
-----
192.0.2.1/32    Push
--             524287
192.168.12.1    1/1/2

192.0.2.1/32    Swap
524286          524287
192.168.12.1    1/1/2

-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

2. PE-2 programs a tunnel entry for prefix 192.0.2.1/32 in the tunnel table.

```
[/]
A:admin@PE-2# show router tunnel-table 192.0.2.1/32

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner      Encap TunnelId Pref  Nexthop      Metric
  Color
-----
192.0.2.1/32         ldp        MPLS  65537    9    192.168.12.1  1
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
```

3. PE-2 advertises a new FEC label binding for prefix 192.0.2.1/32 toward all its LDP neighbors. The result can be shown on ABR-3, as follows:

```
[/]
A:admin@ABR-3# show router ldp bindings prefixes prefix 192.0.2.1/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.3)
              (IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP IPv4 Prefix Bindings
=====
Prefix
Peer          FEC-Flags
IgrLbl        EgrLbl
EgrNextHop    EgrIntf/LspId
-----
192.0.2.1/32
192.0.2.2:0
  --          524286
  --          --
-----
No. of IPv4 Prefix Bindings: 1
=====
```

4. When BGP learns the LDP FEC via the tunnel table and the FEC prefix exists in the BGP route policy, PE-2 originates a BGP labeled route toward all its neighbors that have the advertise label for LDP FEC prefixes enabled. The following output shows the BGP labeled route to RR P-4 for prefix 192.0.2.1/32.

```
[/]
A:admin@PE-2# show router bgp routes label-ipv4 hunt

=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
```

```

Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
BGP Routes
=====
-----
RIB In Entries
-----
---snip---
-----
RIB Out Entries
-----
Network       : 192.0.2.1/32
NextHop       : 192.0.2.2
Path Id       : None
To            : 192.0.2.4
Res. NextHop  : n/a
Local Pref.   : 100
Aggregator AS : None
Atomic Aggr.  : Not Atomic
AIGP Metric   : None
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None
IPv4 Label    : 524280
Lbl Allocation : NEXT-HOP
Origin        : IGP
AS-Path       : No As-Path
Route Tag     : 0
Neighbor-AS   : n/a
Orig Validation: NotFound
Source Class  : 0

Interface Name : NotAvailable
Aggregator     : None
MED            : 1
IGP Cost       : n/a

Peer Router Id : 192.0.2.4
Label Type     : SWAP

Dest Class    : 0

-----
Routes : 3
=====

```

BGP labeled route resolution at PE-2 for traffic from AN-1 to AN-8

The following steps occur at PE-2 for the BGP labeled route resolution for traffic from AN-1 toward AN-8. The situation is similar for PE-7.

1. When there is an LDP LSP to the BGP neighbor advertising the route (PE-7) and PE-2 has received a BGP labeled route via iBGP for AN-8, PE-2 installs the prefix 192.0.2.8/32 in BGP. The LDP tunnel toward PE-7 is shown, then the BGP labeled IPv4 route toward AN-8, as advertised by PE-7.

```

[/]
A:admin@PE-2# show router tunnel-table 192.0.2.7

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner    Encap TunnelId  Pref  NextHop      Metric
  Color
-----
192.0.2.7/32     ldp     MPLS  65542    9    192.168.23.2  50

```



```

-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====

```

```

[/]
A:admin@PE-2# show router bgp routes 192.0.2.8/32 label-ipv4
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  192.0.2.8/32              100        1
      192.0.2.7              None        50
      No As-Path              524280
-----
Routes : 1
=====

```

The BGP label for traffic toward AN-8 is 524280. This is the middle label in the label stack. The next hop is PE-7.

2. PE-2 programs a swap NHLFE in the egress data path to forward packets to 192.0.2.8/32, as follows:

```

[/]
A:admin@PE-2# show router ldp bindings active prefixes prefix 192.0.2.8/32
=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                Op
IngLbl                EgrLbl
EgrNextHop            EgrIf/LspId
-----
192.0.2.8/32(B)      Swap
524279                524280

```

```
192.0.2.7                               LspId 65542
-----
No. of IPv4 Prefix Active Bindings: 1
=====
```

The (B) indicates that 192.0.2.8/32 is a BGP next hop. The ingress label is the LDP transport label from AN-1 for prefix 192.0.2.8/32. The LSP ID 65542 corresponds to the LDP LSP toward egress next-hop PE-7, as shown earlier in the tunnel table. The BGP egress label for traffic toward AN-8 is 524280.

3. PE-2 programs a tunnel table entry for 192.0.2.8/32.

```
[/]
A:admin@PE-2# show router tunnel-table
=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.1/32      ldp        MPLS  65537      9     192.168.12.1  1
192.0.2.3/32      ldp        MPLS  65538      9     192.168.23.2 10
192.0.2.4/32      ldp        MPLS  65539      9     192.168.23.2 20
192.0.2.5/32      ldp        MPLS  65540      9     192.168.23.2 30
192.0.2.6/32      ldp        MPLS  65541      9     192.168.23.2 40
192.0.2.7/32      ldp        MPLS  65542      9     192.168.23.2 50
192.0.2.8/32      bgp        MPLS  262145     12     192.0.2.7    1000
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====
```

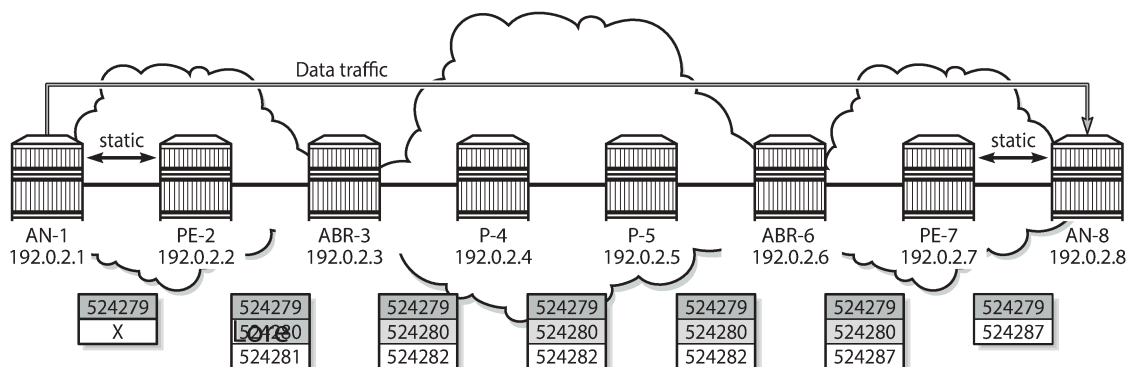
This is the only BGP tunnel in the tunnel table; all tunnels toward the other nodes are LDP tunnels. LDP routes have preference over BGP labeled routes, but there is no LDP route toward 192.0.2.8/32. Therefore, the BGP tunnel will be used for traffic destined to AN-8.

4. PE-2 advertises a new FEC label binding for prefix 192.0.2.8/32 toward AN-1. This is only done after AN-1 requests a label for prefix 192.0.2.8/32, because LDP DoD is enabled. This is possible if the ANs are DSLAMs, but not in this setup with SRs.

Data plane overview for PE-2

[Figure 222: Label stacks for traffic from AN-1 to AN-8](#) shows the label stacks that are used for traffic from AN-1 to AN-8.

Figure 222: Label stacks for traffic from AN-1 to AN-8



25618



Note:

The LDP transport label that is pushed by AN-1 is not known because of the single-hop LDP DoD implementation in 7750 SR. AN-1 cannot request the LDP label for AN-8. Therefore, the LDP transport label is represented by "X".

The service label added for the Epipe on AN-1 for egress traffic to AN-8 is 524279. Ingress traffic on AN-1 has service label 524279. This can be shown as follows:

```
[/]
A:admin@AN-1# show service id 1 labels

=====
Martini Service Labels
=====
Svc Id      Sdp Binding      Type  I.Lbl      E.Lbl
-----
1           181:1           Spok  524279     524279
-----
Number of Bound SDPs : 1
=====
```

This service label remains unchanged end-to-end.

As shown earlier, the (middle) BGP label for traffic with destination AN-8 is 524280, as follows:

```
[/]
A:admin@PE-2# show router ldp bindings active prefixes prefix 192.0.2.8/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static          (M) - Multi-homed Secondary Support
```

```

(B) - BGP Next Hop      (BU) - Alternate Next-hop for Fast Re-Route
(I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
(C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                Op
IngLbl                EgrLbl
EgrNextHop            EgrIf/LspId
-----
192.0.2.8/32(B)      Swap
524279                524280
192.0.2.7            LspId 65542
-----
No. of IPv4 Prefix Active Bindings: 1
=====

```

The next hop is PE-7, which is the PE nearest to AN-8. The BGP label will not be swapped between PE-2 and PE-7 because there is no intermediate node that has set next-hop-self. An intermediate node with next-hop-self would become the next hop instead of PE-7. The BGP label is only added or removed by the next-hop PE.

On PE-2, when a service packet with destination AN-8 is received, the ingress LDP transport label X is swapped into BGP label 524280. To reach PE-7, which is the BGP next hop for traffic toward AN-8, another LDP transport label 524281 is pushed to the packet, as follows:

```

[/]
A:admin@PE-2# show router ldp bindings active prefixes prefix 192.0.2.7/32
=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static          (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop    (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                Op
IngLbl                EgrLbl
EgrNextHop            EgrIf/LspId
-----
192.0.2.7/32          Push
--                    524281
192.168.23.2          1/1/1
-----
192.0.2.7/32          Swap
524281                524281
192.168.23.2          1/1/1
-----
No. of IPv4 Prefix Active Bindings: 2
=====

```

The next hop is ABR-3, where the ingress label 524281 is swapped to egress label 524282, as follows:

```
[/]
A:admin@ABR-3# show router ldp bindings active prefixes prefix 192.0.2.7/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.3)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.7/32                         Push
--                                   524282
192.168.34.2                         1/1/1

192.0.2.7/32                         Swap
524281                               524282
192.168.34.2                         1/1/1

-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

In the subsequent LSRs, the transport label is swapped, as follows:

On P-4:

```
[/]
A:admin@P-4# show router ldp bindings active prefixes prefix 192.0.2.7/32
---snip---
192.0.2.7/32                         Swap
524282                               524282
192.168.45.2                         1/1/1
---snip---
```

On P-5:

```
[/]
A:admin@P-5# show router ldp bindings active prefixes prefix 192.0.2.7/32
---snip---
192.0.2.7/32                         Swap
524282                               524282
192.168.56.2                         1/1/1
```

```
---snip---
```

On ABR-6, the LDP label 524282 is swapped to 524287:

```
[/]
A:admin@ABR-6# show router ldp bindings active prefixes prefix 192.0.2.7/32
---snip---
192.0.2.7/32                               Swap
524282                                     524287
192.168.67.2                               1/1/1
---snip---
```

On PE-7, the LDP label 524287 is popped, as follows:

```
[/]
A:admin@PE-7# show router ldp bindings active prefixes prefix 192.0.2.7/32
---snip---
192.0.2.7/32                               Pop
524287                                       --
--                                           --
---snip---
```

The BGP label is also popped and mapped onto LDP label 524287 that will be pushed by PE-7 on packets toward AN-8.

```
[/]
A:admin@PE-7# show router ldp bindings active prefixes prefix 192.0.2.8/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.7)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.8/32                               Push
--                                       524287
192.168.78.2                           1/1/1

192.0.2.8/32                               Swap
524281                                   524287
192.168.78.2                             1/1/1

-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

OAM

The following operations, administration, and maintenance (OAM) commands—in classic CLI—can be launched to validate an LDP FEC stitched to a BGP IPv4 labeled route and vice versa.

```
A:PE-2# oam lsp-ping bgp-label prefix 192.0.2.8/32
LSP-PING 192.0.2.8/32: 80 bytes MPLS payload
Seq=1, send from intf int-PE-2-ABR-3, reply from 192.0.2.8
    udp-data-len=32 ttl=255 rtt=7.18ms rc=4 (NoFECMapping)

---- LSP 192.0.2.8/32 PING Statistics ----
1 packets sent, 1 packets received, 0.00% packet loss
round-trip min = 7.18ms, avg = 7.18ms, max = 7.18ms, stddev = 0.000ms
```

In a similar way, LSP trace can validate LDP FEC to BGP label route stitching:

```
A:PE-2# oam lsp-trace bgp-label prefix 192.0.2.8/32
lsp-trace to 192.0.2.8/32: 0 hops min, 0 hops max, 104 byte packets
1 192.0.2.3 rtt=1.59ms rc=8(DSRtrMatchLabel)
2 192.0.2.4 rtt=2.99ms rc=8(DSRtrMatchLabel)
3 192.0.2.5 rtt=4.02ms rc=8(DSRtrMatchLabel)
4 192.0.2.6 rtt=5.40ms rc=8(DSRtrMatchLabel)
5 192.0.2.7 rtt=6.24ms rc=8(DSRtrMatchLabel) rsc=1
6 192.0.2.8 rtt=7.20ms rc=4(NoFECMapping) rsc=1
```

The detailed output includes the BGP label to LDP label mapping information at the PE:

```
A:PE-2# oam lsp-trace bgp-label prefix 192.0.2.8/32 detail
lsp-trace to 192.0.2.8/32: 0 hops min, 0 hops max, 104 byte packets
1 192.0.2.3 rtt=1.56ms rc=8(DSRtrMatchLabel)
2 192.0.2.4 rtt=2.86ms rc=8(DSRtrMatchLabel)
3 192.0.2.5 rtt=3.61ms rc=8(DSRtrMatchLabel)
4 192.0.2.6 rtt=5.23ms rc=8(DSRtrMatchLabel)
5 192.0.2.7 rtt=6.38ms rc=8(DSRtrMatchLabel) rsc=1
    DS 1: ipaddr=192.168.78.2 ifaddr=192.168.78.2 iftype=ipv4Numbered MRU=1560
    Label[1]=524287 protocol=3(LDP)
6 192.0.2.8 rtt=7.12ms rc=4(NoFECMapping) rsc=1
```

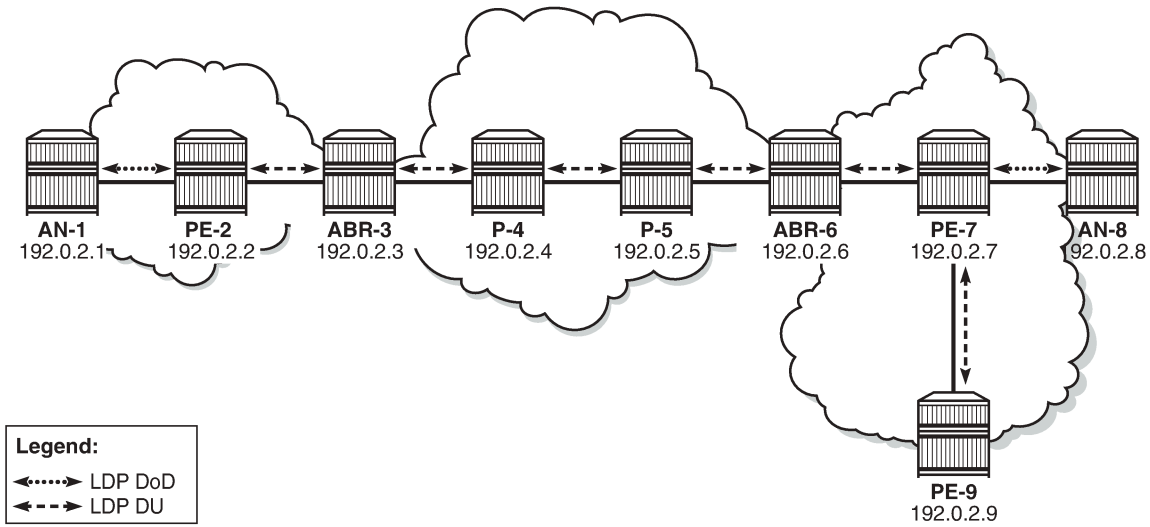
Block BGP label bindings to LDP DU peers

On a PE, labeled BGP prefixes are exported to LDP to allow LDP DoD peers to request these labels. LDP DU peers will also get all labeled BGP prefixes if not explicitly blocked by an LDP export policy, based on prefix lists. This can result in a high administrative and operational effort in large networks.

Blocking BGP label bindings to LDP DU peers is less labor-intensive because per-peer export policies are re-evaluated on NH type change (such as from BGP to LDP or to "unresolved state"), not only on a configuration change.

[Figure 223: Block BGP label bindings to LDP DU peer PE-9](#) shows the extended topology used for this configuration. The additional PE router, PE-9, does not need to know the BGP labeled prefixes. LDP DU is used between PE-7 and PE-9.

Figure 223: Block BGP label bindings to LDP DU peer PE-9



25619

Blocking BGP label bindings to LDP DU peers can be achieved in two ways:

1. LDP export policy based on prefix list.
2. LDP export policy based on BGP NH type change. No prefix list is required.

To compare the two, both are described.

LDP export policy based on prefix list

Before applying the policy to block BGP label bindings from PE-7 to PE-9, the LDP bindings on PE-9 for prefix 192.0.2.1 are the following:

```
[/]
A:admin@PE-9# show router ldp bindings prefixes prefix 192.0.2.1/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.9)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP IPv4 Prefix Bindings
=====
Prefix
Peer          FEC-Flags
IgrLbl       EgrLbl
EgrNextHop   EgrIntf/LspId
-----
192.0.2.1/32
```



```

192.0.2.7:0
--
--
--
-----
No. of IPv4 Prefix Bindings: 1
=====

```

The following policy created on PE-7 is based on a prefix list that only contains the system address of the remote AN: 192.0.2.1.

```

# on PE-7:
configure {
  policy-options {
    prefix-list "remote_AN_prefixes" {
      prefix 192.0.2.1/32 type exact {
      }
    }
  }
  policy-statement "block_BGP_bindings_remote_AN_pol" {
    entry 10 {
      from {
        prefix-list ["remote_AN_prefixes"]
      }
      action {
        action-type reject
      }
    }
  }
}

```

The policy is applied on PE-7 in the LDP session-parameters context for peer 192.0.2.9.

```

# on PE-7:
configure {
  router "Base" {
    ldp {
      session-parameters {
        peer 192.0.2.9 {
          export-prefixes ["block_BGP_bindings_remote_AN_pol"]
        }
      }
    }
  }
}

```

After the policy is applied, there are no LDP bindings for prefix 192.0.2.1 on PE-9:

```

[/]
A:admin@PE-9# show router ldp bindings prefixes prefix 192.0.2.1/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.9)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC

```

```

=====
LDP IPv4 Prefix Bindings
=====
Prefix
Peer                               FEC-Flags
IgrLbl                             EgrLbl
EgrNextHop                         EgrIntf/LspId
-----
No Matching Entries Found
=====
    
```

The original situation is restored by removing the export prefixes in the LDP session-parameters context on PE-7.

```

# on PE-7:
configure {
  router "Base"
  ldp {
    session-parameters {
      peer 192.0.2.9 {
        delete export-prefixes ["block_BGP_bindings_remote_AN_pol"]
      }
    }
  }
}
    
```

```

[/]
A:admin@PE-9# show router ldp bindings prefixes prefix 192.0.2.1/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.9)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP IPv4 Prefix Bindings
=====
Prefix
Peer                               FEC-Flags
IgrLbl                             EgrLbl
EgrNextHop                         EgrIntf/LspId
-----
192.0.2.1/32
192.0.2.7:0                          524279
--
--
-----
No. of IPv4 Prefix Bindings: 1
=====
    
```

LDP export policy based on BGP NH type change

The **from protocol name [bgp]** argument will have a different meaning in the context of per-peer and targeted export policies. For those types of policies, policies are re-evaluated on NH type change; for example, from BGP to LDP or from LDP to "unresolved state". This requires less configuration because no prefix list needs to be specified. The following policy is configured on PE-7.

```
# on PE-7:
configure {
  policy-options {
    policy-statement "block_BGP_to_LDP_DU_policy" {
      entry 10 {
        from {
          protocol {
            name [bgp]
          }
        }
        action {
          action-type reject
        }
      }
    }
  }
}
```

The policy is applied in the LDP session-parameter context for peer 192.0.2.9.

```
# on PE-7:
configure {
  router "Base" {
    ldp {
      session-parameters {
        peer 192.0.2.9 {
          export-prefixes ["block_BGP_to_LDP_DU_policy"]
        }
      }
    }
  }
}
```

PE-7 will not send BGP label mapping information for prefix 192.0.2.1/32 to PE-9, or for any other prefix of a remote AN. In this example, AN-1 with prefix 192.0.2.1/32 is the only remote AN for PE-7.

```
[/]
A:admin@PE-9# show router ldp bindings prefixes prefix 192.0.2.1/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.9)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP IPv4 Prefix Bindings
=====
Prefix
```

```
Peer          FEC-Flags
IgrLbl       EgrLbl
EgrNextHop   EgrIntf/LspId
-----
No Matching Entries Found
=====
```

Conclusion

LDP FEC to BGP label route stitching allows LDP-capable PE devices, such as DSLAMs, to offer services to LDP-capable PE devices in other areas or domains without the need to support BGP labeled routes. This feature can be used in a seamless MPLS environment.

LDP over RSVP Using OSPF as IGP

This chapter provides information about label distribution protocol (LDP) over resource reservation protocol with traffic engineering (RSVP-TE), also called LDP over RSVP, that uses RSVP label switched paths (LSPs) as a transport vehicle to carry the packets using LDP LSPs.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Additional topics](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 7.0.R5, but the configuration in this edition is based on MD-CLI in SR OS Release 21.2.R1. There are no prerequisites.

Overview

Only user packets are tunneled over RSVP LSPs; targeted LDP (T-LDP) control messages are still sent unlabeled using the interior gateway protocol (IGP) shortest path. Because LDP does not have traffic engineering (TE), it can benefit from the RSVP-TE features. LDP fast reroute (FRR) is loopfree alternate (LFA), but with LDP over RSVP, it can use RSVP FRR detour or bypass tunnels.

The main advantage of LDP over RSVP is seen in large networks. A full mesh of intra-area RSVP LSPs between PE nodes (which in some cases is not scalable) is not needed anymore. While a label edge router (LER) may not have that many tunnels, any transit node may have thousands of LSPs, and if each transit node also has to deal with detour tunnels or bypass tunnels, this number can make the label switching router (LSR) overly burdened.

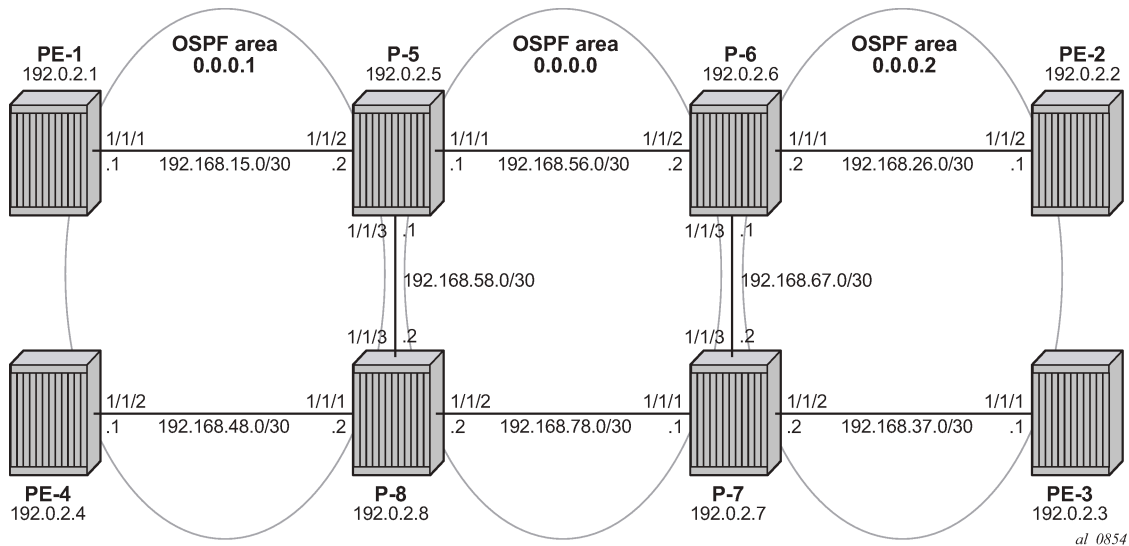
LDP over RSVP can be configured in an intra-area domain and an inter-area domain. Any router in an area can be a stitching point for LDP over RSVP. LDP over RSVP introduces a tunnel-in-tunnel tunnel type (in addition to the existing LDP tunnel type and RSVP tunnel type). If multiple tunnel types match the destination PE forwarding equivalence class (FEC) lookup, LDP prefers an LDP tunnel over an LDP over RSVP tunnel by default.

First, it is important to understand how LDP FEC resolution is working (with LDP over RSVP enabled). A more detailed description can be found later on in this chapter. The ingress LER receives an LDP label message including a FEC with prefix **P** and label **L** from a peer by a T-LDP session. LDP tries to resolve prefix **P** by performing a lookup in the Routing Table Manager (RTM). The result of this is a Next Hop (NH) to the destination PE, either an intra-area PE (intra-area context) or an Area Border Router (ABR) (inter-area context). When the NH matches the targeted LDP peer, LDP performs a second lookup for that NH in the tunnel table which returns a user configured RSVP LSP with the best metric. If there are multiple configured RSVP LSPs with the best metric, LDP selects the first available RSVP LSP. If all user configured RSVP LSPs are down, no more action is taken. If the user did not configure any RSVP LSPs under the T-LDP context, the lookup in the tunnel table returns the first available RSVP LSP which terminates on the ABR (inter-area) or intra-area PE with the lowest metric.

If the lookup in the tunnel table results in no RSVP LSP, the system can fall back to link-level interface LDP (iLDP). In that way, it is possible that the NH is reachable using iLDP. Accordingly, the egress label is then installed on the ingress LER.

Figure 224: Initial example topology shows the example topology with four PE routers and four P routers.

Figure 224: Initial example topology



OSPF area 0.0.0.1 and OSPF area 0.0.0.2 are two metro areas, connected to each other via a core area, represented by OSPF backbone area (area 0.0.0.0). Therefore, P-5, P-6, P-7, and P-8 are all acting as area border routers (ABRs). LDP over RSVP principles will be shown for intra-area PE communication (between PE-1 and PE-4) and inter-area communication (between PE-1 and PE-2).

Configuration

1. Configuring the IP/MPLS network.

The system addresses and IP interface addresses are configured according to Figure 224: Initial example topology. An interior gateway protocol (IGP) is needed to distribute routing information on all routers. In this case, the IGP is Open Shortest Path First (OSPF) using the backbone area 0.0.0.0 in the core and normal areas (area 0.0.0.1 and area 0.0.0.2) in the two metro regions, connected toward the backbone area via ABRs. A configuration example is shown for PE-1 and P-5. A similar configuration can be derived for the other P and PE nodes.

```
# on PE-1:
configure {
  router "Base" {
    interface "int-PE-1-P-5" {
      port 1/1/1
      ipv4 {
        primary {
          address 192.168.15.1
          prefix-length 30
        }
      }
    }
  }
}
```

```
interface "system" {
  ipv4 {
    primary {
      address 192.0.2.1
      prefix-length 32
    }
  }
}
ospf 0 {
  admin-state enable
  traffic-engineering
  area 0.0.0.1 {
    interface "int-PE-1-P-5" {
      interface-type point-to-point
    }
    interface "system" {
    }
  }
}
```

```
# on P-5:
configure {
  router "Base" {
    interface "int-P-5-P-6" {
      port 1/1/1
      ipv4 {
        primary {
          address 192.168.56.1
          prefix-length 30
        }
      }
    }
    interface "int-P-5-P-8" {
      port 1/1/3
      ipv4 {
        primary {
          address 192.168.58.1
          prefix-length 30
        }
      }
    }
    interface "int-P-5-PE-1" {
      port 1/1/2
      ipv4 {
        primary {
          address 192.168.15.2
          prefix-length 30
        }
      }
    }
    interface "system" {
      ipv4 {
        primary {
          address 192.0.2.5
          prefix-length 32
        }
      }
    }
  }
  ospf 0 {
    admin-state enable
    traffic-engineering
    area 0.0.0.0 {
      interface "int-P-5-P-6" {
```

```

        interface-type point-to-point
    }
    interface "int-P-5-P-8" {
        interface-type point-to-point
    }
    interface "system" {
    }
}
area 0.0.0.1 {
    interface "int-P-5-PE-1" {
        interface-type point-to-point
    }
}
}

```

Because Fast Reroute (FRR) will be enabled on the RSVP LSPs in the core area, Traffic Engineering (TE) is needed on the IGP. By doing this, OSPF will generate opaque link state advertisements (LSAs) which are collected in a Traffic Engineering Database (TED), separate from the traditional OSPF topology database. OSPF interfaces are set up as type point-to-point to improve convergence, no Designated Router/Backup Designated Router (DR/BDR) election process is performed. Convergence is beyond the scope of this chapter.

On all nodes originating and terminating a T-LDP session, an explicit **ldp-over-rsvp** parameter must be configured to enable this OSPF instance for LDP over RSVP, as follows:

```

# on PE-1, PE-2, PE-3, PE-4, P-5, P-6, P-7, P-8:
configure {
    router "Base" {
        ospf 0 {
            ldp-over-rsvp true
        }
    }
}

```

To verify that OSPF neighbors are up (state: Full), the **show router ospf neighbor** command is executed. To check if IP interface addresses/subnets are known on all PEs, **show router route-table** or **show router fib </IOM-card-slot>** displays the content of the forwarding information base (FIB).

```

[/]
A:admin@PE-1# show router ospf neighbor
=====
Rtr Base OSPFv2 Instance 0 Neighbors
=====
Interface-Name          Rtr Id      State      Pri  RetxQ  TTL
Area-Id
-----
int-PE-1-P-5           192.0.2.5   Full       1    0      33
0.0.0.1
-----
No. of Neighbors: 1
=====

```

```

[/]
A:admin@PE-1# show router route-table
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]      Type  Proto  Age      Pref
Next Hop[Interface Name] Metric
-----
192.0.2.1/32            Local Local  00h02m37s  0

```



```

system
192.0.2.2/32 Remote OSPF 00h01m15s 10
  192.168.15.2 30
192.0.2.3/32 Remote OSPF 00h00m57s 10
  192.168.15.2 40
192.0.2.4/32 Remote OSPF 00h00m35s 10
  192.168.15.2 30
192.0.2.5/32 Remote OSPF 00h01m31s 10
  192.168.15.2 10
192.0.2.6/32 Remote OSPF 00h01m17s 10
  192.168.15.2 20
192.0.2.7/32 Remote OSPF 00h00m57s 10
  192.168.15.2 30
192.0.2.8/32 Remote OSPF 00h00m37s 10
  192.168.15.2 20
192.168.15.0/30 Local Local 00h02m37s 0
  int-PE-1-P-5 0
192.168.26.0/30 Remote OSPF 00h01m15s 10
  192.168.15.2 30
192.168.37.0/30 Remote OSPF 00h00m57s 10
  192.168.15.2 40
192.168.48.0/30 Remote OSPF 00h00m35s 10
  192.168.15.2 30
192.168.56.0/30 Remote OSPF 00h01m31s 10
  192.168.15.2 20
192.168.58.0/30 Remote OSPF 00h01m31s 10
  192.168.15.2 20
192.168.67.0/30 Remote OSPF 00h01m11s 10
  192.168.15.2 30
192.168.78.0/30 Remote OSPF 00h00m37s 10
  192.168.15.2 30
-----
No. of Routes: 16
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

```

[/]
A:admin@PE-1# show router fib 1

```

```

=====
FIB Display
=====
Prefix [Flags] NextHop Protocol
-----
192.0.2.1/32 LOCAL
  192.0.2.1 (system)
192.0.2.2/32 OSPF
  192.168.15.2 (int-PE-1-P-5)
192.0.2.3/32 OSPF
  192.168.15.2 (int-PE-1-P-5)
192.0.2.4/32 OSPF
  192.168.15.2 (int-PE-1-P-5)
192.0.2.5/32 OSPF
  192.168.15.2 (int-PE-1-P-5)
192.0.2.6/32 OSPF
  192.168.15.2 (int-PE-1-P-5)
192.0.2.7/32 OSPF
  192.168.15.2 (int-PE-1-P-5)
192.0.2.8/32 OSPF

```

```

192.168.15.2 (int-PE-1-P-5)
192.168.15.0/30 LOCAL
192.168.15.0 (int-PE-1-P-5)
192.168.26.0/30 OSPF
192.168.15.2 (int-PE-1-P-5)
192.168.37.0/30 OSPF
192.168.15.2 (int-PE-1-P-5)
192.168.48.0/30 OSPF
192.168.15.2 (int-PE-1-P-5)
192.168.56.0/30 OSPF
192.168.15.2 (int-PE-1-P-5)
192.168.58.0/30 OSPF
192.168.15.2 (int-PE-1-P-5)
192.168.67.0/30 OSPF
192.168.15.2 (int-PE-1-P-5)
192.168.78.0/30 OSPF
192.168.15.2 (int-PE-1-P-5)
-----
Total Entries : 16
-----
=====

```

The next step in the process of setting up the IP/MPLS network, is enabling the IP interfaces in the **mpls** and **rsvp** context on all involved nodes (PE and P nodes). Explicit enabling of **mpls** and **rsvp** context is done by the **admin-state enable** command. The following output displays the MPLS/RSVP configuration for PE-1.

```

# on PE-1:
configure {
  router "Base" {
    mpls {
      admin-state enable
      interface "int-PE-1-P-5" {
      }
    }
    rsvp {
      admin-state enable
      interface "int-PE-1-P-5" {
      }
    }
  }
}

```

2. Configure the RSVP LSPs.

In both metro areas, RSVP LSPs are set up from all PEs toward the ABRs, no intra-area PE-PE RSVP LSPs are needed. In the core/backbone, a full RSVP LSP mesh is required. To simplify the RSVP LSP configuration, no FRR is enabled on the RSVP LSPs in the metro areas, only in the backbone area. All RSVP paths are configured as **strict paths**. As an example, the configuration for PE-1 and P-5 is as follows:

```

# on PE-1:
configure {
  router "Base" {
    mpls {
      admin-state enable
      path "path-PE-1-P-5" {
        admin-state enable
        hop 10 {
          ip-address 192.168.15.2
          type strict
        }
      }
    }
  }
}

```

```
path "path-PE-1-P-5-P-8" {
  admin-state enable
  hop 10 {
    ip-address 192.168.15.2
    type strict
  }
  hop 20 {
    ip-address 192.168.58.2
    type strict
  }
}
lsp "LSP-PE-1-P-5" {
  admin-state enable
  to 192.0.2.5
  type p2p-rsvp
  primary "path-PE-1-P-5" {
  }
}
lsp "LSP-PE-1-P-8" {
  admin-state enable
  to 192.0.2.8
  type p2p-rsvp
  primary "path-PE-1-P-5-P-8" {
  }
}
```

```
# on P-5:
configure {
  router "Base" {
    mpls {
      admin-state enable
      path "path-P-5-P-6" {
        admin-state enable
        hop 10 {
          ip-address 192.168.56.2
          type strict
        }
      }
      path "path-P-5-P-8" {
        admin-state enable
        hop 10 {
          ip-address 192.168.58.2
          type strict
        }
      }
    }
    path "path-P-5-P-6-P-7" {
      admin-state enable
      hop 10 {
        ip-address 192.168.56.2
        type strict
      }
      hop 20 {
        ip-address 192.168.67.2
        type strict
      }
    }
    path "path-P-5-PE-1" {
      admin-state enable
      hop 10 {
        ip-address 192.168.15.1
        type strict
      }
    }
  }
}
```

```
path "path-P-5-P-8-PE-4" {
  admin-state enable
  hop 10 {
    ip-address 192.168.58.2
    type strict
  }
  hop 20 {
    ip-address 192.168.48.1
    type strict
  }
}
lsp "LSP-P-5-PE-1" {
  admin-state enable
  to 192.0.2.1
  type p2p-rsvp
  primary "path-P-5-PE-1" {
  }
}
lsp "LSP-P-5-PE-4" {
  admin-state enable
  to 192.0.2.4
  type p2p-rsvp
  primary "path-P-5-P-8-PE-4" {
  }
}
lsp "LSP-P-5-P-6" {
  admin-state enable
  to 192.0.2.6
  path-computation-method local-cspf
  fast-reroute {
    frr-method facility
  }
  type p2p-rsvp
  primary "path-P-5-P-6" {
  }
}
lsp "LSP-P-5-P-7" {
  admin-state enable
  to 192.0.2.7
  path-computation-method local-cspf
  fast-reroute {
    frr-method facility
  }
  type p2p-rsvp
  primary "path-P-5-P-6-P-7" {
  }
}
lsp "LSP-P-5-P-8" {
  admin-state enable
  to 192.0.2.8
  path-computation-method local-cspf
  fast-reroute {
    frr-method facility
  }
  type p2p-rsvp
  primary "path-P-5-P-8" {
  }
}
```

The following command on PE-1 lists the RSVP LSPs:

```
[/]
A:admin@PE-1# show router mpls lsp
```

```

=====
MPLS LSPs (Originating)
=====
LSP Name          Tun   Fastfail  Adm  Opr
  To              Id     Config
-----
LSP-PE-1-P-5     1     No        Up   Up
  192.0.2.5
LSP-PE-1-P-8     2     No        Up   Up
  192.0.2.8
-----
LSPs : 2
=====

```

The following command on PE-1 shows the tunnel table for the RSVP signaling protocol. By default, RSVP LSPs have preference 7.

```

[/]
A:admin@PE-1# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner   Encap  TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.5/32     rsvp   MPLS   1         7     192.168.15.2 16777215
192.0.2.8/32     rsvp   MPLS   2         7     192.168.15.2 16777215
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====

```

On ABR P-5:

```

[/]
A:admin@P-5# show router mpls lsp

=====
MPLS LSPs (Originating)
=====
LSP Name          Tun   Fastfail  Adm  Opr
  To              Id     Config
-----
LSP-P-5-P-6     1     Yes        Up   Up
  192.0.2.6
LSP-P-5-P-7     2     Yes        Up   Up
  192.0.2.7
LSP-P-5-P-8     3     Yes        Up   Up
  192.0.2.8
LSP-P-5-PE-1    4     No        Up   Up
  192.0.2.1
LSP-P-5-PE-4    5     No        Up   Up
  192.0.2.4
-----
LSPs : 5
=====

```

```

=====
[/]
A:admin@P-5# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner    Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.1/32         rsvp    MPLS    4         7    192.168.15.1 16777215
192.0.2.4/32         rsvp    MPLS    5         7    192.168.58.2 16777215
192.0.2.6/32 [B]    rsvp    MPLS    1         7    192.168.56.2 10
192.0.2.7/32 [B]    rsvp    MPLS    2         7    192.168.56.2 20
192.0.2.8/32 [B]    rsvp    MPLS    3         7    192.168.58.2 10
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====

```

By default, the metric for strict LSPs configured without Constrained Shortest Path First (CSPF) (RSVP LSPs in metro areas) is infinite (value = 16777215). The LSP metric for CSPF LSPs (RSVP LSPs in the core area) follows the IGP cost. LSP metrics can be explicitly set on the LSP level, see also in the [Additional topics](#) section.

```

# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-P-5" {
        metric ?

        metric <number>
        <number>      - <0..16777215>
        Dynamic Default - 0

        LSP metric that forces to a constant value
      }
    }
  }
}

```

Whenever an RSVP LSP comes up, it is by default eligible for LDP over RSVP, meaning that RSVP signals to the relevant IGP (OSPF in this case) that the LSP should be included in the IGP Shortest Path First (SPF) run. The destination of the LSP (192.0.2.5) is considered as a potential endpoint in the Forwarding Equivalence Class (FEC) resolution. With the **info detail** command, all default settings of a context are shown, but the output is filtered for **ldp-over-rsvp**, as follows:

```

# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-P-5" {
        info detail | match ldp-over-rsvp

        [ex:/configure router "Base" mpls lsp "LSP-PE-1-P-5"]
        A:admin@PE-1# info detail | match ldp-over-rsvp
        ldp-over-rsvp true
      }
    }
  }
}

[/]

```

```
A:admin@PE-1# show router mpls lsp "LSP-PE-1-P-5" detail

=====
MPLS LSPs (Originating) (Detail)
=====
Legend :
+ - Inherited
=====
-----
Type : Originating
-----
LSP Name      : LSP-PE-1-P-5
LSP Type      : RegularLsp
LSP Index     : 1
From          : 192.0.2.1
To            : 192.0.2.5
Adm State     : Up
LSP Up Time   : 0d 00:05:42
Transitions   : 1
Retry Limit   : 0
Signaling     : RSVP
Hop Limit     : 255
Adaptive      : Enabled
FastReroute   : Disabled
PathCompMethod : none
FallbkPathComp : not-applicable
Metric        : N/A
Load Bal Wt   : N/A
Include Grps  :
None
Least Fill    : Disabled
BFD Template  : None
BFD Enable    : False
WaitForUpTimer : 4

LSP Tunnel ID : 1
TTM Tunnel Id : 1

Oper State    : Up
LSP Down Time : 0d 00:00:00
Path Changes  : 1
Retry Timer   : 30 sec
Resv. Style   : SE
Negotiated MTU : 1564
ClassType     : 0
Oper FR       : Disabled
ADSPEC        : Disabled

ClassForwarding : Disabled
Exclude Grps    :
None

BFD Ping Intvl : 60
BFD Failure-action : None

Revert Timer    : Disabled
Next Revert In : N/A
Entropy Label   : Enabled+
Oper Entropy Label : Enabled
Negotiated EL   : Disabled
Auto BW         : Disabled
LdpOverRsvp : Enabled
VprnAutoBind    : Enabled
IGP Shortcut    : Enabled
IGP LFA         : Disabled
BGPTransTun    : Enabled
Oper Metric     : 16777215
Prop Adm Grp    : Disabled
PCE Report     : Disabled+
PCE Control     : Disabled
Path Profile    : None
Admin Tags     : None
Lsp Self Ping   : Disabled+
SelfPingOAMFail* : 0

BGP Shortcut    : Enabled
IGP Rel Metric  : Disabled

Primary(a)      : path-PE-1-P-5
Up Time         : 0d 00:05:42

Bandwidth       : 0 Mbps
=====
* indicates that the corresponding row element may have been truncated.
```

The following command makes a specific RSVP LSP ineligible for LDP over RSVP:

```
# on PE-1:
configure {
```

```
router "Base" {
  mpls {
    lsp "LSP-PE-1-P-5" {
      ldp-over-rsvp false
    }
  }
}
```

3. Create T-LDP sessions according to RSVP LSPs.

It is a must that when configuring an RSVP LSP eligible for LDP over RSVP, also a T-LDP session is initiated. This must be done on all PE and P nodes.

```
# on PE-1:
configure {
  router "Base" {
    ldp {
      targeted-session {
        peer 192.0.2.5 {
        }
        peer 192.0.2.8 {
        }
      }
    }
  }
}
```

```
[/]
A:admin@PE-1# show router ldp session family ipv4
```

```
=====
LDP IPv4 Sessions
=====
```

Peer LDP Id	Adj Type	State	Msg Sent	Msg Recv	Up Time
192.0.2.5:0	Targeted	Established	25	27	0d 00:01:47
192.0.2.8:0	Targeted	Established	14	16	0d 00:00:58

```
-----
No. of IPv4 Sessions: 2
=====
```

4. Enable LDP over RSVP.

This is done using the **tunneling** keyword inside the T-LDP session context. This configuration is needed on all PE and ABR nodes.

```
# on PE-1:
configure {
  router "Base" {
    ldp {
      targeted-session {
        peer 192.0.2.5 {
          tunneling {
          }
        }
        peer 192.0.2.8 {
          tunneling {
          }
        }
      }
    }
  }
}
```

As a result of the **tunneling** command, the LDP over RSVP process of FEC resolving is initiated. As already stated in the introduction, FEC resolution is a three-step process. First run an SPF calculation to the destination, then select an endpoint close to that destination followed by a tunnel to that endpoint.

The next two steps go more into detail on this FEC resolution process. Step 5 will handle inter-area FEC resolving and Step 6 will handle intra-area FEC resolving.

5. Inter-area FEC resolving (ingress LER is PE-1, egress LER is PE-2)

a. Verification endpoint nodes and associated RSVP tunnels.

The first thing to do in the inter-area FEC resolving process is for PE-1 to perform an SPF calculation toward PE-2 with the purpose to search for an eligible endpoint, as close as possible to PE-2. An endpoint is eligible when:

- a T-LDP session exists between PE-1 and the endpoint node
- tunneling is configured on the endpoint node
- PE-1 received a label for the destination FEC from the endpoint node
- and an RSVP LSP that can be used for LDP over RSVP exists between PE-1 and the endpoint node

Endpoint node in OSPF area 1 can be either P-5 or P-8 (only those nodes have a T-LDP session toward PE-1). The following command shows that P-5 is the endpoint node (EgrNextHop). The RSVP LSP that is used has ID 1.

```
[/]
A:admin@PE-1# show router ldp bindings active prefixes prefix 192.0.2.2/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
      (IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static          (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop   (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.2/32                         Push
--
192.0.2.5                          LspId 1

192.0.2.2/32                         Swap
524281                               524268
192.0.2.5                          LspId 1
-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

The following command shows that RSVP LSP with ID 1 is LSP-PE-1-P-5:

```
[/]
A:admin@PE-1# show router mpls lsp

=====
MPLS LSPs (Originating)
=====
LSP Name          Tun   Fastfail  Adm  Opr
To                Id     Config
-----
LSP-PE-1-P-5      1      No        Up   Up
 192.0.2.5
LSP-PE-1-P-8      2      No        Up   Up
 192.0.2.8
-----
LSPs : 2
=====
```

The following command shows that the RSVP tunnel toward P-5 has tunnel ID 1 and next-hop 192.168.15.2:

```
[/]
A:admin@PE-1# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner   Encap TunnelId  Pref  Nexthop      Metric
Color
-----
192.0.2.2/32     ldp    MPLS  65540     9    192.0.2.5     30
192.0.2.3/32     ldp    MPLS  65541     9    192.0.2.5     40
192.0.2.4/32     ldp    MPLS  65538     9    192.0.2.5     30
192.0.2.5/32     rsvp   MPLS  1         7    192.168.15.2 16777215
192.0.2.5/32     ldp    MPLS  65537     9    192.0.2.5     10
192.0.2.6/32     ldp    MPLS  65539     9    192.0.2.5     20
192.0.2.7/32     ldp    MPLS  65542     9    192.0.2.5     30
192.0.2.8/32     rsvp   MPLS  2         7    192.168.15.2 16777215
192.0.2.8/32     ldp    MPLS  65543     9    192.0.2.8     20
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====
```

Endpoint node in OSPF area 0 can be either P-6, P-7, or P-8 (only those nodes have a T-LDP session toward P-5). The following command on P-5 shows that P-6 is the endpoint node (EgrNextHop). The RSVP LSP that is used on P-5 has ID 1.

```
[/]A:admin@P-5# show router ldp bindings active prefixes prefix 192.0.2.2/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.5)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
```

```
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static          (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop   (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding

=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.2/32                         Push
--
192.0.2.6                          LspId 1

192.0.2.2/32                         Swap
524268                               524270
192.0.2.6                          LspId 1

-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

The following command shows that the RSVP LSP with ID 1 is LSP-P-5-P-6:

```
[/]
A:admin@P-5# show router mpls lsp

=====
MPLS LSPs (Originating)
=====
LSP Name                               Tun   Fastfail  Adm  Opr
To                                     Id    Config
-----
LSP-P-5-P-6                          1    Yes     Up Up
  192.0.2.6
LSP-P-5-P-7                            2     Yes       Up   Up
  192.0.2.7
LSP-P-5-P-8                            3     Yes       Up   Up
  192.0.2.8
LSP-P-5-PE-1                          4     No        Up   Up
  192.0.2.1
LSP-P-5-PE-4                          5     No        Up   Up
  192.0.2.4

-----
LSPs : 5
=====
```

The following command shows that the RSVP tunnel with ID 1 and destination 192.0.2.6 has next-hop 192.168.56.2 and metric 10:

```
[/]
A:admin@P-5# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
Color
-----
```

```
-----
192.0.2.1/32      rsvp    MPLS  4      7      192.168.15.1  16777215
192.0.2.1/32      ldp     MPLS  65537  9      192.0.2.1     10
192.0.2.2/32      ldp     MPLS  65540  9      192.0.2.6     20
192.0.2.3/32      ldp     MPLS  65541  9      192.0.2.7     30
192.0.2.4/32      rsvp    MPLS  5       7      192.168.58.2  16777215
192.0.2.4/32      ldp     MPLS  65538  9      192.0.2.4     20
192.0.2.6/32 [B] rsvp    MPLS  1       7      192.168.56.2  10
192.0.2.6/32      ldp     MPLS  65539  9      192.0.2.6     10
192.0.2.7/32 [B]  rsvp    MPLS  2       7      192.168.56.2  20
192.0.2.7/32      ldp     MPLS  65542  9      192.0.2.7     20
192.0.2.8/32 [B]  rsvp    MPLS  3       7      192.168.58.2  10
192.0.2.8/32      ldp     MPLS  65543  9      192.0.2.8     10
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
```

On node P-6, the following commands are launched for the final destination node PE-2. Also there, an RSVP LSP toward PE-2 is used as transport tunnel for user packets.

```
[/]
A:admin@P-6# show router ldp bindings active prefixes prefix 192.0.2.2/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.6)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.2/32                         Push
--                                   524285
192.0.2.2                           LspId 4

192.0.2.2/32                         Swap
524270                               524285
192.0.2.2                           LspId 4

-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

```
[/]
A:admin@P-6# show router mpls lsp
```

```

=====
MPLS LSPs (Originating)
=====
LSP Name          Tun   Fastfail  Adm  Opr
  To              Id      Config
-----
LSP-P-6-P-5      1      Yes       Up   Up
  192.0.2.5
LSP-P-6-P-7      2      Yes       Up   Up
  192.0.2.7
LSP-P-6-P-8      3      Yes       Up   Up
  192.0.2.8
LSP-P-6-PE-2    4      No        Up   Up
  192.0.2.2
LSP-P-6-PE-3      5      No        Up   Up
  192.0.2.3
-----
LSPs : 5
=====

```

```

[/]
A:admin@P-6# show router tunnel-table
=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner   Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.1/32     ldp    MPLS  65539    9     192.0.2.5    20
192.0.2.2/32    rsvp  MPLS  4    7     192.168.26.1 16777215
192.0.2.2/32     ldp    MPLS  65537    9     192.0.2.2    10
192.0.2.3/32     rsvp   MPLS  5         7     192.168.67.2 16777215
192.0.2.3/32     ldp    MPLS  65538    9     192.0.2.3    20
192.0.2.4/32     ldp    MPLS  65542    9     192.0.2.8    30
192.0.2.5/32 [B]  rsvp   MPLS  1         7     192.168.56.1 10
192.0.2.5/32     ldp    MPLS  65540    9     192.0.2.5    10
192.0.2.7/32 [B]  rsvp   MPLS  2         7     192.168.67.2 10
192.0.2.7/32     ldp    MPLS  65541    9     192.0.2.7    10
192.0.2.8/32 [B]  rsvp   MPLS  3         7     192.168.67.2 20
192.0.2.8/32     ldp    MPLS  65543    9     192.0.2.8    20
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====

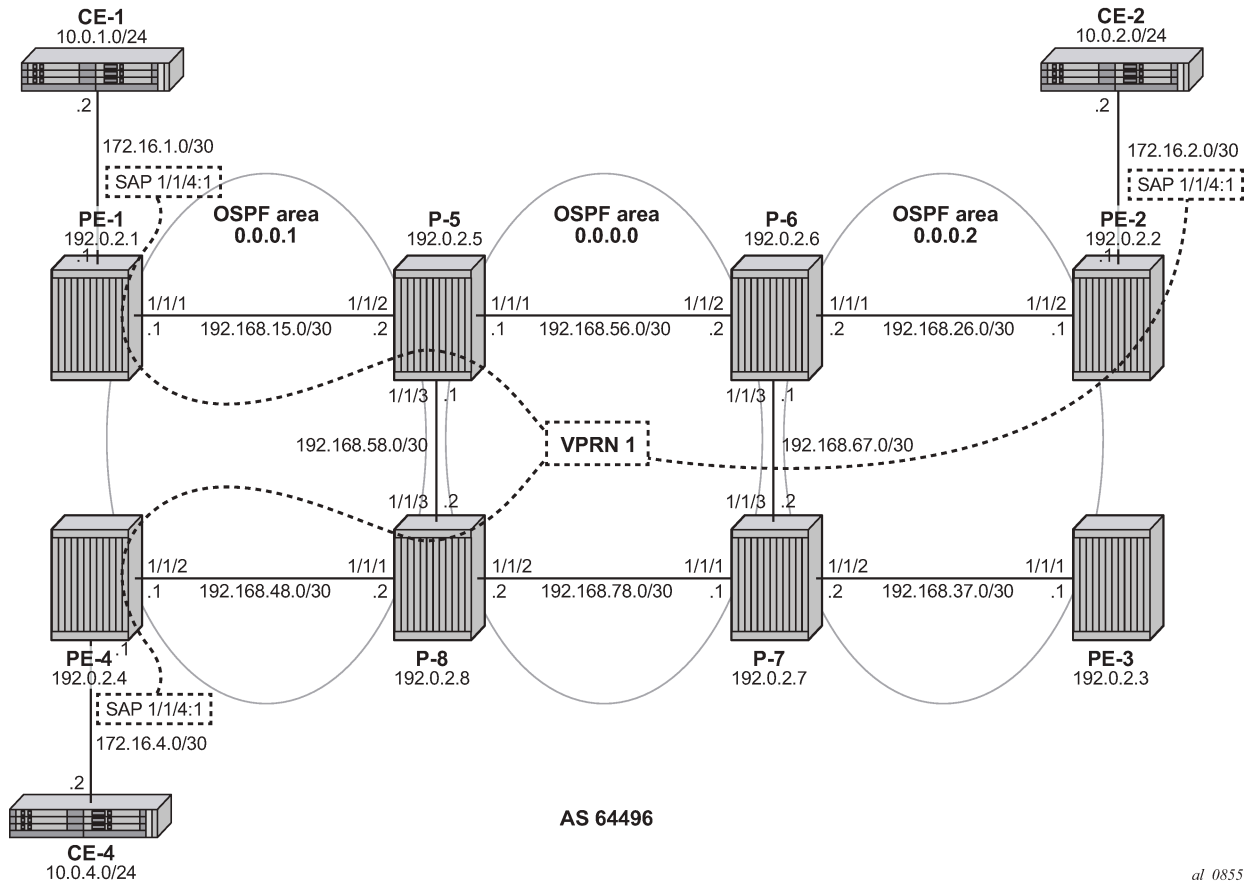
```

Nodes P-5 and P-6 behave as stitching nodes to stitch RSVP LSPs. P-5 stitches LSP-PE-1-P-5 and LSP-P-5-P-6 together while P-6 node stitches LSP-P-5-P-6 and LSP-P-6-PE-2 together.

When the endpoints are defined, one corresponding RSVP LSP to those endpoints is chosen (when ECMP equals 1). Selection criteria are as follows. When RSVP LSPs are configured under the T-LDP **tunneling** command (maximum 4), the one with the lowest LSP metric is selected. When no RSVP LSPs are configured under the T-LDP **tunneling** command, LDP checks the tunnel table for all available RSVP LSPs. The RSVP LSP with the lowest metric and operational state up is selected.

- b.** Traffic verification using a virtual private routed network (VPRN) service.

Figure 225: VPRN 1 with LDP over RSVP and no intra-area PE connectivity



al_0855

VPRN service 1 is set up between three PE nodes (PE-1, PE-2, and PE-4) using the **auto-bind-tunnel resolution-filter ldp resolution filter**. See also [Figure 225: VPRN 1 with LDP over RSVP and no intra-area PE connectivity](#) for the exact addressing scheme.

```
# on PE-1:
configure {
  service {
    vprn "VPRN 1" {
      admin-state enable
      service-id 1
      customer "1"
      autonomous-system 64496
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "64496:1"
          vrf-target {
            community "target:64496:1"
          }
        }
      }
      auto-bind-tunnel {
        resolution filter
        resolution-filter {
          ldp true
        }
      }
    }
  }
}
```

```

    }
  }
  interface "int-PE-1-CE-1" {
    ipv4 {
      primary {
        address 172.16.1.1
        prefix-length 30
      }
    }
    sap 1/1/4:1 {
    }
  }
  static-routes {
    route 10.0.1.0/24 route-type unicast {
      next-hop "172.16.1.2" {
        admin-state enable
      }
    }
  }
}

```

In order to distribute VPRN information (VPN-IPv4 routes and VPRN service labels) across the service provider network, Multi-Protocol Border Gateway Protocol (MP-BGP) is needed. MP-BGP is configured on PE-1, PE-2, and PE-4 with P-5 (192.0.2.5) being the Route Reflector (RR). In this way, no full BGP mesh between the three PE-nodes is needed, only a BGP peering toward the RR.

```

# on PE-1:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      admin-state enable
      group "internal" {
        type internal
        family {
          vpn-ipv4 true
        }
      }
      neighbor "192.0.2.5" {
        group "internal"
      }
    }
  }
}

```

```

# on P-5:
configure {
  router "Base" {
    autonomous-system 64496
    bgp {
      admin-state enable
      group "internal" {
        type internal
        family {
          vpn-ipv4 true
        }
        cluster {
          cluster-id 5.5.5.5
        }
      }
      neighbor "192.0.2.1" {
        group "internal"
      }
      neighbor "192.0.2.2" {

```

```

        group "internal"
    }
    neighbor "192.0.2.4" {
        group "internal"
    }
}

```

If user traffic is monitored between PE-1 (ingress LER) and PE-2 (egress LER), three labels are seen. The outer label is the transport label distributed using the RSVP protocol, the inner label is the service label distributed using MP-BGP. LDP over RSVP adds an extra MPLS transport label between the outer transport and the service label (distributed using LDP). This middle label is used to tell the endpoint nodes (P-5 and P-6 acting as ABR) what to do. The transport label stack contains two labels: an RSVP label and an LDP label.

The following command shows that RSVP transport label 524287 is added as the outer label on each user packet sent on the link from PE-1 to P-5:

```

[/]
A:admin@PE-1# show router rsvp session lsp-name "LSP-PE-1-P-5::path-PE-1-P-5" detail
=====
RSVP Sessions (Detailed)
=====
-----
LSP : LSP-PE-1-P-5::path-PE-1-P-5
-----
From           : 192.0.2.1           To           : 192.0.2.5
Tunnel ID      : 1               LSP ID       : 9728
Style          : SE              State        : Up
Session Type   : Originate
In Interface   : n/a            Out Interface : 1/1/1
In IF Name     : n/a
Out IF Name    : int-PE-1-P-5
In Label       : n/a            Out Label    : 524287
Previous Hop   : n/a            Next Hop     : 192.168.15.2
Hops           :
                192.168.15.2(S)
SetupPriority  : 7               Hold Priority : 0
Class Type     : 0
SubGrpOrig ID : 0               SubGrpOrig Addr:
P2MP ID       : 0
FrrAvailType  : N/A
FrrSrlgStrict : N/A           SrlgDisjoint : N/A

Path Recd     : 0               Path Sent    : 41
Resv Recd     : 46             Resv Sent    : 0
Summary msgs  :
SPath Recd   : 0               SPath Sent   : 0
SResv Recd   : 0               SResv Sent   : 0
LSP Attr Flags : N/A
=====

```

The following command shows that LDP label 524268 is added as the middle label on each user packet:

```

[/]
A:admin@PE-1# show router ldp bindings active prefixes prefix 192.0.2.2/32
=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)

```



```

=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static          (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop   (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.2/32                         Push
--
192.0.2.5                             LspId 1
524268
192.0.2.2/32                         Swap
524281                               524268
192.0.2.5                             LspId 1
-----
No. of IPv4 Prefix Active Bindings: 2
=====

```

Service label 524277 is added as the inner MP-BGP label on each user packet.



Note:

This label will not change at endpoint nodes (P-5 and P-6). Ingress LER (PE-1) will push the service label to the user packet while the egress LER (PE-2) will pop the service label.

```

[/]
A:admin@PE-1# show router bgp routes vpn-ipv4
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                    Path-Id    IGP Cost
      As-Path                               Path-Id    Label
-----
i     64496:1:10.0.1.0/24                   100        None
      192.0.2.1                             None        0
      No As-Path                             None        524277
u*>i  64496:1:10.0.2.0/24                   100        None
      192.0.2.2                             None        30
      No As-Path                             None        524277
u*>i  64496:1:10.0.4.0/24                   100        None
      192.0.2.4                             None        30

```

```

i      No As-Path                               524277
      64496:1:172.16.1.0/30                    100   None
      192.0.2.1                               None   0
      No As-Path                               524277
u*>i  64496:1:172.16.2.0/30                    100   None
      192.0.2.2                               None   30
      No As-Path                               524277
u*>i  64496:1:172.16.4.0/30                    100   None
      192.0.2.4                               None   30
      No As-Path                               524277
-----
Routes : 6
=====

```

The following command shows that RSVP transport label 524284 is added as the top label on each user packet for traffic sent from P-5 to P-6:

```

[/]
A:admin@P-5# show router rsvp session lsp-name "LSP-P-5-P-6::path-P-5-P-6" detail
=====
RSVP Sessions (Detailed)
=====
-----
LSP : LSP-P-5-P-6::path-P-5-P-6
-----
From          : 192.0.2.5                To          : 192.0.2.6
Tunnel ID    : 1                      LSP ID     : 39936
Style        : SE                     State       : Up
Session Type : Originate
In Interface : n/a                    Out Interface : 1/1/1
In IF Name   : n/a
Out IF Name  : int-P-5-P-6
In Label     : n/a                    Out Label  : 524284
Previous Hop : n/a                    Next Hop    : 192.168.56.2
Hops         :
              192.168.56.2(S)
SetupPriority : 7                      Hold Priority : 0
Class Type   : 0
SubGrpOrig ID : 0                    SubGrpOrig Addr:
P2MP ID      : 0
FrrAvailType : Facility
FrrBypassLspName: bypass-link192.168.56.2-61442
FrrSrlgStrict : N/A                  SrlgDisjoint : N/A

Path Recd    : 0                      Path Sent   : 44
Resv Recd    : 43                     Resv Sent   : 0
Summary msgs :
SPath Recd   : 0                      SPath Sent  : 0
SResv Recd   : 0                      SResv Sent  : 0
LSP Attr Flags : N/A
=====

```

LDP label 524270 is added as the middle label on each user packet.

```

[/]A:admin@P-5# show router ldp bindings active prefixes prefix 192.0.2.2/32
=====
LDP Bindings (IPv4 LSR ID 192.0.2.5)
              (IPv6 LSR ID ::)
=====
Label Status:

```

```

U - Label In Use, N - Label Not In Use, W - Label Withdrawn
WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
e - Label ELC
FEC Flags:
LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
BA - ASBR Backup FEC
(S) - Static (M) - Multi-homed Secondary Support
(B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
(I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
(C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.2/32                         Push
--
192.0.2.6                             LspId 1

192.0.2.2/32                         Swap
524268                               524270
192.0.2.6                             LspId 1
-----
No. of IPv4 Prefix Active Bindings: 2
=====

```

Service label 524277 is added as the inner MP-BGP label on each user packet.

The following command shows that RSVP transport label 524287 is added as the outer label on each user packet sent from P-6 to PE-2.

```

[/]
A:admin@P-6# show router rsvp session lsp-name "LSP-P-6-PE-2::path-P-6-PE-2" detail
=====
RSVP Sessions (Detailed)
=====
LSP : LSP-P-6-PE-2::path-P-6-PE-2
-----
From           : 192.0.2.6           To           : 192.0.2.2
Tunnel ID      : 4               LSP ID       : 1536
Style          : SE              State        : Up
Session Type   : Originate
In Interface   : n/a            Out Interface : 1/1/1
In IF Name     : n/a
Out IF Name    : int-P-6-PE-2
In Label       : n/a            Out Label   : 524287
Previous Hop   : n/a            Next Hop     : 192.168.26.1
Hops           :
              192.168.26.1(S)
SetupPriority  : 7               Hold Priority : 0
Class Type     : 0
SubGrpOrig ID : 0               SubGrpOrig Addr:
P2MP ID       : 0
FrrAvailType  : N/A
FrrSrlgStrict : N/A            SrlgDisjoint : N/A

Path Recd     : 0               Path Sent    : 43
Resv Recd     : 44            Resv Sent    : 0

```

```
Summary msgs      :
SPath Recd       : 0                SPath Sent      : 0
SResv Recd       : 0                SResv Sent      : 0
LSP Attr Flags   : N/A
=====
```

- c. LDP label 524285 is added as the middle label on each user packet.

```
[/]
A:admin@P-6# show router ldp bindings active prefixes prefix 192.0.2.2/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.6)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix          Op
IngLbl          EgrLbl
EgrNextHop      EgrIf/LspId
-----
192.0.2.2/32    Push
--             524285
192.0.2.2      LspId 4

192.0.2.2/32    Swap
524270         524285
192.0.2.2      LspId 4

-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

Service label 524277 is added as the inner MP-BGP label on each user packet.

6. Intra-area FEC resolving (ingress LER is PE-1, egress LER is PE-4).

- a. Verification endpoint node and associated RSVP tunnel.

The first thing to do in the intra-area FEC resolving process is for PE-1 to perform an SPF calculation toward PE-4 to search for an eligible endpoint, as close as possible to PE-4. An endpoint is eligible when:

- a T-LDP session exists between PE-1 and the endpoint node
- tunneling is configured on the endpoint node
- PE-1 received a label for the destination FEC from the endpoint node
- and an RSVP LSP that can be used for LDP over RSVP exists between PE-1 and the endpoint node

First endpoint node in OSPF area 1 can be either P-5 or P-8 (only those nodes have a T-LDP session toward PE-1). With **show router ldp bindings active prefixes prefix 192.0.2.4/32**, it can be concluded that P-5 is the endpoint node. Furthermore, LSP ID 1 indicates that an RSVP LSP toward P-5 is used.

```
[/]
A:admin@PE-1# show router ldp bindings active prefixes prefix 192.0.2.4/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.4/32                         Push
--
192.0.2.5                          LspId 1
-----
192.0.2.4/32                         Swap
524283                               524269
192.0.2.5                             LspId 1
-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

```
[/]
A:admin@PE-1# show router mpls lsp

=====
MPLS LSPs (Originating)
=====
LSP Name                               Tun   Fastfail  Adm  Opr
To                                     Id     Config
-----
LSP-PE-1-P-5                          1     No      Up Up
192.0.2.5
LSP-PE-1-P-8                          2     No       Up  Up
192.0.2.8
-----
LSPs : 2
=====
```

```
[/]
A:admin@PE-1# show router tunnel-table
```

```

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner    Encap TunnelId  Pref  NextHop      Metric
  Color
-----
192.0.2.2/32         ldp     MPLS  65540    9    192.0.2.5    30
192.0.2.3/32         ldp     MPLS  65541    9    192.0.2.5    40
192.0.2.4/32         ldp     MPLS  65538    9    192.0.2.5    30
192.0.2.5/32       rsvp   MPLS  1    7    192.168.15.2 16777215
192.0.2.5/32         ldp     MPLS  65537    9    192.0.2.5    10
192.0.2.6/32         ldp     MPLS  65539    9    192.0.2.5    20
192.0.2.7/32         ldp     MPLS  65542    9    192.0.2.5    30
192.0.2.8/32         rsvp    MPLS  2        7    192.168.15.2 16777215
192.0.2.8/32         ldp     MPLS  65543    9    192.0.2.8    20
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====

```

On node P-5, the same commands can be repeated for the final destination node (PE-4). Also there, an RSVP LSP toward PE-4 is used as transport tunnel for user packets.

```

[/]
A:admin@P-5# show router ldp bindings active prefixes prefix 192.0.2.4/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.5)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static          (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop    (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix          Op
Inglbl          EgrLbl
EgrNextHop      EgrIf/LspId
-----
192.0.2.4/32    Push
--             524285
192.0.2.4      LspId 5

192.0.2.4/32    Swap
524269          524285
192.0.2.4     LspId 5
-----
No. of IPv4 Prefix Active Bindings: 2

```

```

=====
[/]
A:admin@P-5# show router mpls lsp to 192.0.2.4
=====
MPLS LSPs (Originating)
=====
LSP Name                Tun   Fastfail  Adm  Opr
To                      Id    Config
-----
LSP-P-5-PE-4          5    No      Up Up
192.0.2.4
-----
LSPs : 1
=====

```

```

[/]
A:admin@P-5# show router tunnel-table
=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner   Encap TunnelId  Pref  Nexthop      Metric
Color
-----
192.0.2.1/32     rsvp   MPLS   4           7     192.168.15.1 16777215
192.0.2.1/32     ldp    MPLS   65537      9     192.0.2.1    10
192.0.2.2/32     ldp    MPLS   65540      9     192.0.2.6    20
192.0.2.3/32     ldp    MPLS   65541      9     192.0.2.7    30
192.0.2.4/32    rsvp  MPLS 5         7    192.168.58.2 16777215
192.0.2.4/32     ldp    MPLS   65538      9     192.0.2.4    20
192.0.2.6/32 [B]  rsvp   MPLS   1           7     192.168.56.2 10
192.0.2.6/32     ldp    MPLS   65539      9     192.0.2.6    10
192.0.2.7/32 [B]  rsvp   MPLS   2           7     192.168.56.2 20
192.0.2.7/32     ldp    MPLS   65542      9     192.0.2.7    20
192.0.2.8/32 [B]  rsvp   MPLS   3           7     192.168.58.2 10
192.0.2.8/32     ldp    MPLS   65543      9     192.0.2.8    10
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====

```

P-5 node acts as a stitching node to stitch RSVP LSPs. P-5 stitches LSP-PE-1-P-5 and LSP-P-5-PE-4 together.

When the endpoint node (P-5) is defined, the corresponding RSVP LSP to this endpoint is chosen. Selection criteria are as follows (when ECMP=1). When RSVP LSPs are configured under the T-LDP **tunneling** command (maximum 4), the one with the lowest LSP metric is selected. When no RSVP LSPs are configured under the T-LDP **tunneling** command, LDP checks the tunnel table for all available RSVP LSPs. The RSVP LSP with the lowest metric and operational state **up** is selected.

- b. Traffic verification using a VPRN service (see [Figure 225: VPRN 1 with LDP over RSVP and no intra-area PE connectivity](#)).

If user traffic between PE-1 (ingress LER) and PE-4 (egress LER) is monitored, three labels are seen. The outer label is the transport label (distributed using RSVP protocol), the inner label is the service label (distributed using MP-BGP). LDP over RSVP adds an extra MPLS transport label

between outer and inner label (distributed using LDP). This middle label is used to tell the endpoint node (P-5) what to do.

The following command shows that transport label 524287 is added as the top RSVP label on each user packet sent from PE-1 to P-5.

```
[/]
A:admin@PE-1# show router rsvp session lsp-name "LSP-PE-1-P-5::path-PE-1-P-5" detail

=====
RSVP Sessions (Detailed)
=====
-----
LSP : LSP-PE-1-P-5::path-PE-1-P-5
-----
-----
From           : 192.0.2.1           To           : 192.0.2.5
Tunnel ID      : 1               LSP ID       : 9728
Style          : SE              State        : Up
Session Type   : Originate
In Interface   : n/a            Out Interface : 1/1/1
In IF Name     : n/a
Out IF Name    : int-PE-1-P-5
In Label       : n/a            Out Label   : 524287
Previous Hop   : n/a            Next Hop     : 192.168.15.2
Hops           :
                192.168.15.2(S)
SetupPriority  : 7               Hold Priority : 0
Class Type    : 0
SubGrpOrig ID : 0               SubGrpOrig Addr:
P2MP ID       : 0
FrrAvailType  : N/A
FrrSrlgStrict : N/A           SrlgDisjoint : N/A

Path Recd     : 0               Path Sent    : 45
Resv Recd     : 51            Resv Sent    : 0
Summary msgs  :
SPath Recd   : 0               SPath Sent   : 0
SResv Recd   : 0               SResv Sent   : 0
LSP Attr Flags : N/A
=====
```

LDP over RSVP label 524269 is added as the middle LDP label on each user packet.

```
[/]
A:admin@PE-1# show router ldp bindings active prefixes prefix 192.0.2.4/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
                (IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
```



```

=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.4/32                         Push
--
192.0.2.5                             LspId 1

192.0.2.4/32                         Swap
524283                               524269
192.0.2.5                             LspId 1

-----
No. of IPv4 Prefix Active Bindings: 2
=====

```

Service label 524277 is added as the inner MP-BGP label on each user packet.



Note:

This label will not change at endpoint node (P-5). Ingress LER (PE-1) will push the service label to the user packet while the egress LER (PE-4) will pop the service label.

```

[/]
A:admin@PE-1# show router bgp neighbor 192.0.2.5 received-routes vpn-ipv4

=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
BGP VPN-IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
i     64496:1:10.0.1.0/24                    100        None
      192.0.2.1                             None        0
      No As-Path                             524277
u*>i 64496:1:10.0.2.0/24                    100        None
      192.0.2.2                             None        30
      No As-Path                             524277
u*>i 64496:1:10.0.4.0/24                    100        None
      192.0.2.4                             None        30
      No As-Path                             524277
i     64496:1:172.16.1.0/30                 100        None
      192.0.2.1                             None        0
      No As-Path                             524277
u*>i 64496:1:172.16.2.0/30                 100        None
      192.0.2.2                             None        30
      No As-Path                             524277
u*>i 64496:1:172.16.4.0/30                 100        None
      192.0.2.4                             None        30
      No As-Path                             524277
-----
Routes : 6

```

The following command shows that P-5 pushes RSVP transport label 524283 as the top label on each user packet sent on path-P-5-P-8-PE-4. This RSVP transport label is swapped by P-8 to label 524287.

```
[/]
A:admin@P-5# show router mpls lsp "LSP-P-5-PE-4" path detail

=====
MPLS LSP LSP-P-5-PE-4 Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use
  b - Bandwidth Protected      n - Node Protected
  s - Soft Preemption          L - Loose
  S - Strict                    + - Inherited
  A - ABR

-----
LSP LSP-P-5-PE-4
Path path-P-5-P-8-PE-4
-----
LSP Name      : LSP-P-5-PE-4
From          : 192.0.2.5
To            : 192.0.2.4
Admin State   : Up
Path Name     : path-P-5-P-8-PE-4
Path LSP ID   : 36352
Path Admin    : Up
Out Interface : 1/1/3
Path Up Time  : 0d 00:23:13
Retry Limit   : 0
Retry Attempt : 0
Oper State    : Up
Path Type     : Primary
Path Oper     : Up
Out Label   : 524283
Path Down Time : 0d 00:00:00
Retry Timer   : 30 sec
Next Retry In : 0 sec

BFD Configuration and State
Template      : None
Enable        : False
WaitForUpTimer : 4 sec
WaitForUpTmLeft : 0 sec
Ping Interval : 60
State         : notApplicable
OperWaitForUpTimer: N/A

Adspec       : Disabled
PathCompMethod : none
MetricType   : igp
Least Fill   : Disabled
FRR          : Disabled
Propagate Adm Grp: Disabled
Inter-area   : False
Oper Adspec   : Disabled
OperPathCompMethod: none
Oper MetricType : igp
Oper LeastFill : Disabled
Oper FRR       : Disabled
Oper Prop Adm Grp : Disabled

PCE Report   : Disabled+
PCE Control  : Disabled
PCE Update ID : 0
Oper PCE Report : Disabled
Oper PCE Control : Disabled

Neg MTU      : 1564
Bandwidth    : No Reservation
Hop Limit    : 255
Record Route : Record
Record Label : Record
Setup Priority : 7
Hold Priority : 0
Class Type   : 0
Backup CT    : None
MainCT Retry : n/a
Oper MTU     : 1564
Oper Bandwidth : 0 Mbps
Oper HopLimit : 255
Oper Record Route : Record
Oper Record Label : Record
Oper SetupPriority: 7
Oper HoldPriority : 0
Oper CT       : 0
```

```

Rem :
MainCT Retry : 0
Limit :
Include Groups : Oper IncludeGroups:
None None
Exclude Groups : Oper ExcludeGroups:
None None

Adaptive : Enabled Oper Metric : 16777215
Preference : n/a
Path Trans : 1 CSPF Queries : 0
Failure Code : noError
Failure Node : n/a
Explicit Hops :
192.168.58.2(S)
-> 192.168.48.1(S)
Actual Hops :
192.168.58.1(192.0.2.5) Record Label : N/A
-> 192.168.58.2(192.0.2.8) Record Label : 524283
-> 192.168.48.1 Record Label : 524287
Resignal Eligible: False
Last Resignal : n/a CSPF Metric : 0
=====

```



Note:

show router rsvp session lsp-name LSP-P-5-PE-4::path-P-5-P-8-PE-4 detail cannot be used because it only shows the outgoing RSVP label toward node P-8. On node P-8, RSVP transport label 524283 will be swapped into RSVP transport label 524287 for the link from P-8 to PE-4.

LDP label 524285 is added as the middle label on each user packet.

```

[/]
A:admin@P-5# show router ldp bindings active prefixes prefix 192.0.2.4/32
=====
LDP Bindings (IPv4 LSR ID 192.0.2.5)
(IPv6 LSR ID ::)
=====
Label Status:
U - Label In Use, N - Label Not In Use, W - Label Withdrawn
WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
e - Label ELC
FEC Flags:
LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
BA - ASBR Backup FEC
(S) - Static (M) - Multi-homed Secondary Support
(B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
(I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
(C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix Op
IngLbl EgrLbl
EgrNextHop EgrIf/LspId
-----
192.0.2.4/32 Push
-- 524285
192.0.2.4 LspId 5
192.0.2.4/32 Swap

```

```

524269                               524285
192.0.2.4                             LspId 5

-----
No. of IPv4 Prefix Active Bindings: 2
=====

```

Service label 524277 is added as the inner MP-BGP label on each user packet.

```

[/]A:admin@P-5# show router bgp neighbor 192.0.2.4 received-routes vpn-ipv4
=====
BGP Router ID:192.0.2.5          AS:64496          Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                  l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
BGP VPN-IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                       Path-Id    IGP Cost
      As-Path                                Label
-----
*>i  64496:1:10.0.4.0/24                       100        None
      192.0.2.4                               None        20
      No As-Path                               524277
*>i  64496:1:172.16.4.0/30                      100        None
      192.0.2.4                               None        20
      No As-Path                               524277
-----
Routes : 2
=====

```

Additional topics

prefer-tunnel-in-tunnel

If the next-hop router advertised the same FEC over link-level LDP (iLDP), LDP prefers the iLDP tunnel by default unless the user explicitly changed the default preference using the **prefer-tunnel-in-tunnel** command. When **prefer-tunnel-in-tunnel** is set to false, an LDP over RSVP tunnel has precedence.

Until now, no RSVP LSPs are configured inside the **ldp targeted-session peer tunneling** context. Therefore, two additional strict non-CSPF RSVP LSPs are added between ingress LER PE-1 and egress LER P-5. Both LSPs have an explicit metric setting and are applied inside the **ldp tunneling** context. On the Layer 3 interface between PE-1 and P-5, iLDP is enabled.

```

# on PE-1:
configure {
  router "Base" {
    ldp {
      interface-parameters {
        interface "int-PE-1-P-5" {
          ipv4 {
            admin-state enable
          }
        }
      }
    }
  }
}

```

```

    }
}

# on P-5:
configure {
  router "Base" {
    ldp {
      interface-parameters {
        interface "int-P-5-PE-1" {
          ipv4 {
            admin-state enable
          }
        }
      }
    }
  }
}

```

```

# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-P-5-metric100" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.5
        metric 100
        primary "path-PE-1-P-5" {
        }
      }
      lsp "LSP-PE-1-P-5-metric200" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.5
        metric 200
        primary "path-PE-1-P-5" {
        }
      }
    }
  }
}

```

```

# on PE-1:
configure
  router "Base" {
    ldp {
      targeted-session {
        peer 192.0.2.5 {
          tunneling {
            lsp "LSP-PE-1-P-5-metric100" { }
            lsp "LSP-PE-1-P-5-metric200" { }
          }
        }
      }
    }
  }
}

```

The following tunnel table on node PE-1 contains four tunnels toward P-5: one LDP tunnel and three RSVP tunnels:

```

[/]
A:admin@PE-1# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner      Encap TunnelId  Pref  Nexthop      Metric
Color

```

```

-----
192.0.2.2/32      ldp      MPLS  65540   9      192.168.15.2  30
192.0.2.3/32      ldp      MPLS  65541   9      192.168.15.2  40
192.0.2.4/32      ldp      MPLS  65538   9      192.168.15.2  30
192.0.2.5/32      rsvp     MPLS   3       7      192.168.15.2  100
192.0.2.5/32      rsvp     MPLS   4       7      192.168.15.2  200
192.0.2.5/32      rsvp     MPLS   1       7      192.168.15.2  16777215
192.0.2.5/32      ldp      MPLS  65537   9      192.168.15.2  10
192.0.2.6/32      ldp      MPLS  65539   9      192.168.15.2  20
192.0.2.7/32      ldp      MPLS  65542   9      192.168.15.2  30
192.0.2.8/32      rsvp     MPLS   2       7      192.168.15.2  16777215
192.0.2.8/32      ldp      MPLS  65543   9      192.168.15.2  20
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====

```

Tunnel ID 1 is a reference to LSP-PE-1-P-5. Tunnel ID 3 is a reference to LSP-PE-1-P-5-metric100. Tunnel ID 4 is a reference to LSP-PE-1-P-5-metric200 and owner LDP is a reference to iLDP.

Taken into account the FEC resolution rules, iLDP prevails, so no LDP over RSVP tunnel is used. The following command shows that the egress interface is 1/1/1; no RSVP LSP is used, so no LSP ID is present:

```

[/]
A:admin@PE-1# show router ldp bindings active prefixes prefix 192.0.2.5/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static          (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop    (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix          Op
IngLbl          EgrLbl
EgrNextHop      EgrIf/LspId
-----
192.0.2.5/32    Push
--             524271
192.168.15.2    1/1/1

192.0.2.5/32    Swap
524284          524271
192.168.15.2    1/1/1

-----
No. of IPv4 Prefix Active Bindings: 2
=====

```

This behavior can be changed by setting the **prefer-tunnel-in-tunnel** command in the **ldp** context to false. Now, the LDP over RSVP tunnel with the best (= lowest) metric is taken.

```
# on PE-1:
configure {
  router "Base" {
    ldp {
      prefer-tunnel-in-tunnel false
    }
  }
}
```

The following command shows that LSP ID 3 is used:

```
[/]
A:admin@PE-1# show router ldp bindings active prefixes prefix 192.0.2.5/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.5/32                          Push
--                                     524271
192.0.2.5                              LspId 3

192.0.2.5/32                          Swap
524284                                 524271
192.0.2.5                              LspId 3

-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

The following command shows that LSP ID 3 corresponds to LSP-PE-1-P-5-metric100:

```
[/]
A:admin@PE-1# show router mpls lsp

=====
MPLS LSPs (Originating)
=====
LSP Name                               Tun   Fastfail  Adm  Opr
To                                     Id     Config
-----
LSP-PE-1-P-5                           1     No        Up   Up
192.0.2.5
```

```

LSP-PE-1-P-8                2      No      Up      Up
 192.0.2.8
LSP-PE-1-P-5-metric100    3      No      Up      Up
 192.0.2.5
LSP-PE-1-P-5-metric200     4      No      Up      Up
 192.0.2.5
-----
LSPs : 4
=====

```

When the LSP-PE-1-P-5-metric100 is disabled, then the LSP-PE-1-P-5-metric200 becomes active.

```

# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-P-5-metric100" {
        admin-state disable
      }
    }
  }
}

```

The following command shows that LSP ID 4 is used for traffic toward 192.0.2.5:

```

[/]
A:admin@PE-1# show router ldp bindings active prefixes prefix 192.0.2.5/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static          (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop    (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.5/32                          Push
--                                     524271
192.0.2.5                              LspId 4

192.0.2.5/32                          Swap
524284                                 524271
192.0.2.5                              LspId 4

-----
No. of IPv4 Prefix Active Bindings: 2
=====

```

The following command shows that LSP ID 4 corresponds to LSP-PE-1-P-5-metric200:

```

[/]

```



```
A:admin@PE-1# show router mpls lsp

=====
MPLS LSPs (Originating)
=====
LSP Name                               Tun   Fastfail Adm  Opr
To                                     Id    Config
-----
LSP-PE-1-P-5                           1     No       Up  Up
  192.0.2.5
LSP-PE-1-P-8                           2     No       Up  Up
  192.0.2.8
LSP-PE-1-P-5-metric100                 3     No       Dwn Dwn
  192.0.2.5
LSP-PE-1-P-5-metric200                4     No       Up  Up
  192.0.2.5
-----
LSPs : 4
=====
```

When LSP-PE-1-P-5-metric200 is disabled too, iLDP resumes.

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-P-5-metric200" {
        admin-state disable
      }
    }
  }
}
```

The following command shows that iLDP is used and the egress interface is port 1/1/1:

```
[/]
A:admin@PE-1# show router ldp bindings active prefixes prefix 192.0.2.5/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.5/32                          Push
  --                                  524271
192.168.15.2                          1/1/1

192.0.2.5/32                          Swap
524284                                  524271
```

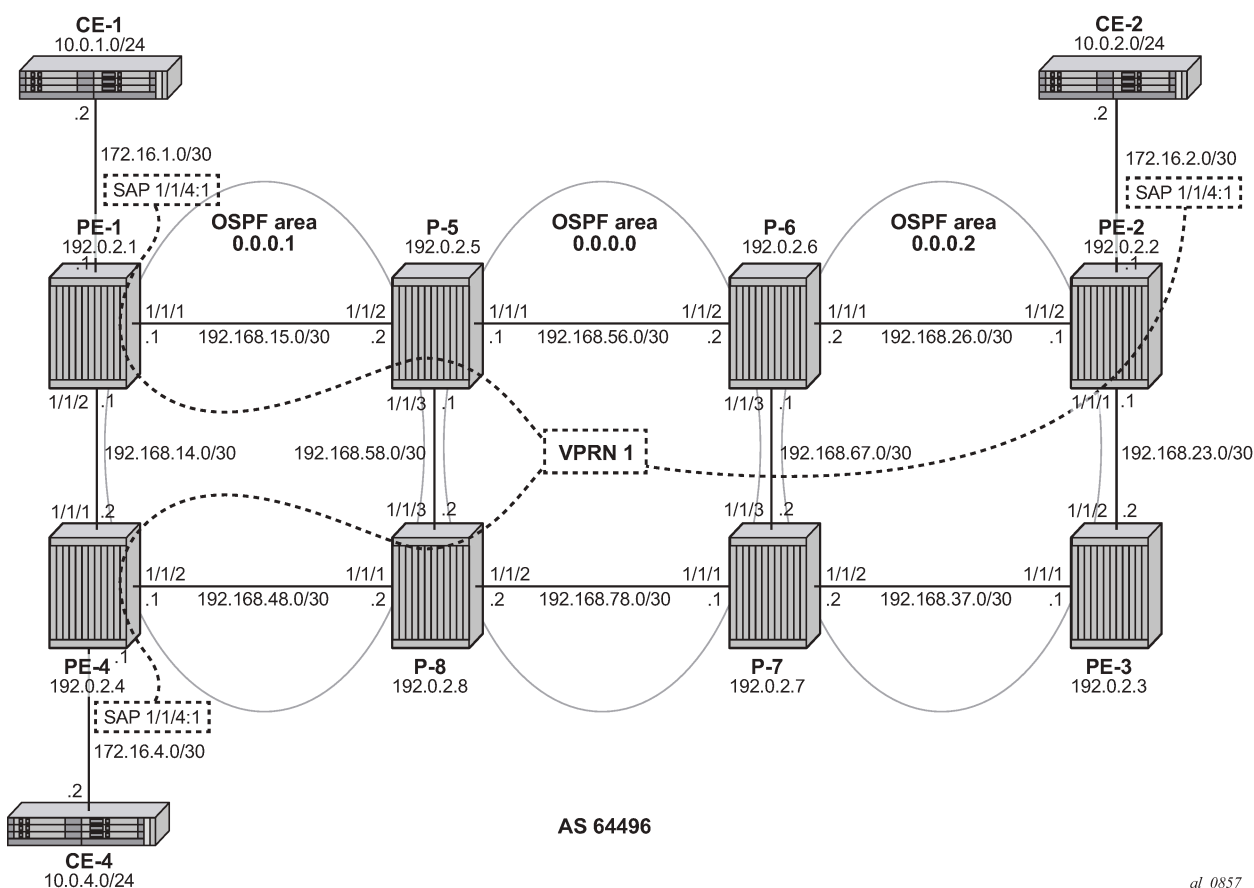
```

192.168.15.2                               1/1/1
-----
No. of IPv4 Prefix Active Bindings: 2
=====
    
```

Intra-PE connectivity changes LDP over RSVP behavior

Figure 226: VPRN 1 with LDP over RSVP and intra-area PE connectivity shows two metro areas; both of the intra PEs are physically connected with each other. Compared with the previous figures, PE-1 is directly connected to PE-4 and PE-2 is directly connected to PE-3 (up to the OSPF level).

Figure 226: VPRN 1 with LDP over RSVP and intra-area PE connectivity



The SPF path calculation on PE-1 toward destination (PE-4) does not point to node P-5 anymore (as was seen before), but now points directly to PE-4 (shortest, lowest IGP metric). As a conclusion, it can be said that when possible intra-area endpoint nodes are not part of the calculated SPF path, LDP over RSVP is not preferred anymore. For this situation, it is advisable to configure iLDP on the intra-PE interfaces to have a fallback mechanism.

This is configured on PE-1 and PE-4 as follows:

```

# on PE-1:
configure {
    
```

```
router "Base" {
  interface "int-PE-1-PE-4" {
    port 1/1/2
    ipv4 {
      primary {
        address 192.168.14.1
        prefix-length 30
      }
    }
  }
  ospf 0 {
    area 0.0.0.1 {
      interface "int-PE-1-PE-4" {
        interface-type point-to-point
      }
    }
  }
}
```

```
# on PE-4:
configure {
  router "Base" {
    interface "int-PE-4-PE-1" {
      port 1/1/1
      ipv4 {
        primary {
          address 192.168.14.2
          prefix-length 30
        }
      }
    }
  }
  ospf 0 {
    area 0.0.0.1 {
      interface "int-PE-4-PE-1" {
        interface-type point-to-point
      }
    }
  }
}
```

LDP is configured on the interfaces between PE-1 and PE-4, as follows:

```
# on PE-1:
configure {
  router "Base" {
    ldp {
      interface-parameters {
        interface "int-PE-1-PE-4" {
          ipv4 {
            admin-state enable
          }
        }
      }
    }
  }
}
```

```
# on PE-4:
configure {
  router "Base" {
    ldp {
      interface-parameters {
        interface "int-PE-4-PE-1" {
          ipv4 {
            admin-state enable
          }
        }
      }
    }
  }
}
```

```
}

```

From the moment iLDP is configured, an LDP LSP is set up. Intra-area PE traffic will flow over this LDP LSP.

```
[/]
A:admin@PE-1# show router tunnel-table 192.0.2.4/32

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.4/32         ldp        MPLS  65544    9    192.168.14.2  10
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====
```

If user traffic is monitored between ingress LER PE-1 and egress LER PE-4, only two labels are seen. The outer label is the transport label distributed using LDP; the inner label is the service label distributed using MP-BGP. No LDP over RSVP label is present anymore. The following command shows that LDP transport label 524285 is pushed by PE-1 as the outer label on packets destined to PE-4:

```
[/]A:admin@PE-1# show router ldp bindings active prefixes prefix 192.0.2.4/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix          Op
IngLbl          EgrLbl
EgrNextHop     EgrIf/LspId
-----
192.0.2.4/32    Push
--             524285
192.168.14.2    1/1/2

192.0.2.4/32    Swap
524283          524285
192.168.14.2    1/1/2
-----
No. of IPv4 Prefix Active Bindings: 2
```

Service label 524277 is added as the inner MP-BGP label on each user packet.

```
[/]
A:admin@PE-1# show router bgp routes vpn-ipv4
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                       Path-Id     IGP Cost
      As-Path                                Label
-----
i     64496:1:10.0.1.0/24                     100        None
      192.0.2.1                               None        0
      No As-Path                               524277
u*>i 64496:1:10.0.2.0/24                     100        None
      192.0.2.2                               None        30
      No As-Path                               524277
u*>i 64496:1:10.0.4.0/24                     100        None
      192.0.2.4                               None        10
      No As-Path                               524277
i     64496:1:172.16.1.0/30                  100        None
      192.0.2.1                               None        0
      No As-Path                               524277
u*>i 64496:1:172.16.2.0/30                  100        None
      192.0.2.2                               None        30
      No As-Path                               524277
u*>i 64496:1:172.16.4.0/30                  100        None
      192.0.2.4                               None        10
      No As-Path                               524277
-----
Routes : 6
=====
```

Conclusion

LDP over RSVP allows tunneling of user packets toward an LDP far-end destination inside an RSVP LSP (with the benefits of RSVP LSPs, fast-reroute (FRR) and traffic engineering (TE)). The main application of this feature is for deployment of MPLS based services, for example, VPRN, virtual leased line (VLL), and virtual private LAN service (VPLS) services, in large networks where a full mesh of LSPs reaches the limits of scalability.

LDP Point-to-Point LSPs

This chapter provides information about label distribution protocol (LDP) point-to-point label switched paths (LSPs)

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter is applicable to SR OS and was originally written for SR OS Release 7.0.R5. The MD-CLI output in the current edition corresponds to SR OS Release 21.2.R1. There are no prerequisites or conditions on the hardware for this configuration.

Overview

Due to the connectionless nature of the network layer protocol IP, packets travel through the network on a hop-by-hop basis with routing decisions made at each node. As a result, hyperaggregation of data on certain links may occur and it may impact the provider's ability to provide guaranteed service levels across the network end-to-end. To address these shortcomings, Multi-Protocol Label Switching (MPLS) was developed.

MPLS provides the capability to establish connection-oriented paths, called Label Switched Paths (LSPs), over a connectionless (IP) network. The LSP offers a mechanism to engineer network traffic independently from the underlying network routing protocol (mostly IP) to improve the network resiliency and recovery options and to permit delivery of services that are not readily supported by conventional IP routing techniques, such as Layer 2 IP Virtual Private Networks (VPNs). These benefits are essential for today's communication network explaining the wide deployment base of the MPLS technology.

RFC 3031, *Multiprotocol Label Switching Architecture*, specifies the MPLS architecture whereas this chapter describes the configuration and troubleshooting of point-to-point LSPs on SR OS.

Packet forwarding

When a packet of a connectionless network layer protocol travels from one router to the next, each router in the network makes an independent forwarding decision by performing the following basic tasks: first analyzing the packet's header, then referencing the local routing table to find the longest match based on the destination address in the IP header, and finally sending out the packet on the selected interface.

In other terms, the first function partitions the entire set of possible packets into a set of Forwarding Equivalence Classes (FECs). All packets associated to a particular FEC will be forwarded along the same logical path to the same destination. The second function maps each FEC to a next hop destination router. Each router along the packet's path performs these actions.

In MPLS, the assignment of a packet to a particular FEC is done just once, when the packet enters the network. In turn, the FEC is mapped to an LSP, which is established prior to packet forwarding.

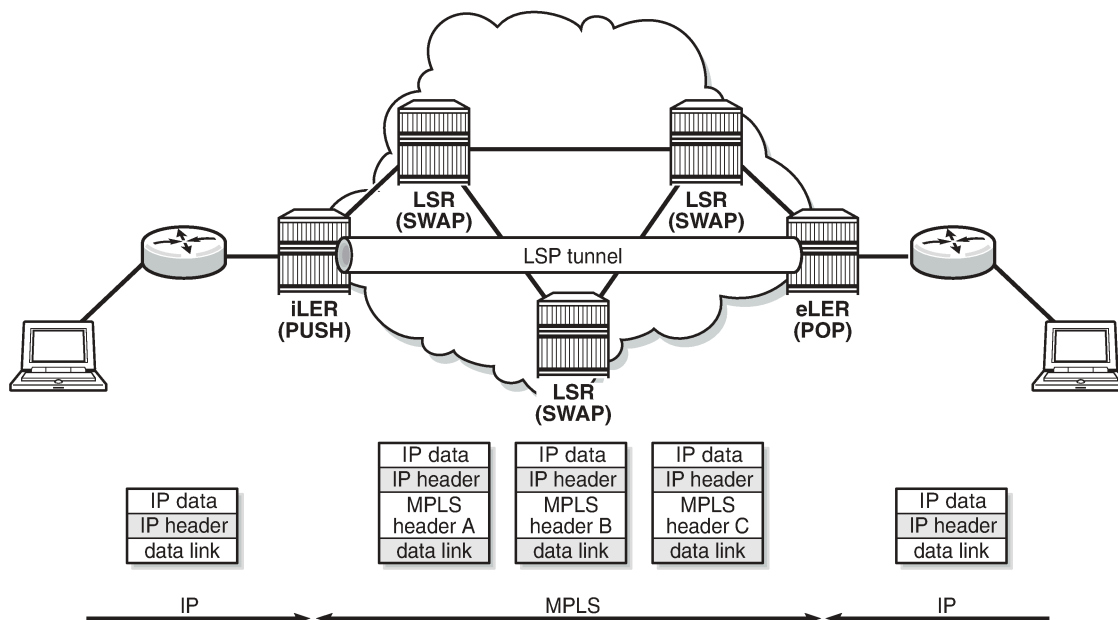
An MPLS label, representing the FEC to which the packet is assigned, is attached to the packet (push operation) and once labeled, the packet is forwarded to the next hop router along that LSP path.

At subsequent hops, no analysis of the packet's network layer header is needed. Instead, the label is used as an index into a table which specifies the next hop and a new label. The old label is replaced with the new label (swap operation), and the packet is forwarded to its next hop.

At the MPLS network egress, the label is removed from the packet (pop operation). If this router is the destination (based on the remaining packet), the packet is handed to the receiving application, such as a Virtual Private LAN Service (VPLS). If this router is not the destination of the packet, the packet will be sent into a new MPLS tunnel or forwarded by conventional IP forwarding toward the Layer 3 destination

Terminology

Figure 227: Generic MPLS network, MPLS label operations



25762

Figure 227: Generic MPLS network, MPLS label operations shows a general network topology clarifying the MPLS-related terms. A Label Edge Router (LER) is a device at the edge of an MPLS network, with at least one interface outside the MPLS domain. A router is usually defined as an LER based on its position relative to a particular LSP. The MPLS router at the head-end of an LSP is called the ingress Label Edge Router (iLER). The MPLS router at the tail-end of an LSP is called the egress Label Edge Router (eLER).

The iLER receives unlabeled packets from outside the MPLS domain, then applies MPLS labels to the packets, and forwards the labeled packets into the MPLS domain.

The eLER receives labeled packets from the MPLS domain, then removes the labels, and forwards unlabeled packets outside the MPLS domain. The eLER can signal an implicit-null label (numeric value

3). This informs the previous hop to send MPLS packets without an outer label and so is known as Penultimate Hop Popping (PHP).

A Label Switching Router (LSR) is a device internal to an MPLS network, with all interfaces inside the MPLS domain. These devices switch labeled packets inside the MPLS domain. In the core of the network, LSRs ignore the packet's network layer (IP) header and simply forward the packet using the MPLS label swapping mechanism.

LSP establishment

Prior to packet forwarding, the LSP must be established. In order to do so, labels need to be distributed for the path. Labels are usually distributed by a downstream router in the upstream direction (relative to the data flow). There are a number of ways used for label distribution: static, LDP, and RSVP. For static P2P LSPs, see chapter [Static Point-to-Point LSPs](#); for RSVP-TE P2P LSPs, see chapter [RSVP Point-to-Point LSPs](#).

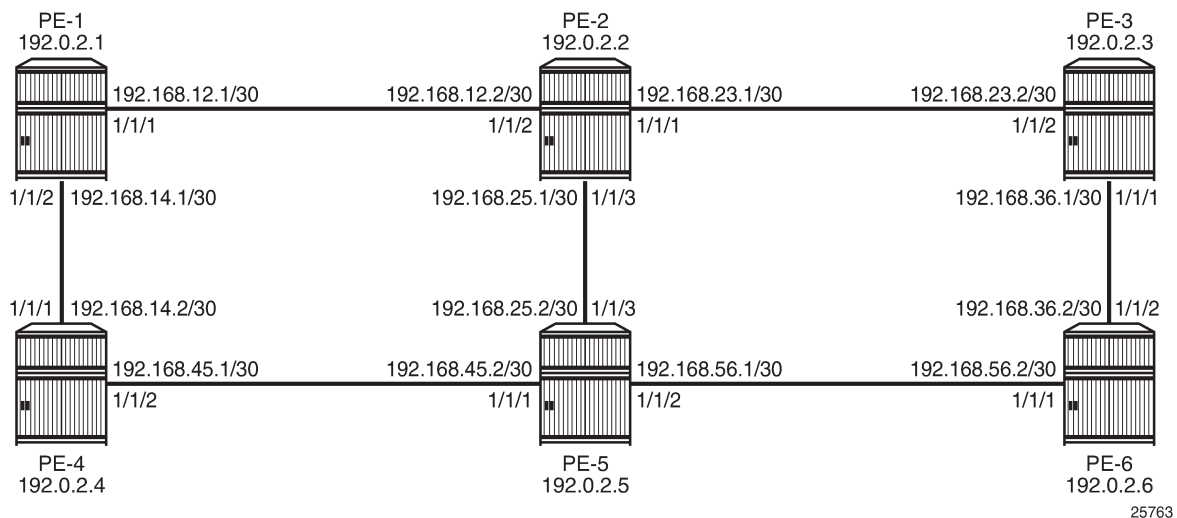
LDP (RFC 5036, *LDP Specification*) can be considered as an extension to the network interior gateway protocol (IGP). As routers become aware of new destination networks, they advertise labels in the upstream direction that will allow upstream routers to reach the destination.

Fast reroute (FRR) allows for computing backup paths and advertising the backup labels before a failure takes place. This way traffic can flow almost continuously, without waiting for routing protocol convergence; see chapter [MPLS LDP FRR using ISIS as IGP](#).

Example topology

[Figure 228: MPLS example topology](#) shows the example topology consisting of six SR OS nodes located in a single autonomous system.

Figure 228: MPLS example topology



Configuration

As a general prerequisite for the configuration of MPLS LSPs, a correctly working Interior Gateway Protocol (IGP) is required. Open Shortest Path First (OSPF) or Intermediate System to Intermediate System (IS-IS) can be used as IGP.

LDP is a simple label distribution protocol with basic MPLS functionality (no traffic engineering). Fast Reroute is supported; see chapter [MPLS LDP FRR using ISIS as IGP](#). LDP relies on the underlying routing information provided by an IGP in order to forward labeled packets. Each LDP configured LSR will originate a label for its system address and a label for each FEC for which it has a next hop that is external to the MPLS domain, without the explicit need to manually configure the LSPs. When deviations from this default behavior are desired, import and export policies can be applied.

The configuration is as simple as enabling the LDP protocol instance and adding all network interfaces, for each node. The configuration on node PE-1 is as follows; similar configurations apply on the other nodes.

```
# on PE-1:
configure {
  router "Base" {
    ldp {
      interface-parameters {
        interface "int-PE-1-PE-2" {
          ipv4 {
            }
          }
        interface "int-PE-1-PE-4" {
          ipv4 {
            }
          }
        }
      }
    }
  }
}
```

The **show router ldp discovery** and **show router ldp session** commands can be used to verify the LDP hello adjacencies and sessions. The adjacency type (AdjType) needs to be **Link** while the state should be **Established**. In this example, only IPv4 addresses are used, so the output can be limited to IPv4 only by adding the keyword `ipv4`.

```
[/]
A:admin@PE-1# show router ldp discovery ipv4

=====
LDP IPv4 Hello Adjacencies
=====
Interface Name          Local Addr          Peer Addr          State
AdjType
-----
int-PE-1-PE-2          192.0.2.1:0        192.0.2.2:0        Estab
Link
int-PE-1-PE-4          192.0.2.1:0        192.0.2.4:0        Estab
Link
-----
No. of IPv4 Hello Adjacencies: 2
=====

[/]
A:admin@PE-1# show router ldp session family ipv4

=====
```

```
LDP IPv4 Sessions
=====
Peer LDP Id          Adj Type  State           Msg Sent  Msg Recv  Up Time
-----
192.0.2.2:0         Link      Established     84        85        0d 00:03:14
192.0.2.4:0         Link      Established     74        76        0d 00:02:50
-----
No. of IPv4 Sessions: 2
=====
```

The **show router ldp bindings prefixes** command displays the contents of the LIB (Label Information Base) and contains all labels locally generated (IngLbl) and those received from any LDP neighbors (EgrLbl), whether they are in use or not. The following output is for IPv4 prefixes:

```
[/]
A:admin@PE-1# show router ldp bindings prefixes ipv4

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP IPv4 Prefix Bindings
=====
Prefix
Peer          FEC-Flags
IgrLbl        EgrLbl
EgrNextHop    EgrIntf/LspId
-----
192.0.2.1/32
192.0.2.2:0
524287U      --
--          --

192.0.2.1/32
192.0.2.4:0
524287U      --
--          --

192.0.2.2/32
192.0.2.2:0
--          524287
192.168.12.2 1/1/1

192.0.2.2/32
192.0.2.4:0
524286U      524285
--          --

192.0.2.3/32
192.0.2.2:0
524285N      524285
192.168.12.2 1/1/1

192.0.2.3/32
192.0.2.4:0
```

```

524285U                               524284
--                                   --

192.0.2.4/32
192.0.2.2:0
524284U                               524284
--                                   --

192.0.2.4/32
192.0.2.4:0
--                                   524287
192.168.14.2                          1/1/2

192.0.2.5/32
192.0.2.2:0
524283N                               524283
192.168.12.2                          1/1/1

192.0.2.5/32
192.0.2.4:0
524283U                               524283
--                                   --

192.0.2.6/32
192.0.2.2:0
524282N                               524282
192.168.12.2                          1/1/1

192.0.2.6/32
192.0.2.4:0
524282U                               524282
--                                   --

-----
No. of IPv4 Prefix Bindings: 12
=====

```

The **show router ldp bindings active prefixes** command displays the content of the Label Forwarding Information Base (LFIB) and contains all active labels and the associated label actions used for label switching packets. The active LDP bindings for IPv4 prefixes are the following:

```

[/]
A:admin@PE-1# show router ldp bindings active prefixes ipv4

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
              (IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static           (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop     (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding

=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op

```

IngLbl EgrNextHop	EgrLbl EgrIf/LspId
192.0.2.1/32 524287 --	Pop -- --
192.0.2.2/32 -- 192.168.12.2	Push 524287 1/1/1
192.0.2.2/32 524286 192.168.12.2	Swap 524287 1/1/1
192.0.2.3/32 -- 192.168.12.2	Push 524285 1/1/1
192.0.2.3/32 524285 192.168.12.2	Swap 524285 1/1/1
192.0.2.4/32 -- 192.168.14.2	Push 524287 1/1/2
192.0.2.4/32 524284 192.168.14.2	Swap 524287 1/1/2
192.0.2.5/32 -- 192.168.12.2	Push 524283 1/1/1
192.0.2.5/32 524283 192.168.12.2	Swap 524283 1/1/1
192.0.2.6/32 -- 192.168.12.2	Push 524282 1/1/1
192.0.2.6/32 524282 192.168.12.2	Swap 524282 1/1/1

No. of IPv4 Prefix Active Bindings: 11	
=====	

In the tunnel table, there are LDP LSPs to all other nodes:

```
[/]
A:admin@PE-1# show router tunnel-table
```

IPv4 Tunnel Table (Router: Base)						
Destination Color	Owner	Encap	TunnelId	Pref	Nexthop	Metric
192.0.2.2/32	ldp	MPLS	65537	9	192.168.12.2	10

```

192.0.2.3/32      ldp      MPLS  65538   9      192.168.12.2  20
192.0.2.4/32      ldp      MPLS  65539   9      192.168.14.2  10
192.0.2.5/32      ldp      MPLS  65540   9      192.168.12.2  20
192.0.2.6/32      ldp      MPLS  65541   9      192.168.12.2  30

```

```

-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====

```

In order to signal PHP with LDP, implicit-null must be configured on the eLER.

```

# on PE-6:
configure {
  router "Base" {
    ldp {
      implicit-null-label true
    }
  }
}

```

The implicit-null is signaled immediately, all related labels are withdrawn and re-advertised with label value of 3. The new label would show up on PE-5 as a swap from the ingress label to an egress label of 3, although label 3 is not pushed on to the frame.

```

[/]
A:admin@PE-5# show router ldp bindings active prefixes prefix 192.0.2.6/32

```

```

=====
LDP Bindings (IPv4 LSR ID 192.0.2.5)
(IPv6 LSR ID ::)
=====

```

Label Status:

```

  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC

```

FEC Flags:

```

  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static           (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop     (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding

```

```

=====
LDP IPv4 Prefix Bindings (Active)
=====

```

Prefix	Op
IngLbl	EgrLbl
EgrNextHop	EgrIf/LspId

192.0.2.6/32	Push
--	3
192.168.56.2	1/1/2

192.0.2.6/32	Swap
524282	3
192.168.56.2	1/1/2

```

-----
No. of IPv4 Prefix Active Bindings: 2
=====

```

Import and export policies

The default label handling behavior is to originate label bindings for the system address and to propagate all FECs received. If this is not the desired behavior, an import/export policy can be applied. An LDP import policy impacts inbound filtering; an LDP export policy impacts outbound filtering. An export policy may be configured to control the set of LDP label bindings advertised by the LER (sending to LDP peers). As such, export policies are used to include additional FECs rather than filtering FECs from those advertised. An import policy can be used to control for which FECs a router will generate labels (accepting from LDP peers). This functionality is not unique to LDP; it can be used for RSVP-TE, OSPF, and IS-IS as well as others.

The policy can be global or LDP peer FEC prefix filtering, both for import and export. LDP peer FEC prefix filtering uses a similar policy context as the LDP global policies and works in addition to these global policies.

```
[/]
A:admin@PE-1# tree flat detail | match import-pref
configure groups group <string> router <string> ldp session-parameters peer <string | ipv4-
address | ipv6-address> import-prefixes <string>
configure groups group <string> router <string> ldp targeted-session import-prefixes <string>
configure router <string> ldp session-parameters peer <ipv4-address | ipv6-address> import-
prefixes <reference>
configure router <string> ldp targeted-session import-prefixes <reference>
```

```
[/]
A:admin@PE-1# tree flat detail | match export-pref
configure groups group <string> router <string> ldp session-parameters peer <string | ipv4-
address | ipv6-address> export-prefixes <string>
configure groups group <string> router <string> ldp targeted-session export-prefixes <string>
configure router <string> ldp session-parameters peer <ipv4-address | ipv6-address> export-
prefixes <reference>
configure router <string> ldp targeted-session export-prefixes <reference>
```

By default, no labels are generated for directly connected (local) interfaces. To change this behavior, an export policy is created and applied to the LDP instance. There is no configuration difference in defining an import and export policy.

A policy contains a list of entries (of which each has a number). An entry typically contains matching criteria (however, it is not required in cases where everything matches) and a corresponding action. Entries without an action are considered incomplete and are rendered inactive. When processing the policy, the router executes the specified action on the first matching statement; it does not process any further matches. For this reason, entries must be sequenced correctly from most to least specific.

The configuration of the LDP export policy for local interfaces is as follows:

```
# on all PEs:
configure {
    policy-options {
        policy-statement "LDP-export" {
            entry 10 {
                from {
                    protocol {
                        name [direct]
                    }
                }
                action {
                    action-type accept
                }
            }
        }
    }
}
```

```

    }
  }
}

```

There are 11 active LDP bindings before applying the export policy, as shown earlier.

The LDP export or import policy is applied to the LDP instance on the router, with the export or import keyword.

```

# on all PEs:
configure {
  router "Base" {
    ldp {
      export-policy ["LDP-export"]
    }
  }
}

```

When the export policy is applied, the active LDP binding table contains additional entries: the local interfaces of PE-x. In the following output, the 11 entries for the system prefixes are snipped; only the 7 additional prefixes are shown:

```

[/]
A:admin@PE-1# show router ldp bindings active prefixes ipv4

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static          (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop    (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----snip-----
192.168.12.0/30                       Pop
524281                                --
--                                     --
192.168.14.0/30                       Pop
524280                                --
--                                     --
192.168.23.0/30                       Swap
524279                                524279
192.168.12.2                          1/1/1
192.168.25.0/30                       Swap
524278                                524278
192.168.12.2                          1/1/1

```

```

192.168.36.0/30          Swap
524277                 524277
192.168.12.2           1/1/1

192.168.45.0/30        Swap
524276                 524276
192.168.14.2           1/1/2

192.168.56.0/30        Swap
524275                 524275
192.168.12.2           1/1/1

-----
No. of IPv4 Prefix Active Bindings: 18
=====

```

OAM

The following operations, administration, and maintenance operations can be launched on an LDP LSP (in classic CLI):

- oam lsp-ping
- oam lsp-trace

As an example, an LSP ping is sent from PE-1 to PE-6:

```

A:PE-1# oam lsp-ping prefix 192.0.2.6/32
LSP-PING 192.0.2.6/32: 80 bytes MPLS payload
Seq=1, send from intf int-PE-1-PE-2, reply from 192.0.2.6
  udp-data-len=32 ttl=255 rtt=4.29ms rc=3 (EgressRtr)

---- LSP 192.0.2.6/32 PING Statistics ----
1 packets sent, 1 packets received, 0.00% packet loss
round-trip min = 4.29ms, avg = 4.29ms, max = 4.29ms, stddev = 0.000ms

```

An LSP trace is sent from PE-1 to PE-6:

```

A:PE-1# oam lsp-trace prefix 192.0.2.6/32
lsp-trace to 192.0.2.6/32: 0 hops min, 0 hops max, 104 byte packets
1 192.0.2.2 rtt=3.58ms rc=8(DSRtrMatchLabel) rsc=1
2 192.0.2.3 rtt=3.89ms rc=8(DSRtrMatchLabel) rsc=1
3 192.0.2.6 rtt=4.63ms rc=3(EgressRtr) rsc=1

```

The return code (**rc**) is 8 for the LSRs and 3 for the eLER.

The detailed output for this LSP trace includes the interface IP address, the interface type, maximum receive unit (MRU), label and protocol; as follows:

```

A:PE-1# oam lsp-trace prefix 192.0.2.6/32 detail
lsp-trace to 192.0.2.6/32: 0 hops min, 0 hops max, 104 byte packets
1 192.0.2.2 rtt=2.44ms rc=8(DSRtrMatchLabel) rsc=1
   DS 1: ipaddr=192.168.23.2 ifaddr=192.168.23.2 iftype=ipv4Numbered MRU=1564
         label[1]=524282 protocol=3(LDP)
2 192.0.2.3 rtt=3.67ms rc=8(DSRtrMatchLabel) rsc=1
   DS 1: ipaddr=192.168.36.2 ifaddr=192.168.36.2 iftype=ipv4Numbered MRU=1564
         label[1]=3 protocol=3(LDP)
3 192.0.2.6 rtt=4.51ms rc=3(EgressRtr) rsc=1

```


LDP statistics

LDP-related statistics can be collected in files. On PE-1, file 1 is configured as follows:

```
# on PE-1:
configure {
  log {
    file 1 {
      rollover 5
      retention 1
      compact-flash-location {
        primary cf1
      }
    }
  }
}
```

The following accounting policy defines which statistics should be recorded:

```
# on PE-1:
configure {
  log {
    accounting-policy 1 {
      admin-state enable
      record combined-ldp-lsp-egress
      destination {
        file 1
      }
    }
  }
}
```

The collection of statistics for prefix 192.0.2.6/32 is enabled on PE-1 in the **ldp** context, as follows:

```
# on PE-1:
configure {
  router "Base" {
    ldp {
      egress-statistics {
        fec-prefix 192.0.2.6/32 {
          admin-state enable
          accounting-policy 1
          collect-stats true
        }
      }
    }
  }
}
```

FEC egress statistics can be displayed with the following **show** command:

```
[/]
A:admin@PE-1# show router ldp fec-egress-stats
```

The FEC egress stats for prefix 192.0.2.6/32 can be retrieved as follows:

```
[/]
A:admin@PE-1# show router ldp fec-egress-stats 192.0.2.6/32
```

```
=====
LDP IPv4 FEC Egress Statistics
=====
```

```
-----
FEC Prefix/Mask      : 192.0.2.6/32
-----
```

```
Collect Stats       : Enabled                Accounting Plcy.    : 1
```

```

Admin State      : Up
FC BE
InProf Pkts     : 0          OutProf Pkts      : 7
InProf Octets   : 0          OutProf Octets    : 858
FC L2
InProf Pkts     : 0          OutProf Pkts      : 0
InProf Octets   : 0          OutProf Octets    : 0
FC AF
InProf Pkts     : 0          OutProf Pkts      : 0
InProf Octets   : 0          OutProf Octets    : 0
FC L1
InProf Pkts     : 0          OutProf Pkts      : 0
InProf Octets   : 0          OutProf Octets    : 0
FC H2
InProf Pkts     : 0          OutProf Pkts      : 0
InProf Octets   : 0          OutProf Octets    : 0
FC EF
InProf Pkts     : 0          OutProf Pkts      : 0
InProf Octets   : 0          OutProf Octets    : 0
FC H1
InProf Pkts     : 0          OutProf Pkts      : 0
InProf Octets   : 0          OutProf Octets    : 0
FC NC
InProf Pkts     : 0          OutProf Pkts      : 0
InProf Octets   : 0          OutProf Octets    : 0

Aggregate Packets : 7
Aggregate Octets  : 858
=====
LDP IPv4 FEC Egress Statistics: 1
=====

```

Statistics can be cleared as follows on PE-1:

```

[/]
A:admin@PE-1# clear router ldp fec-egress-statistics 192.0.2.6/32

```

Debug

LDP debugging can be configured—in classic CLI—per LDP interface or per LDP peer, as follows:

```

# on PE-1:
debug
  router "Base"
    ldp ?
  - ldp
  - no ldp

[no] interface      + Enable/disable and configure debugging for an LDP interface
[no] peer           + Enable/disable and configure debugging for an LDP peer

```

A particular peer is specified by its IPv4 or IPv6 address. It is possible to configure debugging for specific LDP events: bindings or messages, as follows:

```

# on PE-1:
debug
  router "Base"
    ldp
      peer 192.0.2.2 ?

```

```

- no peer <ip-address>
- peer <ip-address>

<ip-address>      : ipv4-address   - a.b.c.d
                   ipv6-address  - x:x:x:x:x:x:x:x (eight 16-bit pieces)
                                     x:x:x:x:x:x:d.d.d.d
                                     x - [0..FFFF]H
                                     d - [0..255]D

[no] event        + Configure debugging for specific LDP events
[no] packet       + Enable/disable debugging for specific LDP packets

```

```

# on PE-1:
debug
  router "Base"
    ldp
      peer 192.0.2.2
        event ?
    - event
    - no event

[no] bindings     - Enable/disable debugging for LDP bindings
[no] messages     - Enable/disable debugging for LDP messages

```

It is also possible to configure debugging for specific packets, such as label packets:

```

# on PE-1:
debug
  router "Base"
    ldp
      peer 192.0.2.2
        packet ?
    - no packet
    - packet

[no] hello        - Enable/disable debugging for LDP Hello packets
[no] init         - Enable/disable debugging for LDP Init packets
[no] keepalive    - Enable/disable debugging for LDP Keepalive packets
[no] label        - Enable/disable debugging for LDP Label packets

```

The following debugging is configured on PE-1:

```

# on PE-1:
debug
  router "Base"
    ldp
      peer 192.0.2.2
        packet
          label detail
        exit
      exit
    exit

```

Some label mapping packets sent to peer 192.0.2.2 are the following, with the label mapping for prefixes 192.0.2.1/32 and 192.0.2.4/32:

```

---snip---

5 2021/02/18 15:52:15.241 CET MINOR: DEBUG #2001 Base LDP
"LDP: LDP
Send Label Mapping packet (msgId 14) to 192.0.2.2:0

```

```
Protocol version = 1
Label 524284 advertised for the following FECs
Prefix Address Family = 1 Prefix = 192.0.2.4/32
"
4 2021/02/18 15:52:04.040 CET MINOR: DEBUG #2001 Base LDP
"LDP: LDP
Send Label Mapping packet (msgId 12) to 192.0.2.2:0
Protocol version = 1
Label 524285 advertised for the following FECs
Prefix Address Family = 1 Prefix = 192.0.2.3/32
"
3 2021/02/18 15:52:04.040 CET MINOR: DEBUG #2001 Base LDP
"LDP: LDP
Recv Label Mapping packet (msgId 12) from 192.0.2.2:0
Protocol version = 1
Label 524285 advertised for the following FECs
Prefix Address Family = 1 Prefix = 192.0.2.3/32
"
2 2021/02/18 15:51:48.931 CET MINOR: DEBUG #2001 Base LDP
"LDP: LDP
Recv Label Mapping packet (msgId 6) from 192.0.2.2:0
Protocol version = 1
Label 524287 advertised for the following FECs
Prefix Address Family = 1 Prefix = 192.0.2.2/32
"
1 2021/02/18 15:51:48.681 CET MINOR: DEBUG #2001 Base LDP
"LDP: LDP
Send Label Mapping packet (msgId 6) to 192.0.2.2:0
Protocol version = 1
Label 524287 advertised for the following FECs
Prefix Address Family = 1 Prefix = 192.0.2.1/32
"
```

Conclusion

MPLS provides the capability to establish connection-oriented paths over a connectionless network. LDP point-to-point LSPs are dynamically signaled and FRR is supported. This can greatly improve network resiliency. In this chapter, the configuration of several LDP point-to-point LSP features is given together with the associated show output which can be used to verify and troubleshoot.

LDP-IGP Synchronization

This chapter provides information about LDP-IGP synchronization

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 14.0.R6, but the MD-CLI in the current edition is based on SR OS Release 21.2.R1.

Label Distribution Protocol - Interior Gateway Protocol (LDP-IGP) synchronization based on RFC 5443 is supported in SR OS Release 6.0, and later. LDP end-of-Label Information Base (LIB), as described in RFC 5919, is supported in SR OS Release 14.0.R1, and later.

Overview

Within an MPLS network using LDP, it is common practice to enable a synchronization timer between LDP and the IGP to give both the IGP and LDP time to converge after a link is restored. Without LDP-IGP synchronization, the IGP and LDP converge independently. Because the IGP converges before LDP, traffic can be black-holed until LDP has converged. When the IGP converges after link restoration and a new next hop is available, this change in next hop causes LDP to stop using the LDP labels for the alternate path. After the adjacency with the new next hop is established, labels are allocated for the new shortest (primary) path. These new labels are not yet signaled by LDP, causing the traffic to be black-holed for all or part of the FECs until LDP converges.

LDP-IGP synchronization based on RFC 5443 consists of temporarily setting the run-time IGP cost of a restored link to infinity to give time for both IGP and LDP to converge. When the LDP synchronization timer expires, the runtime IGP cost is restored to the configured IGP cost and IGP will re-advertise it and use this for the next shortest path first (SPF) computation. The value for infinity of the IGP cost for a router interface depends on the IGP: 0xFFFF (65535) for OSPF, 0x3F (63) for IS-IS regular metric, and 0xFFFFFE (16777214) for IS-IS wide metric. LDP-IGP synchronization is not supported on RIP interfaces.

When the system converges, the IGP starts the LDP synchronization timer when the LDP session to the neighbor is established over the interface. The LDP synchronization timer is running during the exchange of label FEC bindings over the interface. When the LDP synchronization timer expires, the IGP announces the new best next hop and LDP uses this next hop if the label bindings for the neighbor's FEC are available. However, the LDP synchronization timer does not guarantee that all FEC bindings will be exchanged when the timer expires. Operators do not want to configure very large timers on every node, which may result in long synchronization times. The end-of-lib option (RFC 5919) reduces the synchronization time; therefore, operators can configure large synchronization timers that will be aborted when the end-of-lib notification has been received from a downstream node.

By default, LDP-IGP synchronization is enabled for OSPF and for IS-IS, as follows:

```
[/]
A:admin@PE-1# configure {
  router "Base" {
    ospf {
      info detail | match ldp-sync
    }
  }
}

shows:    ldp-sync true
```

```
[/]
A:admin@PE-1# configure {
  router "Base" {
    isis 0 {
      info detail | match ldp-sync
    }
  }
}

shows:    ldp-sync true
```

By default, LDP synchronization is disabled (out-of-service) on each interface, as follows:

```
[/]
A:admin@PE-1# show router ospf interface "int-PE-1-P-2" detail | match Ldp
Ldp Sync      : outOfService      Ldp Sync Wait   : Disabled
Ldp Timer State : Disabled        Ldp Tm Left     : 0
```

```
[/]
A:admin@PE-1# show router isis interface "int-PE-1-P-2" detail | match Ldp
Ldp Sync      : outOfService      Ldp Sync Wait   : Disabled
Ldp Timer State : Disabled        Ldp Tm Left     : 0
```

LDP end-of-lib, as defined in RFC 5919, allows a downstream node to notify its upstream peer that the node has advertised its entire LIB to its upstream peer, which can terminate the LDP synchronization timer. LDP end-of-lib notifications use a FEC TLV with the type wildcard FEC element for all negotiated FEC types. LDP end-of-lib is sent even if the system has no label bindings to advertise. Each node notifies its peer nodes that it is safe to send LDP end-of-lib notifications even if the node is not configured to process them. The node sends an unrecognized notification capability TLV (RFC 5919) in the initialization message, indicating that it will ignore notification messages that carry status TLV with a non-fatal status code unknown to it.

The LDP synchronization timer is configured in seconds with a maximum of 1800 seconds on a per interface basis, as follows:

```
[/]
A:admin@PE-1# configure {
  router "Base" {
    interface "int-PE-1-P-2" {
      ldp-sync-timer ?
    }
  }
}

ldp-sync-timer

end-of-lib      - Terminate LDP sync timer
seconds         - Value for the LDP SYNC timer
```

As an example, an LDP synchronization timer of 300 seconds can be configured on interface "int-PE-1-P-2", with or without the LDP end-of-lib option, as follows:

```
# on PE-1:
[/]
A:admin@PE-1# configure {
  router "Base" {
    interface "int-PE-1-P-2" {
      ldp-sync-timer {
        seconds 300
      }
    }
  }
}
```

```
# on PE-1:
[/]
A:admin@PE-1# configure {
  router "Base" {
    interface "int-PE-1-P-2" {
      ldp-sync-timer {
        seconds 300
        end-of-lib true
      }
    }
  }
}
```

- When the end-of-lib option is not configured, the LDP synchronization timer is started when the LDP hello adjacency comes up over the interface. Any received LDP end-of-lib message is ignored.
- When the end-of-lib option is configured, the receiving node behaves as follows:
 - The LDP synchronization timer is started when the LDP hello adjacency comes up over the interface.
 - When LDP end-of-lib type wildcard FEC messages have been received for all negotiated FEC types for a certain session to an LDP peer for the IGP interface, the LDP synchronization timer is terminated and the system restores the IGP link cost.
 - If the LDP synchronization timer expires before the LDP end-of-lib messages are received for all negotiated FEC types, the system restores the IGP link cost.
 - All unexpected LDP end-of-lib messages are dropped.
- When the end-of-lib option is configured, the sending node will advertise an LDP end-of-lib message for all FECs (prefix and P2MP FECs) after all FECs are sent for all peers that have advertised the unrecognized notification capability TLV.

When a user changes the IGP cost of an interface, the new value is advertised at the next flooding of link attributes by the IGP. If the LDP synchronization timer is running, the new cost value will only be advertised after the timer expires. However, the following **tools** or **configure** commands can be used to terminate the LDP-IGP synchronization, causing the new IGP cost value to be advertised instantly.

The following two **tools** commands do not modify the configuration; they terminate the LDP synchronization timer and restore the actual cost of the IGP interface:

```
[/]
A:admin@PE-1# tools perform router ospf ldp-sync-exit

[/]
```

```
A:admin@PE-1# tools perform router isis ldp-sync-exit
```

The following three commands disable the LDP-IGP synchronization entirely, either from the interface or globally for the IGP (OSPF or IS-IS):

```
# on PE-1:  
[/  
A:admin@PE-1# configure {  
  router "Base" {  
    interface "int-PE-1-P-2" {  
      delete ldp-sync-timer  
    }  
  }  
}
```

```
[/  
A:admin@PE-1# configure {  
  router "Base" {  
    ospf 0 {  
      ldp-sync false  
    }  
  }  
}
```

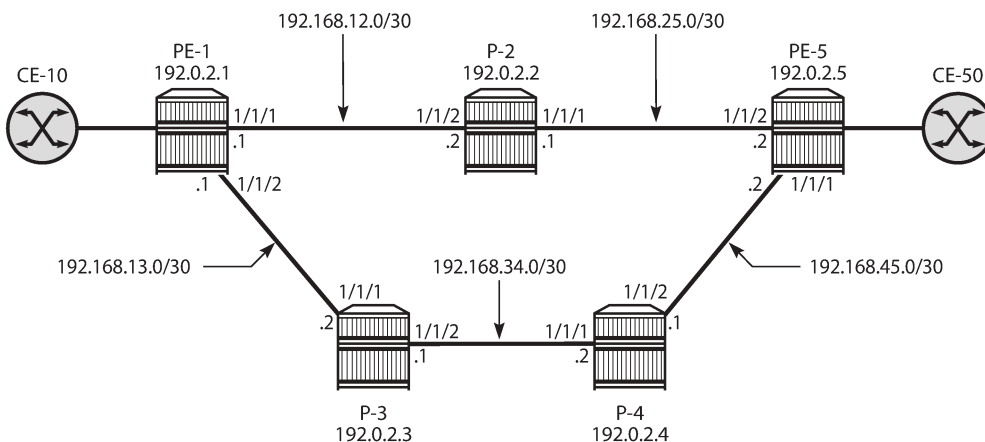
```
[/  
A:admin@PE-1# configure {  
  router "Base" {  
    isis 0 {  
      ldp-sync false  
    }  
  }  
}
```

If the user changes the value of the LDP synchronization timer parameter, the new value will take effect at the next synchronization event. If the timer is still running, it will continue to use the previous value.

Configuration

[Figure 229: Example topology](#) shows the example topology.

Figure 229: Example topology



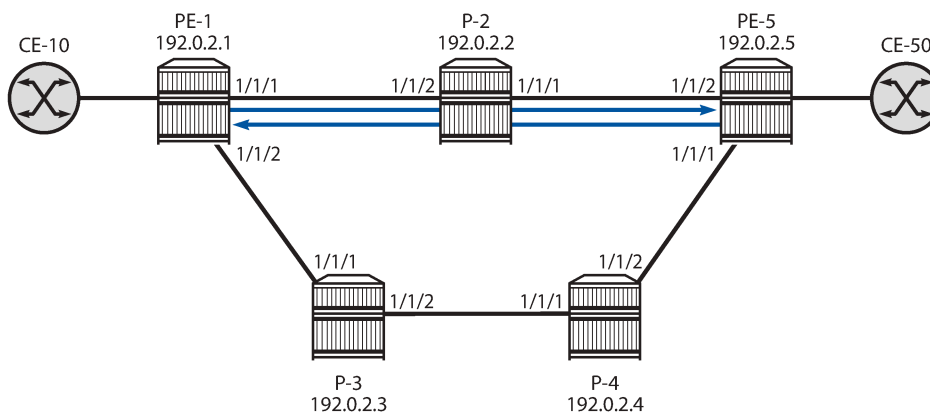
26227

The initial configuration on these nodes includes the following:

- Cards, MDAs, ports
- Router interfaces
- IGP: OSPF on all interfaces between the five P/PE routers (alternatively, IS-IS can be configured)
- LDP on all interfaces (LDP link adjacencies)
- Services on the PEs; for example, an Epipe between PE-1 and PE-5 (LDP targeted adjacencies)
- In this example topology, CE-10 and CE-50 correspond to VPRN_10_name on PE-1 and PE-5 using a hairpin to loop the traffic back to the node.

Default IGP metrics are used on the interfaces and, under normal conditions, traffic between CE-10 and CE-50 is sent over the shortest path via P-2, as shown in [Figure 230: Shortest path between PE-1 and PE-5](#).

Figure 230: Shortest path between PE-1 and PE-5



26228

LDP-IGP synchronization without LDP end-of-lib

LDP-IGP synchronization is, by default, globally enabled for OSPF and IS-IS, but disabled on every interface. In this example, LDP-IGP synchronization will be configured with an LDP synchronization timer of 300 seconds on all the interfaces in all the nodes, as follows:

```
# on PE-1:
[/]
A:admin@PE-1# configure {
  router "Base" {
    interface "int-PE-1-P-2" {
      ldp-sync-timer {
        seconds 300
      }
    }
    interface "int-PE-1-P-3" {
      ldp-sync-timer {
        seconds 300
      }
    }
  }
}
```

```
# on P-2:
[/]
A:admin@P-2# configure {
  router "Base" {
    interface "int-P-2-PE-1" {
      ldp-sync-timer {
        seconds 300
      }
    }
    interface "int-P-2-PE-5" {
      ldp-sync-timer {
        seconds 300
      }
    }
  }
}
```

The configuration is similar on the other nodes. With this configuration, a restored interface will temporarily get an IGP cost of infinity; therefore, the link will not be used for data traffic until the LDP synchronization timer terminates (when it expires after 300 seconds or when it is terminated manually). To simulate a link failure, port 1/1/1 is disabled and re-enabled on PE-1, as follows:

```
# on PE-1:
[/]
A:admin@PE-1# configure {
  port 1/1/1 {
    admin-state disable
  }
}

[/]
A:admin@PE-1# configure {
  port 1/1/1 {
    admin-state enable
  }
}
```

The LDP synchronization timer is not started before the LDP hello adjacency is established. The following output shows the port re-enabled, but before the LDP adjacency is established (Ldp Timer State = Wait for Ldp Adj.):

```
[/]
A:admin@PE-1# show router ospf interface "int-PE-1-P-2" detail | match Ldp
Ldp Sync      : inService      Ldp Sync Wait   : Disabled
Ldp Timer State : Wait for Ldp Adj.  Ldp Tm Left    : 0
```

The following debug messages for OSPF show that the OSPF interface state is up (point-to-point), the LDP Sync Timer state is updated to "WAIT_FOR_ADJ", and afterward the LDP state is updated to "LDP_INTF_HAS_ADJ", as follows:

```
21 2021/07/30 16:36:32.769 CEST MINOR: DEBUG #2001 Base OSPFv2
"OSPFv2: INTF
IF 192.168.12.1 Idx 2 Event: IF_UP state: from DOWN to PTP"

24 2021/07/30 16:36:32.769 CEST MINOR: DEBUG #2001 Base OSPFv2
"OSPFv2: INTF
Updated the LDP Sync Timer state for I/F 2 to WAIT_FOR_ADJ"

26 2021/07/30 16:36:32.869 CEST MINOR: DEBUG #2001 Base OSPFv2
"OSPFv2: INTF
OSPF I/F 2 LDP state: new LDP_INTF_HAS_ADJ old LDP_INTF_DOWN"
```

When the LDP hello adjacency is established, the interface between PE-1 and P-2 gets an IGP cost of infinity and the LDP synchronization timer is started, as follows:

```
27 2021/07/30 16:36:32.869 CEST MINOR: DEBUG #2001 Base OSPFv2
"OSPFv2: INTF
Updated the LDP Sync Timer state for I/F 2 to TMR_ACTIVE"
```

LDP bindings are exchanged as follows, but no message indicates the end-of-lib (and if it were sent by P-2, it would be ignored by PE-1). The LDP synchronization timer is not automatically terminated when the LDP bindings are received, because the configuration does not include the end-of-lib option.

```
29 2021/07/30 16:36:32.912 CEST MINOR: DEBUG #2001 Base LDP
"LDP: Binding
Sending Label mapping label 524287 for Prefix Address Family = 1 Prefix = 192.0.2.1/32 to peer
192.0.2.2:0."

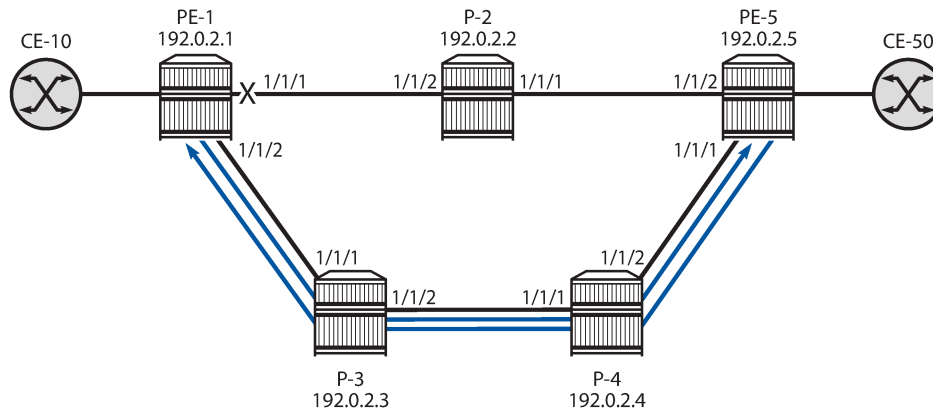
35 2021/07/30 16:36:32.916 CEST MINOR: DEBUG #2001 Base LDP
"LDP: Binding
Sending Label mapping label 524284 for Prefix Address Family = 1 Prefix = 192.0.2.3/32 to peer
192.0.2.2:0."

37 2021/07/30 16:36:32.916 CEST MINOR: DEBUG #2001 Base LDP
"LDP: Binding
Sending Label mapping label 524283 for Prefix Address Family = 1 Prefix = 192.0.2.4/32 to peer
192.0.2.2:0."

39 2021/07/30 16:36:32.916 CEST MINOR: DEBUG #2001 Base LDP
"LDP: Binding
Sending Label mapping label 524282 for Prefix Address Family = 1 Prefix = 192.0.2.5/32 to peer
192.0.2.2:0."
```

As long as the LDP synchronization timer is not terminated, traffic between CE-10 and CE-50 is redirected to the path via P-3 and P-4, as shown in [Figure 231: Rerouting via P-3 and P-4 until LDP synchronization timer terminates](#).

Figure 231: Rerouting via P-3 and P-4 until LDP synchronization timer terminates



26229

The following commands for the OSPF interfaces between PE-1 and P-2 show the Ldp Timer State = Timer Active, Ldp Sync Wait = Enabled; therefore, traffic is rerouted and the remaining time (Ldp Tm Left):

```
[/]
A:admin@PE-1# show router ospf interface "int-PE-1-P-2" detail | match Ldp
Ldp Sync      : inService          Ldp Sync Wait   : Enabled
Ldp Timer State : Timer Active      Ldp Tm Left     : 275
```

```
[/]
A:admin@P-2# show router ospf interface "int-P-2-PE-1" detail | match Ldp
Ldp Sync      : inService          Ldp Sync Wait   : Enabled
Ldp Timer State : Timer Active      Ldp Tm Left     : 263
```

The restored interface between PE-1 and P-2 will have an infinite IGP cost, so will not be used for data traffic as long as the LDP synchronization timer is active. All traffic between the CEs takes the path via P-3 and P-4, which can be verified as follows. The port statistics are cleared and 100 ICMP echo requests are sent by CE-10 to CE-50. On PE-1, port 1/1/1 is used toward P-2 and port 1/1/2 is used toward P-3. All traffic is expected to take the path toward P-3. However, there will be some IGP and LDP signaling on all interfaces, so the packet count will be greater than 100, as follows:

```
[/]
A:admin@PE-1# clear port 1/1/1 statistics
```

```
[/]
A:admin@PE-1# clear port 1/1/2 statistics
```

```
[/]
A:admin@PE-1# ping 172.16.10.2 router-instance "VPRN_10_name" count 100
PING 172.16.10.2 56 data bytes
---snip---

---- 172.16.10.2 PING Statistics ----
100 packets transmitted, 100 packets received, 0.00% packet loss
```

```
round-trip min = 3.38ms, avg = 4.40ms, max = 4.71ms, stddev = 0.249ms
```

```
[/]
A:admin@PE-1# show port 1/1/1 statistics
```

```
=====  
Port Statistics on Slot 1  
=====
```

Port Id	Ingress Packets Egress Packets	Ingress Octets Egress Octets
1/1/1	81 81	9005 9005

```
[/]
A:admin@PE-1# show port 1/1/2 statistics
```

```
=====  
Port Statistics on Slot 1  
=====
```

Port Id	Ingress Packets Egress Packets	Ingress Octets Egress Octets
1/1/2	275 275	29165 28984

The port statistics on the other nodes will also show that these packets are sent via P-3 and P-4 instead of via P-2.

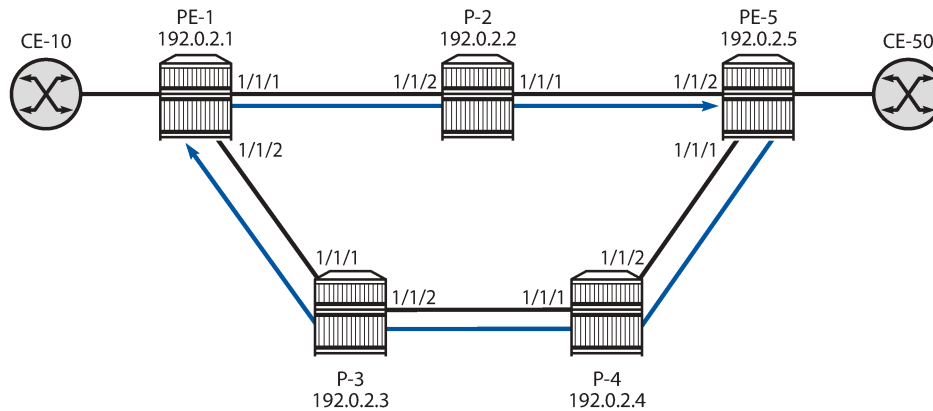
Even though the LIB was exchanged within seconds, the restored link only gets its normal IGP cost after the LDP synchronization timer has terminated. This can be done manually for a specific IGP (in this example, for OSPF on interface "int-PE-1-P-2" on PE-1) as follows:

```
[/]
A:admin@PE-1# tools perform router ospf ldp-sync-exit
Done.

[/]
A:admin@PE-1# show router ospf interface "int-PE-1-P-2" detail | match Ldp
Ldp Sync      : inService      Ldp Sync Wait   : Disabled
Ldp Timer State : Manual Exit      Ldp Tm Left     : 0
```

The LDP synchronization timer can be configured independently for each IGP on each interface. The LDP synchronization timer for OSPF on interface "int-PE-1-P-2" is terminated manually (Ldp Timer State = Manual Exit; Ldp Sync Wait = Disabled; Ldp Tm Left = 0). Traffic from CE-10 to CE-50 can use interface "int-PE-1-P-2" because that interface has its configured (default) IGP cost. However, traffic from CE-50 to CE-10 will not use interface "int-P-2-PE-1" because that interface still has an infinite IGP cost as long as the LDP synchronization timer is not terminated; therefore, traffic toward CE-10 will pass via P-3 instead. This leads to an asymmetric traffic flow: the shortest path from CE-10 to CE-50 is via P-2, while the shortest path from CE-50 to CE-10 is via P-4 and P-3, as shown in [Figure 232: Restored link with one LDP synchronization timer terminated](#).

Figure 232: Restored link with one LDP synchronization timer terminated



26230

When the second LDP synchronization timer is also terminated, the shortest path is via P-2 for all traffic between CE-10 and CE-50.

The LDP synchronization timer needs to be configured to a value that is long enough to prevent traffic being black-holed, but not too long to cause unnecessary suboptimal routing after the LIB has been exchanged and before the termination of the LDP synchronization timer. The end-of-lib option reduces the LDP synchronization time when the configured LDP synchronization timer is longer than required for the exchange of the LIB, as described in the next section.

LDP synchronization is disabled on the interfaces of PE-1, as follows:

```
# on PE-1:
[/]
A:admin@PE-1# configure {
  router "Base" {
    interface "int-PE-1-P-2" {
      delete ldp-sync-timer
    }
    interface "int-PE-1-P-3" {
      delete ldp-sync-timer
    }
  }
}
```

Similar commands to disable LDP synchronization on an interface can be configured on the other nodes.

LDP-IGP synchronization with LDP end-of-lib

The LDP synchronization is configured with the end-of-lib option on all interfaces on all nodes; for example, for PE-1, as follows:

```
# on PE-1:
[/]
A:admin@PE-1# configure {
  router "Base" {
    interface "int-PE-1-P-2" {
      ldp-sync-timer {
        seconds 300
      }
    }
  }
}
```

```

        end-of-lib true
    }
}
interface "int-PE-1-P-3" {
    ldp-sync-timer {
        seconds 300
        end-of-lib true
    }
}
}
}
}

```

The configuration on the other nodes is similar.

A link failure is simulated by disabling and re-enabling port 1/1/1 on PE-1. Initially, the Ldp Timer State is "Wait for Ldp Adj.", as follows:

```

# on PE-1:
[/]
A:admin@PE-1# configure {
    port 1/1/1 {
        admin-state disable
    }
}

```

```

[/]
A:admin@PE-1# configure {
    port 1/1/1 {
        admin-state enable
    }
}

```

```

[/]
A:admin@PE-1# show router ospf interface "int-PE-1-P-2" detail | match Ldp
Ldp Sync      : inService      Ldp Sync Wait   : Disabled
Ldp Timer State : Wait for Ldp Adj.    Ldp Tm Left     : 0

```

After the LDP hello adjacency is established on the restored link, the LDP synchronization timer is started and PE-1 sends all LDP bindings to its peer P-2, as follows:

```

26 2021/07/30 16:49:16.069 CEST MINOR: DEBUG #2001 Base OSPFv2
"OSPFv2: INTF
OSPF I/F 2 LDP state: new LDP_INTF_HAS_ADJ old LDP_INTF_DOWN"

27 2021/07/30 16:49:16.069 CEST MINOR: DEBUG #2001 Base OSPFv2
"OSPFv2: INTF
Updated the LDP Sync Timer state for I/F 2 to TMR_ACTIVE"

29 2021/07/30 16:49:16.162 CEST MINOR: DEBUG #2001 Base LDP
"LDP: Binding
Sending Label mapping label 524287 for Prefix Address Family = 1 Prefix = 192.0.2.1/32 to peer
192.0.2.2:0."

35 2021/07/30 16:49:16.165 CEST MINOR: DEBUG #2001 Base LDP
"LDP: Binding
Sending Label mapping label 524284 for Prefix Address Family = 1 Prefix = 192.0.2.3/32 to peer
192.0.2.2:0."

37 2021/07/30 16:49:16.165 CEST MINOR: DEBUG #2001 Base LDP
"LDP: Binding

```

```
Sending Label mapping label 524283 for Prefix Address Family = 1 Prefix = 192.0.2.4/32 to peer
192.0.2.2:0."

39 2021/07/30 16:49:16.165 CEST MINOR: DEBUG #2001 Base LDP
"LDP: Binding
Sending Label mapping label 524282 for Prefix Address Family = 1 Prefix = 192.0.2.5/32 to peer
192.0.2.2:0."

43 2021/07/30 16:49:16.413 CEST MINOR: DEBUG #2001 Base OSPFv2
"OSPFv2: INTF
OSPF I/F 2 LDP state: new LDP_LBL_EXCH_DONE old LDP_INTF_HAS_ADJ"

44 2021/07/30 16:49:16.413 CEST MINOR: DEBUG #2001 Base OSPFv2
"OSPFv2: INTF
Updated the LDP Sync Timer state for I/F 2 to EXCH_DONE"
```

When a downstream node has sent its entire LIB to its upstream peer, the node sends an end-of-lib (RFC 5919) notification. When the upstream peer receives an end-of-lib notification from its downstream peer, LDP is considered to be fully operational for the link. LDP triggers the IGP to advertise the link with normal cost instead of infinity and transit traffic can be sent on the restored link. In the preceding debug messages, the LDP Sync Timer state changes to "EXCH_DONE"; in the following show command output:

```
[/]
A:admin@PE-1# show router ospf interface "int-PE-1-P-2" detail | match Ldp
Ldp Sync      : inService      Ldp Sync Wait   : Disabled
Ldp Timer State : Label Exchg. Done  Ldp Tm Left    : 0
```

The LDP synchronization timer is terminated when the entire LIB is exchanged. In this example setup, the LDP synchronization time is reduced from 300 seconds to less than 10 seconds after enabling LDP end-of-lib.

Conclusion

LDP-IGP synchronization (RFC 5443) allows directly connected nodes to delay the use of a restored link for transit IP packets until the LDP labels have been exchanged. RFC 5919 adds the end-of-lib option that reduces the LDP synchronization time to the minimum, so operators can configure large values for the LDP synchronization timer.

LDP-SR Stitching for IPv4 Prefixes (IS-IS)

This chapter provides information about LDP-SR Stitching for IPv4 Prefixes (IS-IS).

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 14.0.R5. The MD-CLI in the current edition is based on SR OS Release 21.2.R1.

Overview

Segment Routing (SR) allows for the construction of source-routed Label Switched Paths (LSPs) where the series of hops to be taken through the network are indicated by one or more Segment Identifiers (SIDs) assigned at the ingress PE. In the case of an MPLS data plane, these SIDs are MPLS labels learned through extensions to the OSPF/IS-IS control plane. SR provides benefits to the MPLS data plane, such as high scalability (due to lack of soft-state), traffic engineering capability, and topology-independent fast reroute.

When SR is configured in an IP/MPLS network that runs the Label Distribution Protocol (LDP), it is possible that SR and LDP will coexist, in which case preference for LDP or SR is a local matter at the LSP head end. It is equally possible that not all devices will have the capability to support SR, in which case some kind of interworking between SR and LDP is necessary to create an end-to-end LSP. Fast reroute coverage can also benefit from this SR-LDP interworking function, where SR is used to increase Loop Free Alternate (LFA) coverage using Remote or Directed LFA.

This chapter describes the configuration requirements for the interworking of LDP and SR to form a single end-to-end LSP when using IS-IS as an IGP. The chapter shows how this interworking function can be used to extend fast reroute coverage for LDP-based LSPs.

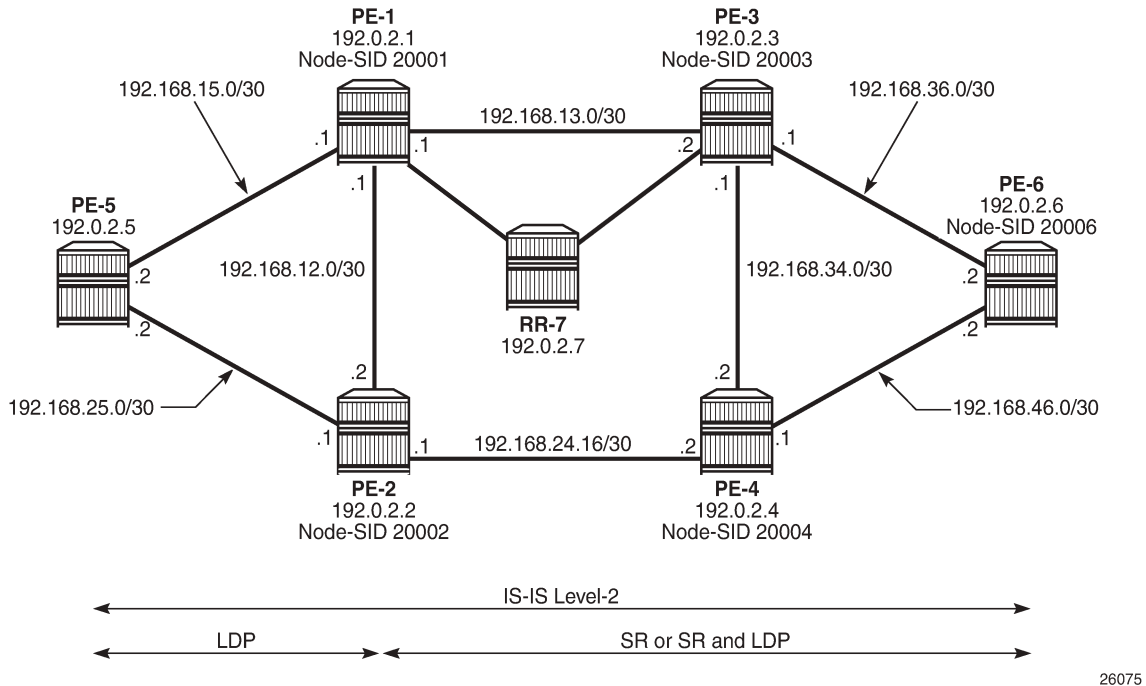
Configuration

Example topology

The topology shown in [Figure 233: Example topology](#) provides an example of SR-LDP interworking. All routers within the topology form part of Autonomous System 64496 and are IBGP clients of RR-7 for the VPN-IPv4 address family. All routers in the topology belong to the same IS-IS Level-2 area, and all link metrics are set to 100. RR-7 does not participate in any MPLS data plane, and signals the IS-IS overload bit to avoid being used for transit traffic.

PE-5 is a router that does not support SR and, therefore, runs only LDP to its connected peers PE-1 and PE-2. PE-1, PE-2, PE-3, PE-4, and PE-6 are capable of running both SR and LDP, but are initially configured to only run SR with the associated node-SIDs shown in [Figure 233: Example topology](#). When explicitly described, LDP will be enabled in conjunction with SR on these routers to show the difference between the two approaches, and to show how SR can be used as a fast reroute backup for SR primary LSPs.

Figure 233: Example topology



26075

The LDP configuration at PE-1 toward PE-5 is shown in the following output. The configuration at PE-2 is similar with the only exception being IP addressing.

```
# on PE-1:
configure {
  router "Base" {
    ldp {
      interface-parameters {
        interface "int-PE-1-PE-5" {
          ipv4 {
            }
          }
        }
      }
    }
  }
}
```

PE-1, PE-2, PE-3, PE-4, and PE-6 run SR. The following output provides an example of the relevant SR configuration parameters at PE-1, with similar configurations on the remaining SR routers. For a description of these parameters, see chapter [Segment Routing with IS-IS Control Plane](#).

```
# on PE-1:
```

```
configure {
  router "Base" {
    mpls-labels {
      sr-labels {
        start 20000
        end 20099
      }
    }
    isis 0 {
      admin-state enable
      advertise-router-capability as
      area-address [49.0001]
      segment-routing {
        admin-state enable
        prefix-sid-range {
          start-label 20000
          max-index 99
        }
      }
      interface "int-PE-1-PE-2" {
        interface-type point-to-point
        level-capability 2
        level 2 {
          metric 100
        }
      }
      interface "int-PE-1-PE-3" {
        interface-type point-to-point
        level-capability 2
        level 2 {
          metric 100
        }
      }
      interface "int-PE-1-PE-5" {
        interface-type point-to-point
        level-capability 2
        level 2 {
          metric 100
        }
      }
      interface "int-PE-1-RR-7" {
        interface-type point-to-point
        level-capability 2
        level 2 {
          metric 100
        }
      }
      interface "system" {
        passive true
        level-capability 2
        ipv4-node-sid {
          label 20001
        }
      }
      level 2 {
        wide-metrics-only true
      }
    }
  }
}
```

SR Mapping Server

An SR Mapping Server (SR-MS) is an integral part of SR-LDP interoperability and has the responsibility for advertising prefixes to SID/label mappings on behalf of routers that do not support SR. When using IS-IS, a SID/label-binding TLV (TLV 149) containing a prefix-SID sub-TLV is used to advertise one or more SID index/labels and one or more prefixes. In the example topology, PE-4 is selected as the SR-MS and will advertise a prefix SID for the non-SR-capable router, PE-5.

The following output provides an example of the configuration required to implement SR-MS functionality. Under the **segment-routing** node, a **mapping-server** context is created that allows for origination of a SID/label-binding TLV and prefix-SID sub-TLV. The syntax begins with **node-sid-map** and is followed by an index. In SR OS, prefix-SIDs are always advertised with an index value (as opposed to an absolute label value), and the formula {start-label + SID index} is used to derive the label value. In this example, **index 5** is used and, therefore, the derived label value is {20000+5} 20005 for the PE-5 prefix 192.0.2.5/32.

An optional **prefix-sid-range** argument allows for advertisement of a contiguous range of prefixes and associated SIDs using the configured index/prefix as the beginning of the range. Non-contiguous ranges require multiple entries and are advertised as separate SID/label-binding TLVs. An additional optional **set-flags s** argument can also be used to set the S-flag, which controls the flooding scope. When set, the flooding scope is the entire IS-IS domain. When not set, the flooding scope is the IS-IS level into which the TLV was advertised.

```
# on PE-4:
configure {
  router "Base" {
    isis 0 {
      segment-routing {
        prefix-sid-range {
          start-label 20000
          max-index 99
        }
        mapping-server {
          admin-state enable
          node-sid-map 5 {
            ip-prefix 192.0.2.5/32
          }
        }
      }
    }
  }
}
```

The relevant part of the IS-IS LSP generated by PE-4, showing the SID/label-binding TLV, is shown in the following output:

```
[/]
A:admin@PE-4# show router isis database PE-4.00-00 detail

=====
Rtr Base ISIS Instance 0 Database (detail)
=====
---snip---
Displaying Level 2 database
-----
LSP ID      : PE-4.00-00                Level      : L2
Sequence    : 0x6                      Checksum   : 0x5102  Lifetime   : 1172
Version     : 1                        Pkt Type  : 20      Pkt Ver    : 1
Attributes  : L1L2                     Max Area  : 3        Alloc Len  : 1492
SYS ID     : 1920.0000.2004             SysID Len : 6        Used Len   : 258
```

```

---snip---
TLVs :
  Area Addresses:
    Area Address : (3) 49.0001
  Supp Protocols:
    Protocols    : IPv4
  IS-Hostname   : PE-4
  Router ID    :
    Router ID   : 192.0.2.4
  Router Cap   : 192.0.2.4, D:0, S:0
    TE Node Cap : B E M P
    SR Cap: IPv4 MPLS-IPv6
      SRGB Base:20000, Range:100
    SR Alg: metric based SPF
    Node MSD Cap: BMI : 12 ERLD : 15
  SID Label Binding:
    Prefix: 192.0.2.5/32 Range:1 Weight:0 bFlgs:v4 SID:5 Algo:0 pFlgs:N
---snip---
Level (2) LSP Count : 1
-----
Control Info      : D = Prefix Leaked Down
                  S = Sub-TLVs Present
Attribute Flags  : N = Node Flag
                  R = Re-advertisement Flag
                  X = External Prefix Flag
                  E = Entropy Label Capability (ELC) Flag
Adj-SID Flags    : v4/v6 = IPv4 or IPv6 Address-Family
                  B = Backup Flag
                  V = Adj-SID carries a value
                  L = value/index has local significance
                  S = Set of Adjacencies
                  P = Persistently allocated
Prefix-SID Flags : R = Re-advertisement Flag
                  N = Node-SID Flag
                  nP = no penultimate hop POP
                  E = Explicit-Null Flag
                  V = Prefix-SID carries a value
                  L = value/index has local significance
Lbl-Binding Flags: v4/v6 = IPv4 or IPv6 Address-Family
                  M = Mirror Context Flag
                  S = SID/Label Binding flooding
                  D = Prefix Leaked Down
                  A = Attached Flag
SABM-flags Flags: R = RSVP-TE
                  S = SR-TE
                  F = LFA
                  X = FLEX-ALGO
FAD-flags Flags:  M = Prefix Metric
=====

```

At other routers within the SR domain, the presence of the advertised prefix can be validated as shown in the following output taken at PE-1. The SRMS field is set to Y for prefix 192.0.2.5/32, indicating that the prefix was advertised by an SR-MS. (In the case of IS-IS, the prefix-SID is a sub-TLV of the SID/label-binding TLV and the "N" (node-SID) flag is set; therefore, it can be recognized as being advertised by a mapping server.) The Y is followed by an "(S)" flag, indicating that the SRMS prefix-SID is selected to be programmed. This indication is provided in case there are multiple advertisements for the same prefix and/or node-SID from different SR mapping servers that result in some kind of conflict or inconsistency. If there are multiple mapping servers advertising the same prefix-SID, the advertising router with the lowest system/router ID is preferred.

[/]

```
A:admin@PE-1# show router isis prefix-sids
=====
Rtr Base ISIS Instance 0 Prefix/SID Table
=====
Prefix                               SID      Lvl/Typ  SRMS   AdvRtr
MT                                     Flags
-----
192.0.2.1/32                          1        2/Int.   N      PE-1
0                                     NnP
192.0.2.2/32                          2        2/Int.   N      PE-2
0                                     NnP
192.0.2.3/32                          3        2/Int.   N      PE-3
0                                     NnP
192.0.2.4/32                          4        2/Int.   N      PE-4
0                                     NnP
192.0.2.5/32                          5        2/Int.   Y(S)   PE-4
0                                     NnP
192.0.2.6/32                          6        2/Int.   N      PE-6
0                                     NnP
-----
No. of Prefix/SIDs: 6 (6 unique)
-----
SRMS : Y/N = prefix SID advertised by SR Mapping Server (Y) or not (N)
      S     = SRMS prefix SID is selected to be programmed
Flags: R    = Re-advertisement
      N     = Node-SID
      nP    = no penultimate hop POP
      E     = Explicit-Null
      V     = Prefix-SID carries a value
      L     = value/index has local significance
=====
```

SR-LDP interworking

Interworking SR and LDP essentially consists of stitching an LDP FEC and an SR node-SID route for the same prefix. In the example topology, PE-1 and PE-2 will act as the SR-LDP interworking nodes.

In the LDP-to-SR data plane direction, LDP uses an **export-tunnel-table** command under the **ldp** context to reference a policy that defines which prefixes should be redistributed from the IS-IS/SR domain into LDP. When applied, the LDP process monitors the tunnel-table until it locates a /32 SR tunnel of type sr-isis that matches a prefix defined in the export policy. LDP then programs an LDP Incoming Label Map (ILM) entry and stitches it to the SR node-SID tunnel endpoint. The LDP process also originates a FEC for the prefix and advertises that FEC to its peers.

The following output provides an example of the route policy and application of the policy at PE-1 and PE-2. In this policy, PE-1 advertises LDP FECs to PE-5 for PE-3 (192.0.2.3), PE-4 (192.0.2.4), and PE-6 (192.0.2.6), provided that PE-1 has a /32 SR tunnel of type sr-isis in the tunnel-table for those same prefixes. PE-1 also programs an LDP ILM entry for each prefix and stitches it to the appropriate SR tunnel.

```
# on PE-1 and PE-2:
configure {
  policy-options {
    prefix-list "sr-domain-prefixes" {
      prefix 192.0.2.3/32 type exact {
      }
      prefix 192.0.2.4/32 type exact {
      }
      prefix 192.0.2.6/32 type exact {
```

```

    }
  }
  policy-statement "SR-to-LDP-policy" {
    entry 10 {
      from {
        prefix-list ["sr-domain-prefixes"]
        protocol {
          name [isis]
        }
      }
      to {
        protocol {
          name [ldp]
        }
      }
      action {
        action-type accept
      }
    }
  }
}
router "Base" {
  ldp {
    export-tunnel-table ["SR-to-LDP-policy"]
  }
}
}

```

In the SR-to-LDP data plane direction, the **export-tunnel-table ldp** command within the **segment-routing** context is the only required configuration. Unlike the LDP-to-SR data plane direction, where policy is used to control which prefixes are stitched, in the SR-to-LDP direction, no policy is explicitly referenced because the SR-MS provides a network-wide policy for the prefixes that SR needs to stitch to a corresponding LDP FEC. With the **export-tunnel-table ldp** command applied, whenever a /32 LDP tunnel destination matches a prefix for which a prefix-SID sub-TLV was received from a mapping server, the SR ILM is stitched to the corresponding LDP tunnel endpoint.

The following output shows the configuration applied at PE-1 to implement SR-to-LDP data plane interworking:

```

# on PE-1:
configure {
  router "Base" {
    isis 0 {
      segment-routing {
        admin-state enable
        export-tunnel-table ldp
      }
    }
  }
}
}

```

With the required configuration in the SR-LDP interworking routers (PE-1 and PE-2), it is possible to validate the correct ILM entries. In the LDP-to-SR data plane direction, the following output shows the active LDP bindings at PE-1. Each of the entries for PE-3 (192.0.2.3), PE-4 (192.0.2.4), and PE-6 (192.0.2.6) have an "(I)" flag to indicate that the prefix has an SR-ISIS next-hop. Each entry also has an ingress label and an egress label. The ingress label represents the LDP FEC advertised for the corresponding prefix (in this case, advertised only to PE-5). The egress label represents the SR node-SID for the same prefix. Therefore, a mapping exists between LDP FEC and SR node-SID.

[/]

```
A:admin@PE-1# show router ldp bindings active prefixes ipv4

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static          (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop    (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.1/32                          Pop
524287                                 --
--                                     --

192.0.2.3/32(I)                       Swap
524280                                 20003
192.168.13.2                          1/1/1:100

192.0.2.4/32(I)                       Swap
524279                                 20004
192.168.12.2                          1/1/3:100

192.0.2.5/32                          Push
--                                     524287
192.168.15.2                          1/1/2:100

192.0.2.6/32(I)                       Swap
524278                                 20006
192.168.13.2                          1/1/1:100

-----
No. of IPv4 Prefix Active Bindings: 5
=====
```

In the SR-to-LDP data plane direction, the following output, taken at PE-1, shows a dump of the SR database for next-hops resolved to LDP. There is a single entry with index 5 (label value 20005) advertised by the SR-MS for the PE-5 prefix 192.0.2.5. The final line of the entry shows that an LDP FEC is the SID next-hop for SR-LDP stitching. The tunnel LSP ID is 65537. The tunnel-table verifies that this is an LDP tunnel to PE-5 (192.0.2.5).

```
[/]
A:admin@PE-1# tools dump router isis sr-database nh-type ldp detail
=====
Rtr Base ISIS Instance 0 SR Database

Legend:
label stack is ordered from bottom-most to top-most
=====
```



```

-----
SID 5
-----
Label           : 20005           Adv System Id      : 1920.0000.2005
Prefix          : 192.0.2.5
Route Level     : 2              MT Id              : 0
Rtm Preference  : 18            Ttm Preference     : 0
Metric          : 0              Last Action        : AddTnl
Num Ip NextHop  : 0              Num SR-Tnl NextHop : 1
Mtu             : 0
Mtu Prim        : 0              Mtu Backup         : -
Exclude from LFA : 0            LFA Type           : -
Duplicate Pending : 0           Tunnel Active State : Reported/Ack
SR Error        : SR_ERR_OK
-----

NHOP: IP                IsTunl  GIfId/  IfId/ PgId  IsAdv Label  IsLfaX
                        TunlType LspId
-----
192.0.2.5                Y       2       65537 0    0    0    0
-----

No. of Entries: 1
-----
LDP = LDP FEC is the SID NH for SR-LDP stitching
=====

```

To verify that the data plane is intact from end-to-end, a VPRN service is configured at the non-SR-capable PE-5 and the SR-capable PE-6, each with a locally configured subnet that is used to test IP connectivity. The configuration of the VPRN at PE-5 is shown in the following output. The **auto-bind-tunnel** configuration uses a resolution filter allowing only **ldp** to be used to resolve BGP next-hops for VPN-IPv4 routes. Usually, this could be configured for **resolution any**, but this configuration shows that LDP is being used. The local IP address at PE-5 is 172.31.5.1/24.

```

# on PE-5:
configure {
  service {
    vprn "VPRN1-name" {
      admin-state enable
      service-id 1
      customer "1"
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "64496:1"
          vrf-target {
            community "target:64496:1"
          }
          auto-bind-tunnel {
            resolution filter
            resolution-filter {
              ldp true
            }
          }
        }
      }
    }
  }
  interface "Local-Subnet" {
    ipv4 {
      primary {
        address 172.31.5.1
        prefix-length 24
      }
    }
  }
}

```

```

    }
    sap 1/2/1:1 {
    }
  }
}

```

The configuration of the VPRN at PE-6 is shown in the following output. Again, the **auto-bind-tunnel** configuration uses a resolution filter, but this time it is configured for **sr-isis**. It could be set to **resolution any**, so that the tunnel-table preference would resolve an LSP with the lowest preference/metric, but the resolution filter configuration again shows that SR is being used. The **auto-bind-tunnel** context allows the transport mechanism to be a local decision at service level. The local IP address at PE-6 is 172.31.6.1/24.



Note:

An alternative approach would be to configure the auto-bind-tunnel context for **resolution any**, then modify the tunnel-table preference for SR using the **tunnel-table-pref** command in the **segment-routing** context.

```

# on PE-6
configure {
  service {
    vprn "VPRN1-name" {
      admin-state enable
      service-id 1
      customer "1"
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "64496:1"
          vrf-target {
            community "target:64496:1"
          }
          auto-bind-tunnel {
            resolution filter
            resolution-filter {
              sr-isis true
            }
          }
        }
      }
    }
  }
  interface "Local-Subnet" {
    ipv4 {
      primary {
        address 172.31.6.1
        prefix-length 24
      }
    }
    sap 1/2/1:1 {
    }
  }
}
}
}

```

A VPRN ping between 172.31.5.1 at PE-5 and 172.31.6.1 at PE-6 verifies that the data plane is intact:

```

[/]
A:admin@PE-5# ping 172.31.6.1 source-address 172.31.5.1 router-instance "VPRN1-name"
PING 172.31.6.1 56 data bytes

```

```
64 bytes from 172.31.6.1: icmp_seq=1 ttl=64 time=3.27ms.
64 bytes from 172.31.6.1: icmp_seq=2 ttl=64 time=3.31ms.
. 64 bytes from 172.31.6.1: icmp_seq=3 ttl=64 time=3.41ms.
. 64 bytes from 172.31.6.1: icmp_seq=4 ttl=64 time=3.40ms.
. 64 bytes from 172.31.6.1: icmp_seq=5 ttl=64 time=2.70ms.

---- 172.31.6.1 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 2.70ms, avg = 3.22ms, max = 3.41ms, stddev = 0.266ms
```

SR and LDP coexistence

The previous example demonstrates the use of SR-LDP interworking when the SR domain runs only SR. A more common scenario is that SR will coexist with LDP, because LDP is already deployed and the SR deployment will be added. In this sub-section, PE-1, PE-2, PE-3, PE-4, and PE-6 are configured to run LDP in conjunction with SR. PE-5 remains the same as the previous sub-section in that it runs only LDP to its connected peers PE-1 and PE-2.

In the SR-to-LDP data plane direction, there is no notable change when LDP coexists in the SR domain. Whenever a /32 LDP tunnel destination matches a prefix for which a prefix-SID sub-TLV was received from a mapping server, the SR ILM is stitched to the corresponding LDP tunnel endpoint.

In the LDP-to-SR data plane direction, there is a significant change. If only SR is running within the SR domain, the LDP process monitors the tunnel-table and when a /32 SR tunnel of type sr-isis is found that matches a prefix in the (**export-tunnel-table**) export policy, LDP programs an LDP ILM and stitches it to the SR node-SID tunnel endpoint. However, if an LDP FEC exists for the same /32 prefix, SR OS will resolve the LDP ILM entry to the LDP FEC. This is because LDP attempts to resolve the prefix in the route table first before looking in the tunnel-table and, therefore, prefers the LDP tunnel to the SR tunnel.

The following output is taken at PE-1 when LDP and SR coexist in the SR domain. The previous version of this output (when LDP was not running in the SR domain) showed the prefixes for PE-3, PE-4, and PE-6 as known via an SR-ISIS next-hop, and the egress labels as node-SIDs. When LDP is active in conjunction with SR, the egress labels resolve to an LDP FEC.

```
[/]
A:admin@PE-1# show router ldp bindings active prefixes ipv4

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                                EgrLbl
EgrNextHop                            EgrIf/LspId
```

```

-----
192.0.2.1/32                Pop
524287                    --
--                        --

192.0.2.2/32                Push
--                        524287
192.168.12.2                1/1/3:100

192.0.2.2/32                Swap
524282                    524287
192.168.12.2                1/1/3:100

192.0.2.3/32                Push
--                        524283
192.168.13.2                1/1/1:100

192.0.2.3/32                Swap
524280                    524283
192.168.13.2                1/1/1:100

192.0.2.4/32                Push
--                        524279
192.168.12.2                1/1/3:100

192.0.2.4/32                Swap
524279                    524279
192.168.12.2                1/1/3:100

192.0.2.5/32                Push
--                        524287
192.168.15.2                1/1/2:100

192.0.2.5/32                Swap
524283                    524287
192.168.15.2                1/1/2:100

192.0.2.6/32                Push
--                        524278
192.168.13.2                1/1/1:100

192.0.2.6/32                Swap
524278                    524278
192.168.13.2                1/1/1:100

-----
No. of IPv4 Prefix Active Bindings: 11
=====

```

That the LDP-to-SR data path resolves to LDP FECs rather than SR tunnels may result in an asymmetric data path. Taking the previously used VPRN service between PE-5 and PE-6 as an example:

- Traffic from PE-6 to PE-5 will use SR between PE-6 and one of the SR-LDP interworking gateways at PE-1 or PE-2, after which it will use LDP.
- Traffic from PE-5 to PE-6 will use LDP between ingress and egress. The interworking function between SR and LDP has no effect.

Both directions still use an MPLS data plane. However, the MPLS control plane differs in each direction.

LDP fast reroute using SR tunnels

With the ability to interwork LDP and SR, primary LSPs signaled using LDP can select a remote LFA SR tunnel as backup. This provides the potential to increase fast reroute coverage. As with any other backup or fast reroute mechanism, the SR backup tunnel can be installed in the forwarding database before any failure, but can only be activated when the failure of the primary path has been detected.

The ability to detect a failure quickly forms a significant part of the overall reconvergence time and may require the use of failure detection mechanisms, such as Bidirectional Forwarding Detection (BFD), the 802.3ah Ethernet in the First Mile (EFM), or just Loss of Signal (LoS). These mechanisms are beyond the scope of this chapter.

To use SR as a backup for LDP, the **fast-reroute backup-sr-tunnel** command must be configured in the **ldp** context. The **export-tunnel-table** command previously described should also be present, and should reference a policy including all of the prefixes for which backup is required. There is no requirement for an SR-MS when using SR tunnels for LDP backup, nor is there a requirement to enable SR-to-LDP interworking using the **export-tunnel-table ldp** command within the **segment-routing** context.

The following output shows the configuration applied at PE-6. When this configuration is applied, if the LFA SPF does not find an adjacent IP next-hop prefix for an LDP FEC, but can compute a remote LFA tunnel next-hop, LDP programs the LDP FEC using an LDP Next-Hop Label Forwarding Entry (NHLFE), and a backup next-hop using an LDP NHLFE pointing to the SR tunnel endpoint. The LDP packet is not tunneled over the SR tunnel, but rather the LDP label is stitched to the segment-routing label stack. This behavior is similar to the LDP-SR interworking function previously described within this chapter, but is modified such that the stitching of an LDP ILM entry to an SR tunnel only takes place if no adjacent LFA next-hop could be found for the prefix.

```
# on PE-6:
configure {
  policy-options {
    prefix-list "sr-domain-prefixes" {
      prefix 192.0.2.0/24 type longer {
      }
    }
  }
  policy-statement "SR-to-LDP-policy" {
    entry 10 {
      from {
        prefix-list ["sr-domain-prefixes"]
        protocol {
          name [isis]
        }
      }
      to {
        protocol {
          name [ldp]
        }
      }
      action {
        action-type accept
      }
    }
  }
}
router "Base" {
  isis 0 {
    loopfree-alternate {
      remote-lfa {
      }
    }
  }
}
```

```

    ldp {
      export-tunnel-table ["SR-to-LDP-policy"]
      fast-reroute {
        backup-sr-tunnel true
      }
    }
  }
}

```

With the preceding configuration in place at PE-6, it is possible to verify whether a backup exists for a specific prefix, using the command shown in the following output. In this example, the backup is displayed for the PE-6 adjacent neighbor PE-3 (192.0.2.3). There are two LSPs for the prefix 192.0.2.3/32; one is known via LDP and one is known via SR-ISIS, indicated in the protocol column. The entries are defined as follows:

- The first line of the LDP entry is the primary LSP with a next-hop of 192.168.36.1 using interface 1/1/2:100 (direct to PE-3).
- The second line of the LDP entry is the backup indicated by a "(B)" flag, with a next-hop of 192.168.46.1 using interface 1/1/1:100 (via PE-4). This backup is a basic LFA, which is possible to compute due to the example topology, or more explicitly the triangular mesh between PE-6, PE-4, and PE-3. Due to this topology, if the link between PE6 and PE3 fails, PE-6 can forward packets destined for PE-3 toward PE-4. PE-4 will then forward them directly toward PE-3, not return them to PE-6 (which would create a transient micro-loop until the next SPF is run).
- The first line of the SR-ISIS entry is the primary LSP with a next-hop of 192.168.36.1 using interface 1/1/2:100 (direct to PE-3).
- The second line of the SR-ISIS entry is the backup LSP indicated by the "(B)" flag, with a next-hop of 192.168.46.1 using interface 1/1/1:100 (via PE-4). Both the primary and backup LSPs use the label 20003, representing the PE-3 node-SID. As with the LDP backup entry, the SR-ISIS backup is a basic LFA.

```

[/]
A:admin@PE-6# show router fp-tunnel-table 1 192.0.2.3/32

=====
IPv4 Tunnel Table Display

Legend:
Label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol      Tunnel-ID
  Lbl
  NextHop
  Lbl      (backup)                        Intf/Tunnel
  NextHop  (backup)
-----
192.0.2.3/32                                LDP           -
524283
  192.168.36.1                               1/1/2:100
524281
  192.168.46.1(B)                             1/1/1:100
192.0.2.3/32                                SR-ISIS-0     524291
20003
  192.168.36.1                               1/1/2:100
20003
  192.168.46.1(B)                             1/1/1:100
-----
Total Entries : 2

```

To show the benefits that SR provides in increasing fast reroute coverage, the link between PE-4 and PE-3 is removed from the example topology, creating a ring topology. With this link removed, it is no longer possible for PE-6 to compute a basic LFA to PE-3 for the link between PE-6 and PE-3. If that link failed and PE-6 forwarded packets destined for PE-3 toward PE-4, PE-4 would return them to PE-6 until the next SPF was complete. Therefore, a backup tunnel is needed to a place in the network that will not loop packets back; essentially a remote LFA.

The following output at PE-6 shows the primary and backup LSPs for PE-3 (192.0.2.3) with the modified topology. Again, there are two LSPs: one known through via LDP and one known via SR-ISIS. The entries are defined as follows:

- The first line of the LDP entry is the primary LSP with a next-hop of 192.168.36.1 using interface 1/1/2:100 (direct to PE-3).
- The second line of the LDP entry is the backup indicated by a "(B)" flag with a next-hop of 192.0.2.3 (PE-3), which uses an SR tunnel. The label of "3" (implicit-null) indicates that the LDP label is not tunneled through the SR tunnel, but rather popped before the primary LDP LSP is stitched to the backup SR LSP.
- The first line of the SR-ISIS entry is the primary LSP with a next-hop of 192.168.36.1 using interface 1/1/2:100 (direct to PE-3). This LSP assigns a single label of value 20003, representing the node-SID of PE-3.
- The second line of the SR-ISIS entry is the backup indicated by a "(B)" flag with a next-hop of 192.168.46.1 using interface 1/1/1:100 (via PE-4). There are two labels assigned to this backup tunnel. The upper label has a value of 20002, which represents the node-SID of PE-2. This is the remote LFA "PQ-node". The second label has a value of 20003, which represents the node-SID of the destination, PE-3.

When this backup tunnel is operational, PE-6 encapsulates traffic destined for PE-3 to a point in the network where it will not be looped back toward the source. In the example topology, that node is PE-2. When traffic arrives at PE-2, it pops the top label (20002) and forwards traffic for PE-3 (with label 20003) on the shortest path toward the destination.

```
[/]
A:admin@PE-6# show router fp-tunnel-table 1 192.0.2.3/32

=====
IPv4 Tunnel Table Display

Legend:
Label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol      Tunnel-ID
  Lbl
  NextHop
  Lbl      (backup)                        Intf/Tunnel
  NextHop  (backup)
-----
192.0.2.3/32                                LDP           -
 524283
 192.168.36.1                               1/1/2:100
 3
 192.0.2.3(B)                               SR
192.0.2.3/32                                SR-ISIS-0     524291
20003
```

```
192.168.36.1 1/1/2:100
20003/20002
192.168.46.1(B) 1/1/1:100
-----
Total Entries : 2
-----
=====
```

Conclusion

The SR control plane can (and likely will) coexist with other MPLS control plane clients, such as RSVP, LDP, or BGP. It is possible that these control plane clients will operate independently. However, where a mix of SR-capable and non-SR-capable routers exist within the same domain, SR-LDP interworking is necessary to form an end-to-end LSP. This chapter shows how that is possible using one or more SR mapping servers and one or more interworking routers.

SR-LDP interworking also provides an opportunity to increase fast reroute coverage in LDP-based networks. Before the introduction of SR-LDP interworking, a remote LFA could only be constructed using LDP-over-RSVP, which required the RSVP LSP to be manually configured and placed. When SR-LDP interworking is used, primary LDP LSPs can use a backup tunnel to a remote LFA signaled using SR. This requires no manual configuration, which provides the potential to greatly increase fast reroute coverage with minimal effort.

MPLS LDP FRR using ISIS as IGP

This chapter describes Multi- Protocol Label Switching (MPLS) Label Distribution Protocol (LDP) Fast Reroute (FRR) using Intermediate System to Intermediate System (IS-IS) as the Interior Gateway Protocol (IGP).

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 9.0.R6, but the configuration in the current edition corresponds to MD-CLI SR OS Release 21.2.R1. There are no prerequisites for this configuration.

Overview

LDP FRR improves convergence in case of a single link or single node failure in the network. Convergence times will be in the order of tens of milliseconds. This is important to some application services, such as voice over IP (VoIP), which are sensitive to traffic loss when running over the MPLS network.

Without FRR, link and/or node failures inside an MPLS LDP network result in traffic loss in the order of hundreds of milliseconds. The reason for that is that LDP depends on the convergence of the underlying IGP (IS-IS sending link state PDUs (LSPs) in this case). After IGP convergence, LDP itself needs to compute new primary Next Hop Label Forwarding Entries (NHLFEs) for all affected Forwarding Equivalence Classes (FECs). Finally, the different Label Forwarding Information Bases (LFIBs) are updated.

When FRR is configured on a node, the node computes primary NHLFEs for all FECs and, in addition, it will compute backup NHLFEs for all FECs. The backup NHLFE corresponds to the label received for the same FEC from a Loop-Free Alternate (LFA) next hop, see RFC 5286, *Basic Specification for IP Fast Reroute: Loop-Free Alternates*. Both primary NHLFEs and backup NHLFEs are programmed in the IOM/IMM, which makes it possible to converge very quickly.

The SR OS software has implemented Inequality 1 (link criterion) and Inequality 3 (node criterion) of RFC 5286. Similar to the Shortest Path Tree (SPT) computation that is part of standard link-state routing functionality, also the LFA next hop computation is based on the IGP metric.

The underlying LFA formulas appear in the following format:

Inequality 1:

$$SP(\text{backup NHR}, D) < \{SP(\text{backup NHR}, S) + SP(S, D)\}$$

Inequality 3:

$$SP(\text{backup NHR}, D) < \{SP(\text{backup NHR}, PN) + SP(PN, D)\}$$

In these inequalities 'SP' is 'shortest IGP metric path', 'NHR' is 'next hop router', 'D' is 'destination', 'S' is 'source node or upstream node doing the actual LFA next-hop computation', and 'PN' is 'protected node'. The Inequality 3 rule is stricter than the Inequality 1 rule. See [Additional topics](#) for a practical example on these inequalities.

Configuration

This section includes the following:

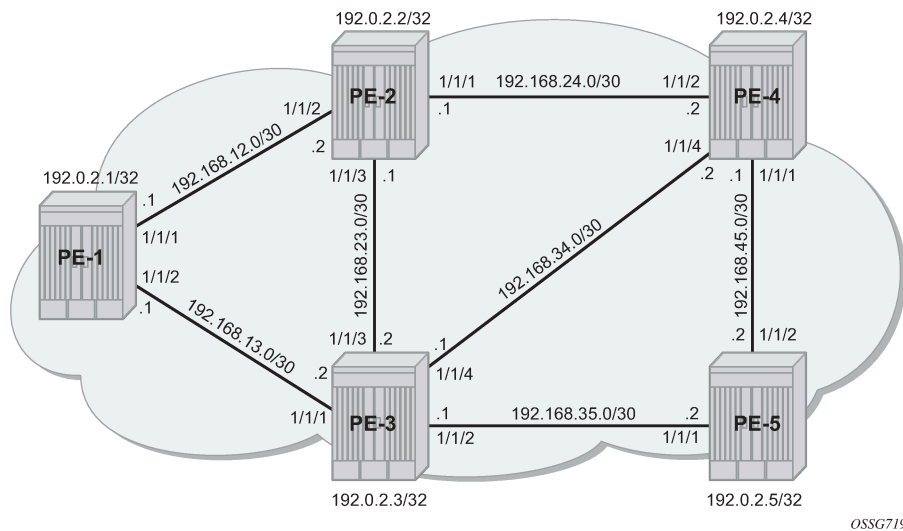
- [Configure the IP/MPLS network](#)
- [Enable LDP FRR and verify](#)
- [Enable synchronization timer](#)
- [Verify data path](#)

The subsection [Additional topics](#) includes:

- [Metric change](#)
- [IS-IS overload](#)

[Figure 234: Initial example topology](#) shows the example topology with five PEs in the same autonomous system.

Figure 234: Initial example topology



Configure the IP/MPLS network

The system addresses and IP interface addresses are configured according to [Figure 234: Initial example topology](#). An interior gateway protocol (IGP) is needed to distribute routing information on all PEs. In this

case, the IGP is IS-IS where each PE is acting as a level 2 router. On PE-1, the IS-IS configuration is as follows. The configuration is similar on the other PEs.

```
# on PE-1:
configure {
  router "Base" {
    isis 0 {
      admin-state enable
      level-capability 2
      area-address [49.0001]
      interface "int-PE-1-PE-2" {
        interface-type point-to-point
      }
      interface "int-PE-1-PE-3" {
        interface-type point-to-point
      }
      interface "system" {
      }
      level 2 {
        wide-metrics-only true
      }
    }
  }
}
```

IS-IS interfaces are set up as type point-to-point to improve convergence because no Designated Router/ Backup Designated Router (DR/BDR) election process is done. The **show router isis adjacency** command on PE-1 verifies that the IS-IS adjacencies are up:

```
[/]
A:admin@PE-1# show router isis adjacency

=====
Rtr Base ISIS Instance 0 Adjacency
=====
System ID                Usage State Hold Interface          MT-ID
-----
PE-2                      L2   Up   27   int-PE-1-PE-2                0
PE-3                      L2   Up   20   int-PE-1-PE-3                0
-----
Adjacencies : 2
=====
```

The **show router route-table** command on PE-1 verifies which IP interface addresses or subnets are known on the PE:

```
[/]
A:admin@PE-1# show router route-table

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]      Type  Proto  Age           Pref
Next Hop[Interface Name] Metric
-----
192.0.2.1/32            Local  Local  00h03m34s    0
  system
192.0.2.2/32            Remote  ISIS   00h02m59s    18
  192.168.12.2          10
192.0.2.3/32            Remote  ISIS   00h02m11s    18
  192.168.13.2          10
192.0.2.4/32            Remote  ISIS   00h01m34s    18
  192.168.12.2          20
```

```

192.0.2.5/32                               Remote  ISIS    00h01m00s 18
      192.168.13.2                          20
192.168.12.0/30                            Local   Local   00h03m34s  0
      int-PE-1-PE-2                          0
192.168.13.0/30                            Local   Local   00h03m34s  0
      int-PE-1-PE-3                          0
192.168.23.0/30                            Remote  ISIS    00h02m59s 18
      192.168.12.2                          20
192.168.24.0/30                            Remote  ISIS    00h02m59s 18
      192.168.12.2                          20
192.168.34.0/30                            Remote  ISIS    00h02m11s 18
      192.168.13.2                          20
192.168.35.0/30                            Remote  ISIS    00h02m11s 18
      192.168.13.2                          20
192.168.45.0/30                            Remote  ISIS    00h01m34s 18
      192.168.12.2                          30
-----
No. of Routes: 12
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

The **show router fib 1** command on PE-1 shows the content of the forwarding information base (FIB):

```

[/]
A:admin@PE-1# show router fib 1

=====
FIB Display
=====
Prefix [Flags]                               Protocol
NextHop
-----
192.0.2.1/32                                 LOCAL
      192.0.2.1 (system)
192.0.2.2/32                                 ISIS
      192.168.12.2 (int-PE-1-PE-2)
192.0.2.3/32                                 ISIS
      192.168.13.2 (int-PE-1-PE-3)
192.0.2.4/32                                 ISIS
      192.168.12.2 (int-PE-1-PE-2)
192.0.2.5/32                                 ISIS
      192.168.13.2 (int-PE-1-PE-3)
192.168.12.0/30                              LOCAL
      192.168.12.0 (int-PE-1-PE-2)
192.168.13.0/30                              LOCAL
      192.168.13.0 (int-PE-1-PE-3)
192.168.23.0/30                              ISIS
      192.168.12.2 (int-PE-1-PE-2)
192.168.24.0/30                              ISIS
      192.168.12.2 (int-PE-1-PE-2)
192.168.34.0/30                              ISIS
      192.168.13.2 (int-PE-1-PE-3)
192.168.35.0/30                              ISIS
      192.168.13.2 (int-PE-1-PE-3)
192.168.45.0/30                              ISIS
      192.168.12.2 (int-PE-1-PE-2)
-----
Total Entries : 12
=====

```

Initially, the following default IS-IS Level 2 metric applies to all interfaces.

```
[/]
A:admin@PE-1# show router isis status | match "L2 Default Metric"
L2 Default Metric      : 10
```

The next step in the process of setting up the IP/MPLS network is setting up interface-LDP sessions on all interfaces. The LDP configuration on PE-1 is as follows:

```
# on PE-1:
configure {
  router "Base" {
    ldp {
      interface-parameters {
        interface "int-PE-1-PE-2" {
          ipv4 {
            admin-state enable
          }
        }
        interface "int-PE-1-PE-3" {
          ipv4 {
            admin-state enable
          }
        }
      }
    }
  }
}
```

There is now a full mesh of LDP label switched paths (LSPs) set up between all system interfaces of the PEs, and the tunnel table on PE-1 looks as follows:

```
[/]
A:admin@PE-1# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.2/32         ldp       MPLS  65537    9    192.168.12.2  10
192.0.2.3/32         ldp       MPLS  65538    9    192.168.13.2  10
192.0.2.4/32         ldp       MPLS  65539    9    192.168.12.2  20
192.0.2.5/32         ldp       MPLS  65540    9    192.168.13.2  20
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====
```

The LDP LSP metric follows the IGP cost. Optionally, LSP metrics can be applied but that is beyond the scope for this chapter.

Enable LDP FRR and verify

Because LDP FRR is using LFA next-hop computation by the IGP, as described in RFC 5286, LFA must be enabled in the IGP context, as follows:

```
# on PE-1:
configure {
  router "Base" {
    isis 0 {
      loopfree-alternate {
    }
  }
}
```

The **show router isis status** command on PE-1 verifies that LFA is enabled in IS-IS:

```
[/]
A:admin@PE-1# show router isis status | match "Loopfree"
Loopfree-Alternate      : Enabled
```

After enabling LFA inside the IGP context, FRR needs to be enabled within the LDP context, as follows:

```
# on PE-1:
configure {
  router "Base" {
    ldp {
      fast-reroute {
    }
  }
}
```

The **show router ldp status** command on PE-1 verifies that FRR is enabled in LDP:

```
[/]
A:admin@PE-1# show router ldp status | match FRR
FRR      : Enabled          Mcast Upstream FRR  : Disabled
Mcast Upst ASBR FRR: Disabled
```

This chapter describes FRR for unicast LDP. For multicast upstream FRR, see the "Multicast Label Distribution Protocol" chapter in the MPLS volume of the *7450 ESS, 7750 SR, and 7950 XRS Advanced Configuration Guide — Part I*. After these two CLI commands, the software computes for each LDP FEC in the network both a primary and a backup NHLFE and uploads it to the IOM/IMM. The primary NHLFE corresponds to the label of the FEC received from the primary next-hop as per standard LDP resolution of the FEC prefix in the Routing Table Manager (RTM). The backup NHLFE corresponds to the label received for the same FEC from an LFA next hop.

For point-to-point interfaces, when multiple LFA next hops are found for a primary next hop, the following selection criteria are used:

- It will pick the node-protect type in favor of the link-protect type.
- If there is more than one LFA next hop within the selected type, then it will pick one based on the lowest cost.
- If more than one LFA next hop with the same cost, SPF will select the first one. This is not a deterministic selection and will vary following each SPF calculation.

Several show commands are possible to display LFA information:

The **show router isis statistics** command shows the number of LFA runs on a specific node.

```
[/]
A:admin@PE-1# show router isis statistics
```

```

=====
Rtr Base ISIS Instance 0 Statistics
=====
---snip---

LFA Statistics
LFA Runs      : 1
Last scheduled : 03/11/2021 08:49:32
Partial LFA Runs : 0

---snip---

```

Remote LFA (RLFA) statistics and Topology-independent LFA (TI-LFA) statistics have been removed from the preceding output, because they are beyond the scope of this chapter. RLFA and TI-LFA are used in segment routing and described in the "Segment Routing with IS-IS Control Plane" chapter in the MPLS volume of the *7450 ESS, 7750 SR, and 7950 XRS Advanced Configuration Guide — Part I* and [Topology-Independent Loop-Free Alternate for Link Protection](#).

The **show router isis lfa-coverage** command performs a mathematical calculation between the number of nodes and IPv4/IPv6 routes in the network versus present LFA next-hop protections. In the example topology (see [Figure 234: Initial example topology](#)), all IS-IS links have a default level 2 metric of 10. This results in all four nodes and all IS-IS routes learned by PE-1 being 100% LFA protected (link or node), as follows:

```

[/]
A:admin@PE-1# show router isis lfa-coverage

=====
Rtr Base ISIS Instance 0 LFA Coverage
=====
Topology          Level  Node      IPv4      IPv6
-----
IPV4 Unicast     L1     0/0(0%)  9/9(100%) 0/0(0%)
IPV6 Unicast     L1     0/0(0%)  0/0(0%)  0/0(0%)
IPV4 Multicast   L1     0/0(0%)  0/0(0%)  0/0(0%)
IPV6 Multicast   L1     0/0(0%)  0/0(0%)  0/0(0%)
IPV4 Unicast     L2     4/4(100%) 9/9(100%) 0/0(0%)
IPV6 Unicast     L2     0/0(0%)  0/0(0%)  0/0(0%)
IPV4 Multicast   L2     0/0(0%)  0/0(0%)  0/0(0%)
IPV6 Multicast   L2     0/0(0%)  0/0(0%)  0/0(0%)
=====

```

The **show router isis topology lfa detail** command shows the LFA protection type (link or node), as follows:

```

[/]
A:admin@PE-1# show router isis topology lfa detail

=====
Rtr Base ISIS Instance 0 Topology Table
=====
IS-IS IP paths (MT-ID 0), Level 2
-----
Node      : PE-2.00
Nexthop   : PE-2
Interface : int-PE-1-PE-2
SNPA      : none                               Metric    : 10
LFA nh    : PE-3

```

```

LFA intf  : int-PE-1-PE-3          LFA Metric  : 20
LFA type  : linkProtection

Node      : PE-3.00
Nexthop   : PE-3
Interface : int-PE-1-PE-3          Metric      : 10
SNPA     : none

LFA nh    : PE-2
LFA intf  : int-PE-1-PE-2          LFA Metric  : 20
LFA type  : linkProtection

Node      : PE-4.00
Nexthop   : PE-2
Interface : int-PE-1-PE-2          Metric      : 20
SNPA     : none

LFA nh    : PE-3
LFA intf  : int-PE-1-PE-3          LFA Metric  : 20
LFA type  : nodeProtection

Node      : PE-5.00
Nexthop   : PE-3
Interface : int-PE-1-PE-3          Metric      : 20
SNPA     : none

LFA nh    : PE-2
LFA intf  : int-PE-1-PE-2          LFA Metric  : 30
LFA type  : linkProtection

```

The **show router route-table** command adds an 'L' flag as reference that the associated prefix is having also an LFA next hop available.

```

[/]
A:admin@PE-1# show router route-table 192.0.2.4

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age           Pref
  Next Hop[Interface Name]                               Metric
-----
192.0.2.4/32 [L]           Remote ISIS  00h18m47s  18
  192.168.12.2                               20
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====

```

The **show router route-table alternative** command or **show router isis routes alternative** command show detailed interface address information used by the LFA calculation:

```

[/]
A:admin@PE-1# show router route-table 192.0.2.4 alternative

=====
Route Table (Router: Base)
=====

```



```

=====
Dest Prefix[Flags]                               Type  Proto  Age      Pref
  Next Hop[Interface Name]                       Metric
  Alt-NextHop                                     Alt-
                                                Metric
-----
192.0.2.4/32                                     Remote ISIS  00h18m47s 18
  192.168.12.2                                   20
  192.168.13.2 (LFA)                             20
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
Backup = BGP backup route
LFA = Loop-Free Alternate nexthop
S = Sticky ECMP requested
=====

```

```

[/]
A:admin@PE-1# show router isis routes alternative ip-prefix-prefix-length 192.0.2.4

```

```

=====
Rtr Base ISIS Instance 0 Route Table (alternative)
=====
Prefix[Flags]                               Metric  Lvl/Typ  Ver.  SysID/Hostname
  NextHop                                     MT      AdminTag/SID[F]
  Alt-Nexthop                                Alt-    Alt-Type
                                                Metric
-----
192.0.2.4/32                                  20      2/Int.   5     PE-2
  192.168.12.2                                0       0
  192.168.13.2(L)                             20      NP
-----
No. of Routes: 1 (1 path)
-----
Flags      : L = Loop-Free Alternate nexthop
Alt-Type   : LP = linkProtection, NP = nodeProtection
SID[F]      : R = Re-advertisement
              N = Node-SID
              nP = no penultimate hop POP
              E = Explicit-Null
              V = Prefix-SID carries a value
              L = value/index has local significance
=====

```

On PE-1, PE-4 (192.0.2.4/32) has a primary SPF next-hop pointing toward PE-2 (192.168.12.2) and an LFA next-hop pointing toward PE-3 (192.168.13.2).

The Inequality 3 formula on PE-1 for prefix 192.0.2.4/32 results in the following:

Inequality 3:

- $[SP(\text{backup NHR}, D) < \{SP(\text{backup NHR}, PN) + SP(PN, D)\}]$ or
- $[SP(\text{PE-3}, \text{PE-4}) < \{SP(\text{PE-3}, \text{PE-2}) + SP(\text{PE-2}, \text{PE-4})\}]$ or
- $[10 < \{10 + 10\}]$

This means that Inequality 3 is met. The calculated LFA next-hop for prefix 192.0.2.4/32 on PE-1 is protecting node PE-2, see [Figure 234: Initial example topology](#) for a graphical representation.

The **show router ldp bindings** command displays the Label Information Base (LIB). A BU flag is present in case the associated label is used as backup NHLFE for the prefix. As an example, a display on PE-1 for prefix PE-4 is as follows.

This is only possible because the SR OS LDP implementation is using liberal retention mode which means that every label mapping received by a peer is retained regardless of whether the LSR is the next hop for the advertised mapping.

```
[/]
A:admin@PE-1# show router ldp bindings prefixes prefix 192.0.2.4/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
              (IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP IPv4 Prefix Bindings
=====
Prefix
Peer                               FEC-Flags
IgrLbl                             EgrLbl
EgrNextHop                         EgrIntf/LspId
-----
192.0.2.4/32
192.0.2.2:0
524284N                             524284
192.168.12.2                       1/1/1

192.0.2.4/32
192.0.2.3:0
524284U                             524284BU
192.168.13.2                       1/1/2

-----
No. of IPv4 Prefix Bindings: 2
=====
```

The **show router ldp bindings active** command displays the label forwarding information base (LFIB). Also, the BU flag is present and, in addition, a reference to the label action itself: **pop** for eLER, **push** for iLER and **swap** for LSR.

```
[/]
A:admin@PE-1# show router ldp bindings active prefixes prefix 192.0.2.4/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
              (IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static                (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop          (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop     (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
```

```

=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.4/32                         Push
--                                   524284
192.168.12.2                         1/1/1

192.0.2.4/32                         Push
--                                   524284BU
192.168.13.2                         1/1/2

192.0.2.4/32                         Swap
524284                               524284
192.168.12.2                         1/1/1

192.0.2.4/32                         Swap
524284                               524284BU
192.168.13.2                         1/1/2

-----
No. of IPv4 Prefix Active Bindings: 4
=====

```

Enable synchronization timer

Within an MPLS network using LDP, it is common practice to enable a synchronization timer between LDP and the IGP. Also, when LDP FRR is enabled, a situation can occur in which a synchronization timer between IGP and LDP will help: the revert scenario. When the interface for the previous primary next hop is restored, IGP may re-converge before LDP completed the FEC exchange with its neighbor over that interface. This may cause LDP to remove the LFA next hop from the FEC and blackhole traffic.

In order to avoid traffic being blackholed, it is recommended to first enable IGP-LDP synchronization on the interface. The time is expressed in seconds and can have a value between 1 and 1800 seconds. It is also possible to configure an end-of-LIB option to optimize the synchronization time, see the [LDP-IGP Synchronization](#) chapter. On PE-1, the following configures the LDP synchronization timer with a value of 10 seconds on the interfaces "int-PE-1-PE-2" and "int-PE-1-PE-3":

```

# on PE-1:
configure {
  router "Base" {
    interface "int-PE-1-PE-2" {
      port 1/1/1
      ldp-sync-timer {
        seconds 10
      }
    }
    ipv4 {
      primary {
        address 192.168.12.1
        prefix-length 30
      }
    }
  }
  interface "int-PE-1-PE-3" {
    port 1/1/2
    ldp-sync-timer {

```

```

        seconds 10
    }
    ipv4 {
        primary {
            address 192.168.13.1
            prefix-length 30
        }
    }
}

```

The configuration on the other nodes is similar.

When this timer is enabled, it means that when an interface is restored, the IGP will advertise this link in the network with an infinite metric. The **ldp-sync-timer** is started, LDP adjacencies are brought up together with a label exchange. After the **ldp-sync-timer** expires, the normal metric is advertised in the network again.

Verify data path

Data path verification is performed using a Layer 2 Epipe service. Traffic generator ports are connected toward PE-1 and PE-5, and an Epipe service is created using an MPLS LDP based Service Distribution Path (SDP) on both PE-1 and PE-5. The service configuration on PE-1 is as follows:

```

# on PE-1:
configure {
    service {
        epipe "Epipe 1" {
            admin-state enable
            service-id 1
            customer "1"
            service-mtu 1450
            spoke-sdp 15:1 {
            }
            sap 1/1/3:1 {
            }
        }
        sdp 15 {
            admin-state enable
            delivery-type mpls
            ldp true
            far-end {
                ip-address 192.0.2.5
            }
        }
    }
}

```

The service configuration on PE-5 is similar.

The IS-IS Level 2 metric value on the interface between PE-4 and PE-5 is decreased to 5, see [Figure 235: Data verification in the direction from PE-1 to PE-5 using Epipe service.](#)

```

# on PE-4:
configure {
    router "Base" {
        isis 0 {
            interface "int-PE-4-PE-5" {
                interface-type point-to-point
                level 2 {
                    metric 5
                }
            }
        }
    }
}

```

```

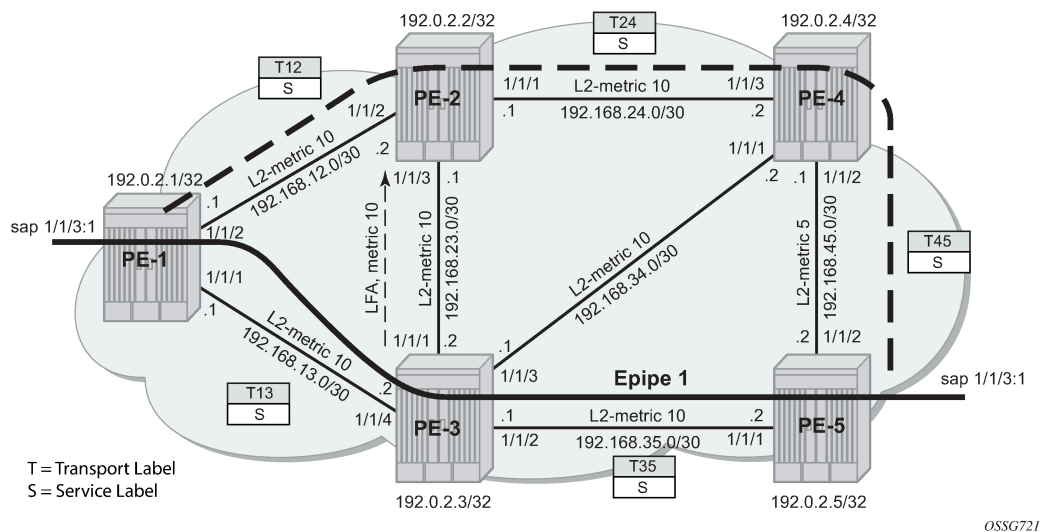
}

# on PE-5:
configure {
  router "Base" {
    isis 0 {
      interface "int-PE-5-PE-4" {
        interface-type point-to-point
        level 2 {
          metric 5
        }
      }
    }
  }
}

```

Figure 235: Data verification in the direction from PE-1 to PE-5 using Epipe service shows the preferred data path for Epipe 1 via PE-3 and the LFA for PE-5 that is protecting node PE-3.

Figure 235: Data verification in the direction from PE-1 to PE-5 using Epipe service



In this setup, the following LFA for prefix PE-5 from PE-1 is protecting the node PE-3:

```

[/]
A:admin@PE-1# show router isis topology lfa detail

=====
Rtr Base ISIS Instance 0 Topology Table
=====
-----
IS-IS IP paths (MT-ID 0), Level 2
-----
---snip---

Node       : PE-5.00
Nexthop    : PE-3
Interface  : int-PE-1-PE-3
SNPA      : none                               Metric      : 20

LFA nh     : PE-2
LFA intf   : int-PE-1-PE-2                     LFA Metric   : 25

```

```
LFA type : nodeProtection

=====

[/]
A:admin@PE-1# show router isis routes alternative ip-prefix-prefix-length 192.0.2.5

=====
Rtr Base ISIS Instance 0 Route Table (alternative)
=====
Prefix[Flags]                Metric    Lvl/Typ    Ver.  SysID/Hostname
NextHop                      MT        AdminTag/SID[F]
Alt-Nexthop                   Alt-      Alt-Type
                               Metric
-----
192.0.2.5/32                  20        2/Int.     6     PE-3
192.168.13.2                  0         0         0
192.168.12.2(L)              25        NP
-----
No. of Routes: 1 (1 path)
-----
Flags      : L = Loop-Free Alternate nexthop
Alt-Type   : LP = linkProtection, NP = nodeProtection
SID[F]     : R = Re-advertisement
            N = Node-SID
            nP = no penultimate hop POP
            E = Explicit-Null
            V = Prefix-SID carries a value
            L = value/index has local significance
=====
```

In normal conditions, MPLS traffic from PE-1 toward PE-5 over Epipe 1 will have two MPLS labels: an outer (transport) label given by LDP protocol, swapped on each intermediate LSR and an inner (service) label given by T-LDP, the same end-to-end. See the following show commands.

The T-LDP service label is S (524282):

```
[/]
A:admin@PE-1# show router ldp bindings services service-id 1

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  S - Status Signaled Up, D - Status Signaled Down, e - Label ELC
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
Service Type:
  E - Epipe Service, V - VPLS Service, M - Mirror Service
  A - Apipe Service, F - Fpipe Service, I - IES Service, R - VPRN service
  P - Ipipe Service, C - Cpipe Service
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP Service FEC 128 Bindings
=====
Type          VCId      SDPId      LMTU
Peer          SvcId     IngLbl     RMTU
              EgrLbl
-----
```

```

E-Eth                               1          15          1436
192.0.2.5:0                         1          524282U    1436
                                      524282S

-----
No. of VC Labels: 1
=====

LDP Service FEC 129 Bindings
=====
SAII                                AGII        IngLbl      LMTU
TAII                                Type        EgrLbl      RMTU
Peer                                SvcId       SDPIId

-----
No Matching Entries Found
=====

```

The transport LDP label between PE-1 and PE-3 for prefix 192.0.2.5/32 is T13 (524283):

```

[/]
A:admin@PE-1# show router ldp bindings active prefixes prefix 192.0.2.5/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static           (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop    (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding

=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                                Op
IngLbl                                EgrLbl
EgrNextHop                            EgrIf/LspId
-----
192.0.2.5/32                          Push
--                                     524283
192.168.13.2                          1/1/2

192.0.2.5/32                          Push
--                                     524283BU
192.168.12.2                          1/1/1

192.0.2.5/32                          Swap
524283                                 524283
192.168.13.2                          1/1/2

192.0.2.5/32                          Swap
524283                                 524283BU
192.168.12.2                          1/1/1

-----
No. of IPv4 Prefix Active Bindings: 4

```

The transport LDP label between PE-3 and PE-5 for prefix 192.0.2.5/32 is T35 (524287):

```
[/]
A:admin@PE-3# show router ldp bindings active prefixes prefix 192.0.2.5/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.3)
(IPv6 LSR ID ::)
=====
---snip---
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.5/32                         Push
--                                   524287
192.168.35.2                         1/1/2

192.0.2.5/32                         Push
--                                   524283BU
192.168.34.2                         1/1/4

192.0.2.5/32                         Swap
524283                               524287
192.168.35.2                         1/1/2

192.0.2.5/32                         Swap
524283                               524283BU
192.168.34.2                         1/1/4

-----
No. of IPv4 Prefix Active Bindings: 4
=====
```

When PE-3 reboots, PE-1 performs an immediate swap to LFA next-hop for prefix 192.0.2.5/32 bypassing PE-3. The service label remains the same; only the transport labels can change on the network ports from PE-1 to PE-2, from PE-2 to PE-4, and from PE-4 to PE-5. See the following show commands.



Note:

The LDP FRR MPLS label stack will never contain more than two labels. This is different from RSVP-TE FRR facility mode which uses a three-label MPLS stack.

The T-LDP service label is S (524282):

```
[/]
A:admin@PE-1# show router ldp bindings services service-id 1

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  S - Status Signaled Up, D - Status Signaled Down, e - Label ELC
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
Service Type:
```



```

E - Epipe Service, V - VPLS Service, M - Mirror Service
A - Apipe Service, F - Fpipe Service, I - IES Service, R - VPRN service
P - Ipipe Service, C - Cpipe Service
FEC Flags:
LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
BA - ASBR Backup FEC
=====
LDP Service FEC 128 Bindings
=====
Type                VCId      SDPId      LMTU
Peer                SvcId     IngLbl     RMTU
                   EgrLbl
-----
E-Eth                1         15         1436
192.0.2.5:0         1         524282U    1436
                   524282S
-----
No. of VC Labels: 1
=====
---snip---

```

The transport LDP label value between PE-1 and PE-2 for prefix 192.0.2.5/32 is the same label (previously tagged as BU) as before the node failure event: T12 (524283):

```

[/]
A:admin@PE-1# show router ldp bindings active prefixes prefix 192.0.2.5/32
=====
---snip---
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                Op
IngLbl                EgrLbl
EgrNextHop            EgrIf/LspId
-----
192.0.2.5/32          Push
--                    524283
192.168.12.2          1/1/1
-----
192.0.2.5/32          Swap
524283                524283
192.168.12.2          1/1/1
-----
No. of IPv4 Prefix Active Bindings: 2
=====

```

The transport LDP label between PE-2 and PE-4 for prefix 192.0.2.5/32 is T24 (524283):

```

[/]
A:admin@PE-2# show router ldp bindings active prefixes prefix 192.0.2.5/32
=====
---snip---
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                Op
IngLbl                EgrLbl
EgrNextHop            EgrIf/LspId
-----

```

```
-----
192.0.2.5/32          Push
--                  524283
192.168.24.2         1/1/1

192.0.2.5/32          Swap
524283              524283
192.168.24.2         1/1/1

-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

The transport LDP label between PE-4 and PE-5 for prefix 192.0.2.5/32 is T45 (524287):

```
[/]
A:admin@PE-4# show router ldp bindings active prefixes prefix 192.0.2.5/32

=====
---snip---
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                Op
IngLbl                EgrLbl
EgrNextHop            EgrIf/LspId
-----
192.0.2.5/32          Push
--                  524287
192.168.45.2         1/1/1

192.0.2.5/32          Swap
524283              524287
192.168.45.2         1/1/1

-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

Additional topics

Metric change

On PE-4 and PE-5, the default level 2 metrics are restored, as follows:

```
[ex:/configure router "Base" isis 0 interface "int-PE-4-PE-5" level 2]
A:admin@PE-4# delete metric
```

```
[ex:/configure router "Base" isis 0 interface "int-PE-5-PE-4" level 2]
A:admin@PE-5# delete metric
```

When the IS-IS level 2 metric between PE-2 and PE-3 changes to 30, then 100% LFA coverage is no longer possible. The IS-IS level 2 metric is modified as follows:

```
[ex:/configure router "Base" isis 0 interface "int-PE-2-PE-3" level 2]
```

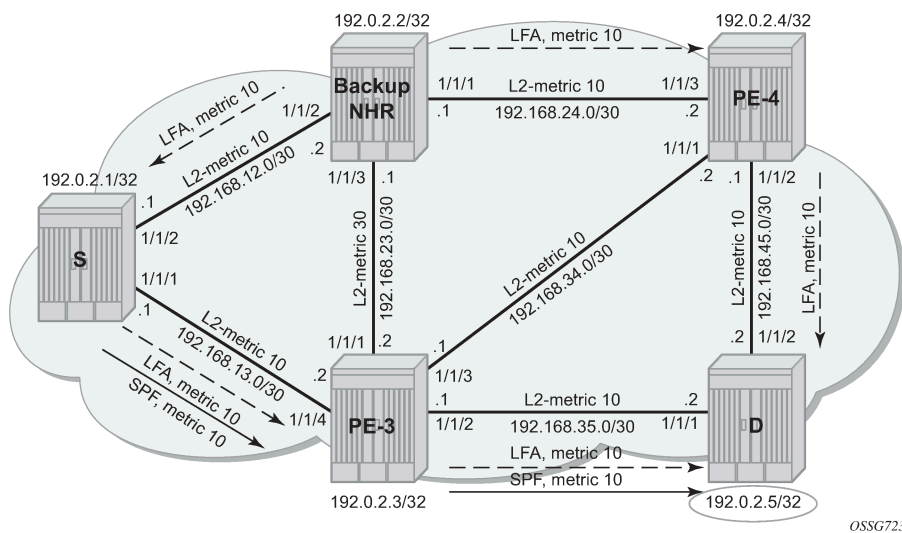
```
A:admin@PE-2# metric 30
```

```
[ex:/configure router "Base" isis 0 interface "int-PE-3-PE-2" level 2]
A:admin@PE-3# metric 30
```

On PE-1, Inequality 3 formula will find LFA next-hop coverages for prefix PE-4 and PE-5. Inequality formula 1 will find LFA next-hop coverages for prefix PE-4, PE-5, and the subnet between PE-4 and PE-5.

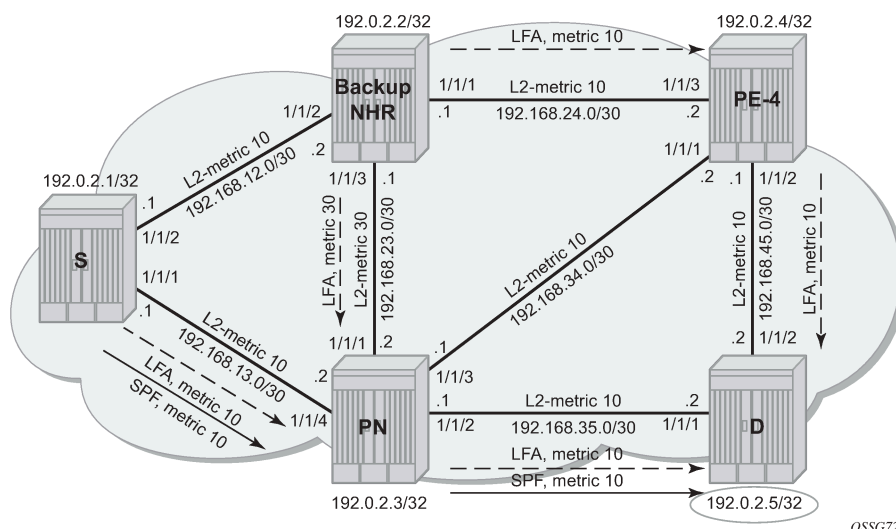
Both inequality formulas are visualized in [Figure 237: LFA computation: Inequality 3 for prefix PE-5 \(D\) on PE-1 \(S\)](#) and [Figure 236: LFA computation: Inequality 1 for prefix PE-5 \(D\) on PE-1 \(S\)](#) for prefix 192.0.2.5/32 (= PE-5) on PE-1 which serves as the source node for LFA next-hop computation.

Figure 236: LFA computation: Inequality 1 for prefix PE-5 (D) on PE-1 (S)



OSSG723

Figure 237: LFA computation: Inequality 3 for prefix PE-5 (D) on PE-1 (S)



Inequality 3 formula:

$$[SP(\text{backup NHR}, D) < \{SP(\text{backup NHR}, PN) + SP(PN, D)\}]$$

For a node LFA next-hop calculation of prefix 192.0.2.5/32 (D) on PE-1, this means that the shortest path from backup next-hop router PE-2 toward destination PE-5 must be smaller than the sum of the shortest path from backup next-hop router PE-2 toward protected node PE-3 with the shortest path from protected node PE-3 to destination PE-5.

The shortest path from backup next-hop router PE-2 toward destination PE-5 is going via PE-4, using IS-IS level 2 metric 10 for interface "int-PE-2-PE-4" and IS-IS level 2 metric 10 for interface "int-PE-4-PE-5". The shortest path from backup next-hop router (PE-2) toward protected node (PE-3) uses IS-IS level 2 metric 30 for interface "int-PE-2-PE-3". The shortest path from protected node (PE-3) to destination (PE-5) uses IS-IS level 2 metric 10 for interface "int-PE-3-PE-5". The calculation is as follows:

```
Prefix 192.0.2.5/32: SP (PE-2, PE-5) < SP (PE-2, PE-3) + SP (PE-3, PE-5)
                    10 + 10 < 30 + 10 => OK
```

Inequality 1 formula:

$$SP(\text{backup NHR}, D) < \{SP(\text{backup NHR}, S) + SP(S, D)\}$$

For a link LFA next-hop calculation of prefix 192.0.2.5/32 (D) on PE-1, this means that the shortest path from backup next-hop router PE-2 toward destination PE-5 must be smaller than the sum of the shortest path from backup next-hop router PE-2 toward source PE-1 with the shortest path from source PE-1 to destination PE-5.

The shortest path from backup next-hop router PE-2 toward destination PE-5 is going over PE-4, using IS-IS level 2 metric 10 for interface "int-PE-2-PE-4" and IS-IS level 2 metric 10 for interface "int-PE-4-PE-5". The shortest path from backup next-hop router PE-2 toward source PE-1 uses IS-IS level 2 metric 10 for interface "int-PE-2-PE-1". The shortest path from source PE-1 to destination PE-5 follows the normal SPF

calculation, going over PE-3, using IS-IS level 2 metric 10 for interface "int-PE-1-PE-3", and IS-IS level 2 metric 10 for interface "int-PE-3-PE-5".

The calculation is as follows:

```
Prefix 192.0.2.5/32 : SP(PE-2,PE-5) < SP(PE-2,PE-1) + SP(PE-1,PE-5)
                    10 + 10      <    10 + (10 + 10)                => OK
```

For completeness, all the other Inequality 1 calculations on PE-1 are as follows:

```
Prefix 192.0.2.2/32 : SP(PE-3,PE-2) < SP(PE-3,PE-1) + SP(PE-1,PE-2)           => NOK
                    30      <    10 + 10
Prefix 192.0.2.3/32 : SP(PE-2,PE-3) < SP(PE-2,PE-1) + SP(PE-1,PE-3)           => NOK
                    30      <    10 + 10
Prefix 192.0.2.4/32 : SP(PE-3,PE-4) < SP(PE-3,PE-1) + SP(PE-1,PE-4)           => OK
                    10      <    10 + (10 + 10)
Prefix 192.168.23.0/30 : SP(PE-3,D) < SP(PE-3,PE-1) + SP(PE-1,D)             => NOK
                    30      <    10 + (10 + 10)
Prefix 192.168.24.0/30 : SP(PE-3,D) < SP(PE-3,PE-1) + SP(PE-1,D)             => NOK
                    30 + 10 <    10 + (10 + 10)
Prefix 192.168.34.0/30 : SP(PE-2,D) < SP(PE-2,PE-1) + SP(PE-1,D)             => NOK
                    30 + 10 <    10 + (10 + 10)
Prefix 192.168.35.0/30 : SP(PE-2,D) < SP(PE-2,PE-1) + SP(PE-1,D)             => NOK
                    30 + 10 <    10 + (10 + 10)
Prefix 192.168.45.0/30 : SP(PE-3,D) < SP(PE-3,PE-1) + SP(PE-1,D)             => OK
                    10 + 10 <    10 + (10 + 10 + 10)
```

Considering all Inequality 1 calculations, only three of these are valid (OK).

In SR OS, the **show router isis lfa-coverage** summary command exists for LFA coverage on the router:

```
[/]
A:admin@PE-1# show router isis lfa-coverage

=====
Rtr Base ISIS Instance 0 LFA Coverage
=====
Topology      Level  Node      IPv4      IPv6
-----
IPV4 Unicast  L1    0/0(0%)   3/9(33%)  0/0(0%)
IPV6 Unicast  L1    0/0(0%)   0/0(0%)   0/0(0%)
IPV4 Multicast L1    0/0(0%)   0/0(0%)   0/0(0%)
IPV6 Multicast L1    0/0(0%)   0/0(0%)   0/0(0%)
IPV4 Unicast  L2    2/4(50%)  3/9(33%)  0/0(0%)
IPV6 Unicast  L2    0/0(0%)   0/0(0%)   0/0(0%)
IPV4 Multicast L2    0/0(0%)   0/0(0%)   0/0(0%)
IPV6 Multicast L2    0/0(0%)   0/0(0%)   0/0(0%)
=====
```

The **show router route-table alternative** command on PE-1 shows which prefixes are protected:

```
[/]
A:admin@PE-1# show router route-table alternative

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]      Type  Proto  Age  Pref
Next Hop[Interface Name]  Metric
Alt-NextHop              Alt-
                          Metric
```

```

-----
192.0.2.1/32                               Local  Local  00h39m57s  0
  system
192.0.2.2/32                               Remote ISIS  00h39m21s  18
  192.168.12.2
192.0.2.3/32                               Remote ISIS  00h03m26s  18
  192.168.13.2
192.0.2.4/32                               Remote ISIS  00h37m57s  18
  192.168.12.2
  192.168.13.2 (LFA)
192.0.2.5/32                               Remote ISIS  00h03m26s  18
  192.168.13.2
  192.168.12.2 (LFA)
192.168.12.0/30                           Local  Local  00h39m57s  0
  int-PE-1-PE-2
192.168.13.0/30                           Local  Local  00h39m57s  0
  int-PE-1-PE-3
192.168.23.0/30                           Remote ISIS  00h00m48s  18
  192.168.12.2
192.168.24.0/30                           Remote ISIS  00h39m21s  18
  192.168.12.2
192.168.34.0/30                           Remote ISIS  00h03m26s  18
  192.168.13.2
192.168.35.0/30                           Remote ISIS  00h03m26s  18
  192.168.13.2
192.168.45.0/30                           Remote ISIS  00h02m29s  18
  192.168.12.2
  192.168.13.2 (LFA)
-----
No. of Routes: 12
Flags: n = Number of times nexthop is repeated
      Backup = BGP backup route
      LFA = Loop-Free Alternate nexthop
      S = Sticky ECMP requested
=====

```

The default IS-IS level 2 metrics are restored, as follows:

```
[ex:/configure router "Base" isis 0 interface "int-PE-2-PE-3" level 2]
A:admin@PE-2# delete metric
```

```
[ex:/configure router "Base" isis 0 interface "int-PE-3-PE-2" level 2]
A:admin@PE-3# delete metric
```

IS-IS overload

As stated in RFC 3137, *OSPF Stub Router Advertisement*, sometimes it is useful and desirable for a router not to be a transit node. For those cases, it is also desirable not to have that router used as transit node during the LFA next-hop computation. Within the IS-IS protocol, this is achieved by configuring IS-IS overload. When other routers detect that IS-IS overload is configured, they will only use this router for packets destined to the overloaded router's directly connected networks and IP prefixes.

In the IS-IS overload context, the **max-metric** can be configured, as follows:

```
[ex:/configure router "Base" isis 0]
A:admin@PE-2# overload ?

overload
```

max-metric - Advertise transit links with maximum metric instead of setting overload bit

As an example, IS-IS overload is configured on PE-2, as follows:

```
*[ex:/configure router "Base" isis 0 overload]
A:admin@PE-2# max-metric false
```

With IS-IS overload on PE-2 configured **max-metric false**, the LFA coverage on PE-1 is as follows:

```
[/]
A:admin@PE-1# show router isis lfa-coverage

=====
Rtr Base ISIS Instance 0 LFA Coverage
=====
Topology          Level   Node           IPv4            IPv6
-----
IPV4 Unicast      L1      0/0(0%)        3/9(33%)       0/0(0%)
IPV6 Unicast      L1      0/0(0%)        0/0(0%)       0/0(0%)
IPV4 Multicast    L1      0/0(0%)        0/0(0%)       0/0(0%)
IPV6 Multicast    L1      0/0(0%)        0/0(0%)       0/0(0%)
IPV4 Unicast      L2      1/4(25%)       3/9(33%)       0/0(0%)
IPV6 Unicast      L2      0/0(0%)        0/0(0%)       0/0(0%)
IPV4 Multicast    L2      0/0(0%)        0/0(0%)       0/0(0%)
IPV6 Multicast    L2      0/0(0%)        0/0(0%)       0/0(0%)
=====
```

```
[/]
A:admin@PE-1# show router route-table alternative

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type  Proto  Age           Pref
Next Hop[Interface Name]                         Metric
Alt-NextHop                                       Alt-
Metric
-----
192.0.2.1/32                                     Local  Local  00h45m01s    0
system
192.0.2.2/32                                     Remote ISIS   00h44m26s 18
192.168.12.2                                     10
192.168.13.2 (LFA)                               20
192.0.2.3/32                                     Remote  ISIS   00h08m30s    18
192.168.13.2                                     10
192.0.2.4/32                                     Remote  ISIS   00h00m29s    18
192.168.13.2                                     20
192.0.2.5/32                                     Remote  ISIS   00h08m30s    18
192.168.13.2                                     20
192.168.12.0/30                                  Local  Local  00h45m01s    0
int-PE-1-PE-2                                    0
192.168.13.0/30                                  Local  Local  00h45m01s    0
int-PE-1-PE-3                                    0
192.168.23.0/30                                     Remote ISIS   00h04m15s 18
192.168.12.2                                     20
192.168.13.2 (LFA)                               30
192.168.24.0/30                                     Remote ISIS   00h44m26s 18
192.168.12.2                                     20
192.168.13.2 (LFA)                               30
192.168.34.0/30                                  Remote  ISIS   00h08m30s    18
```

```

192.168.13.2          20
192.168.35.0/30      Remote ISIS    00h08m30s 18
192.168.13.2          20
192.168.45.0/30      Remote ISIS    00h00m29s 18
192.168.13.2          30
-----
No. of Routes: 12
Flags: n = Number of times nexthop is repeated
      Backup = BGP backup route
      LFA = Loop-Free Alternate nexthop
      S = Sticky ECMP requested
=====

```

```

[/]
A:admin@PE-1# show router isis routes alternative

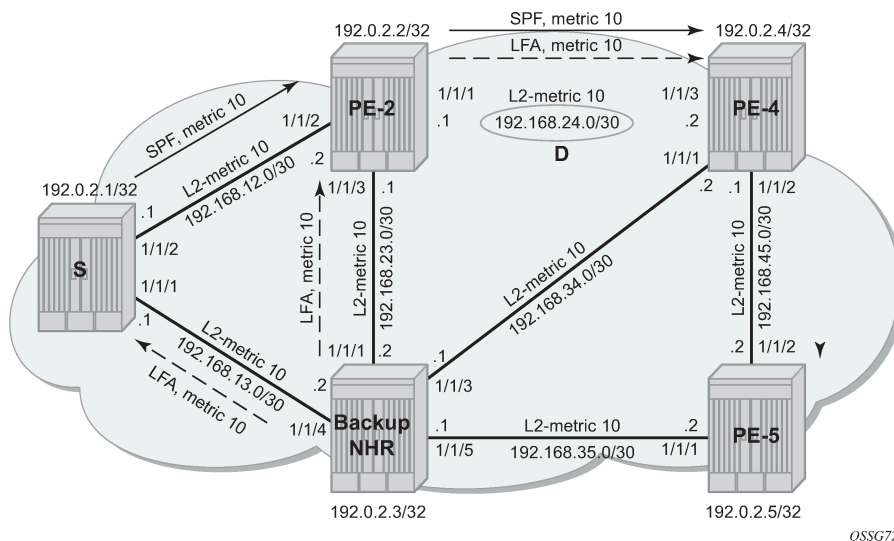
=====
Rtr Base ISIS Instance 0 Route Table (alternative)
=====
Prefix[Flags]          Metric    Lvl/Typ    Ver.  SysID/Hostname
NextHop                MT         AdminTag/SID[F]
Alt-Nexthop           Alt-      Alt-Type
                        Metric
-----
192.0.2.1/32           0         2/Int.     2     PE-1
  0.0.0.0              0         0
192.0.2.2/32          10        2/Int.     3     PE-2
  192.168.12.2         0         0
  192.168.13.2(L)      20        LP
192.0.2.3/32          10        2/Int.     16    PE-3
  192.168.13.2         0         0
192.0.2.4/32          20        2/Int.     24    PE-3
  192.168.13.2         0         0
192.0.2.5/32          20        2/Int.     16    PE-3
  192.168.13.2         0         0
192.168.12.0/30       10        2/Int.     2     PE-1
  0.0.0.0              0         0
192.168.13.0/30       10        2/Int.     16    PE-1
  0.0.0.0              0         0
192.168.23.0/30       20        2/Int.     22    PE-2
  192.168.12.2         0         0
  192.168.13.2(L)      30        LP
192.168.24.0/30       20        2/Int.     3     PE-2
  192.168.12.2         0         0
  192.168.13.2(L)      30        LP
192.168.34.0/30       20        2/Int.     16    PE-3
  192.168.13.2         0         0
192.168.35.0/30       20        2/Int.     16    PE-3
  192.168.13.2         0         0
192.168.45.0/30       30        2/Int.     24    PE-3
  192.168.13.2         0         0
-----
No. of Routes: 12 (12 paths)
-----
Flags      : L = Loop-Free Alternate nexthop
Alt-Type   : LP = linkProtection, NP = nodeProtection
SID[F]     : R = Re-advertisement
            N = Node-SID
            nP = no penultimate hop POP
            E = Explicit-Null
            V = Prefix-SID carries a value
            L = value/index has local significance
=====

```


On PE-1, only three Inequality 1 calculations are possible, as seen in the previous show commands. The Inequality 1 calculation on PE-1 for destination 192.168.24.0/30 is as follows:

```
[SP(backup NHR,D) < {SP(backup NHR,S) + SP(S,D)}]
SP(PE-3,D) < SP(PE-3,PE-1) + SP(PE-1,D)
10 + 10 < 10 + (10 + 10) => OK
```

Figure 238: IS-IS overload on PE-2, Inequality 1 for 192.168.24.0/30 (D) on PE-1 (S)



The overload configuration on PE-2 is removed as follows:

```
[ex:/configure router "Base" isis 0]
A:admin@PE-2# delete overload
```

Conclusion

In production MPLS networks where FRR needs to be deployed, a trade off must be made between RSVP-TE FRR versus LDP FRR. The two main advantages of using LDP FRR compared to RSVP FRR are the simple configuration and the fact that LFA next-hop calculation is a local decision, which means there are no interoperability issues when working in a multi-vendor environment. The main disadvantage of using LDP FRR is that LFA next-hop calculation has to deal with the source-route paradigm (inequality formulas exclude a path going over the original source router).

Multicast Label Distribution Protocol

This chapter provides information about multicast Label Distribution Protocol (mLDP).

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter is applicable to SR OS and was initially written for Release 13.0.R6. The CLI in the current edition corresponds to Release 23.7.R2.

In this chapter, the emphasis is on IPv4. However, multicast Label Distribution Protocol (mLDP) is also supported on IPv6 interfaces.

Overview

Multicast Label Distribution Protocol provides extensions to LDP for the setup of point-to-multipoint (P2MP) Label Switched Paths (LSPs) and multipoint-to-multipoint (MP2MP) LSPs in MPLS networks.

The protocol is described in RFC 6388, *Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths*.

Multicast LSPs can be applied for IP multicast or support for multicast in BGP/MPLS Layer 3 Virtual Private Networks (L3 VPNs).

Compared to RSVP P2MP LSPs, mLDP P2MP LSPs are easier to configure and the setup direction is different. Whereas the RSVP P2MP LSPs are set up from the root node toward the leaf nodes, mLDP P2MP LSPs are set up from the leaf nodes toward the root node.

P2MP Terminology

The following terminology is used.

Table 14: Terminology

Node	Description
Ingress / Root	P2MP LSPs have just one ingress (root) node. The root node receives IP multicast traffic and maps the traffic to a P2MP LSP (push). The node may perform MPLS multicast replication.
Egress / Leaf	P2MP LSPs have multiple egress (leaf) nodes. A leaf node removes data packets from a P2MP LSP (pop) for further processing. The node may perform IP multicast replication.

Node	Description
Transit	A transit Label Switching Router (LSR) can reach the root node via a directly connected upstream LSR. A transit LSR also has one or more directly connected downstream LSRs. The LSR swaps the MPLS label and may perform MPLS multicast replication.
Branch	A branch LSR is a transit LSR that has several directly connected LSRs. The LSR swaps the MPLS label and performs MPLS multicast replication.
Bud	A bud node is an egress node, but also a transit node. The node has directly connected receivers and also one or more directly connected downstream LSRs.

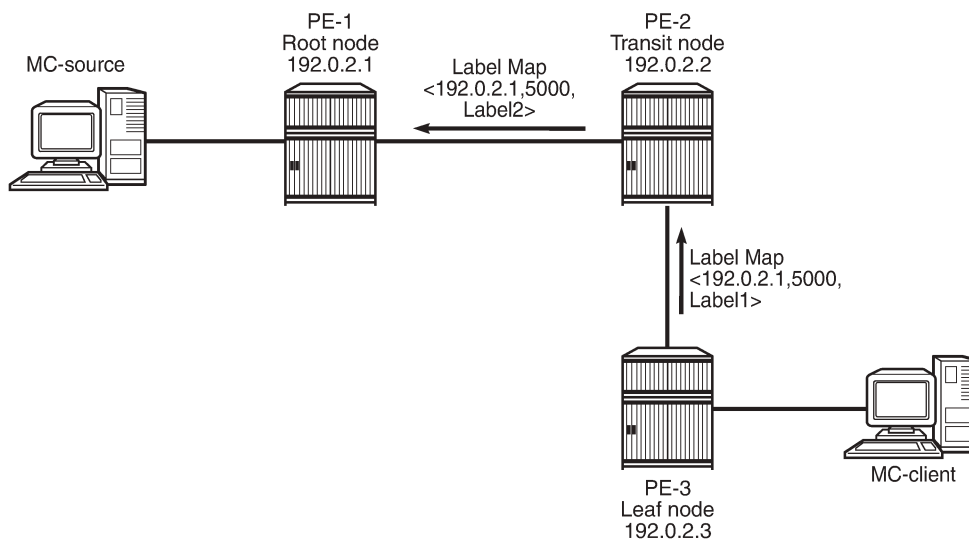
Setup of mLDP P2MP LSP

The setup of the P2MP LSP in the control plane is as follows.

1. The leaf node initiates a tree setup according to what is configured. Mandatory parameters are the IP address of the root and an opaque value. The leaf node sends an LDP label map message to its upstream hop toward the root node of the tree.
2. Each transit node receives the LDP label map message and sends another LDP label map message to its upstream hop toward the root node of the tree. Each label can be different.
3. The root node receives the LDP label map message.

The label map message contains the root node address, an opaque value, and a label. In the example in Figure 1, the root node address is 192.0.2.1 and the opaque value is 5000.

Figure 239: Setup of mLDP P2MP LSP



25513

After the LDP label map messages are sent in the control plane, the nodes program pop, swap, or push entries for the corresponding labels in the data plane.

1. The leaf node programs a pop entry for the label it sent upstream.
2. The transit node programs a swap entry for the label it sent upstream with the next-hop address and the label it received from the downstream node.
3. The root node programs a push entry and a next-hop address for the label it received from the downstream node.

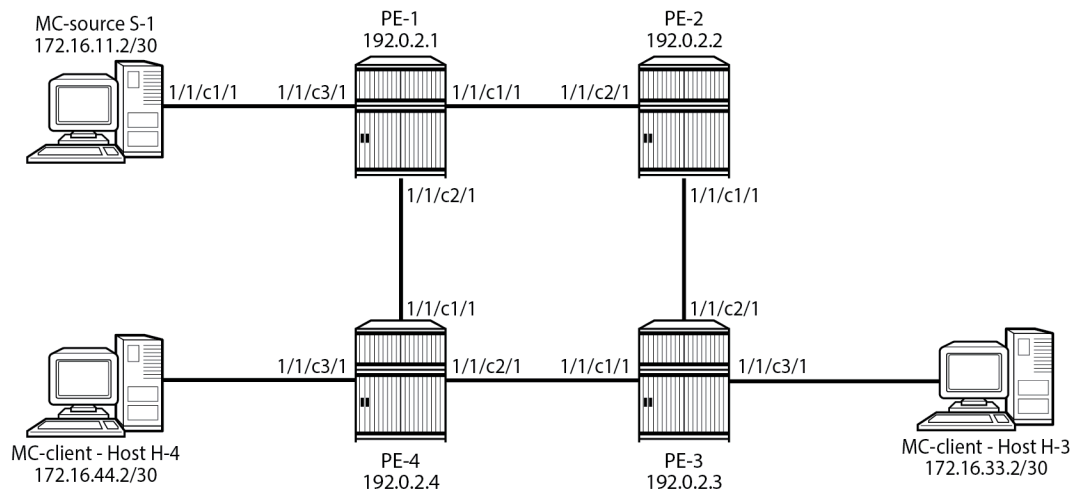
If multiple Equal Cost Multi-Path (ECMP) paths exist between two adjacent nodes, the upstream node of the multicast receiver must program all the entries in the forwarding plane. Only one entry must be active based on the ECMP hashing algorithm.

Configuration

The example topology shown in [Figure 240: Example topology](#) is used. The multicast source S-1 is connected to root node PE-1. PE-2 or PE-4 is the transit node for traffic destined for PE-3. There are two leaf nodes: PE-3 and PE-4. Multicast client H-3 is connected to PE-3, whereas multicast client H-4 is connected to PE-4.

Under normal circumstances, PE-2 is the transit node for traffic toward PE-3 and PE-4 is an egress node. If PE-4 is the transit node for traffic toward PE-3, while it also has a directly connected receiver, PE-4 is a bud node.

Figure 240: Example topology



25514b

Configure LDP P2MP LSP

Initial configuration

The PEs should have the following initial configuration:

- Cards, MDAs, ports
- Router interfaces
- IGP (OSPF or IS-IS)

As an example, the router interfaces and OSPF configuration on PE-1 are as follows:

```
# on PE-1:
configure {
  router "Base" {
    interface "int-PE-1-PE-2" {
      port 1/1/c1/1
      ipv4 {
        primary {
          address 192.168.12.1
          prefix-length 30
        }
      }
    }
    interface "int-PE-1-PE-4" {
      port 1/1/c2/1
      ipv4 {
        primary {
          address 192.168.14.1
          prefix-length 30
        }
      }
    }
    interface "int-PE-1-S-1" {
      port 1/1/c3/1
      ipv4 {
        primary {
          address 172.16.11.1
          prefix-length 30
        }
      }
    }
    interface "system" {
      ipv4 {
        primary {
          address 192.0.2.1
          prefix-length 32
        }
      }
    }
  }
  ospf 0 {
    admin-state enable
    area 0.0.0.0 {
      interface "int-PE-1-PE-2" {
        interface-type point-to-point
      }
      interface "int-PE-1-PE-4" {
        interface-type point-to-point
      }
      interface "int-PE-1-S-1" {
        interface-type point-to-point
      }
      interface "system" { }
    }
  }
}
```

Enabling mLDP

When LDP is enabled, mLDP is enabled by default

The following command enables mLDP on a specific interface:

```
configure {
  router "Base" {
    ldp {
      interface-parameters {
        interface "<ip-int-name>" {
          ipv4 {
            fec-type-capability {
              p2mp-ipv4 true
            }
          }
        }
      }
    }
  }
}
```

Enable LDP (including mLDP) on the router interfaces of PE-1, as follows:

```
#on PE-1:
configure {
  router "Base" {
    ldp {
      interface-parameters {
        interface "int-PE-1-PE-2" {
          ipv4 { }
        }
        interface "int-PE-1-PE-4" {
          ipv4 { }
        }
      }
    }
  }
}
```

Verify that mLDP is enabled (P2MP: Enabled), as follows:

```
[/]
A:admin@PE-1# show router ldp status

=====
LDP Status for IPv4 LSR ID 192.0.2.1
          IPv6 LSR ID ::
=====
---snip---
Admin State      : Up
IPv4 Oper State  : Up
IPv4 Up Time     : 0d 00:04:29
IPv4 Oper Down Rea*: n/a
IPv4 Oper Down Eve*: 0
IPv6 Oper State  : Down
IPv6 Down Time   : 0d 00:04:29
IPv6 Oper Down Reason: systemIpDown
IPv6 Oper Down Events: 0
---snip---
-----
Capabilities
-----
Dynamic          : Enabled
IPv4 Prefix Fec  : Enabled
Service Fec128   : Enabled
MP MBB           : Enabled
Unrecognized Notif*: Enabled
P2MP             : Enabled
IPv6 Prefix Fec  : Enabled
Service Fec129   : Enabled
Overload         : Enabled
=====
* indicates that the corresponding row element may have been truncated.
```

Verify that mLDP is enabled on the interface "int-PE-1-PE-2" (**IPv4 P2MP Fec Cap**), as follows:

```
[/]
A:admin@PE-1# show router ldp interface "int-PE-1-PE-2" detail

=====
LDP Interfaces
=====

Interface "int-PE-1-PE-2"
=====
-----
BASE
-----
Admin State      : Up           Oper State      : Up
BFDP Status      : Disabled      Load Bal Wt     : None
-----
IPv4
-----
IPv4 Admin State : Up           IPv4 Oper State : Up
Last Oper Chg    : 0d 00:01:39
Hold Time        : 15           Hello Factor     : 3
Oper Hold Time   : 15
Keepalive Timeout : 30         Keepalive Factor : 3
Transport Addr   : System       Last Modified    : 09/06/23 21:40:48
Active Adjacencies : 1
Local LSR Type   : System
Local LSR        : None         32-BitLocalLsr  : Disabled
IPv4 Pfx Fec Cap : Enabled      IPv6 Pfx Fec Cap : Enabled
IPv4 P2MP Fec Cap : Enabled    IPv6 P2MP Fec Cap: Enabled
=====
No. of Interfaces: 1
=====
```

Disable mLDP on interface "int-PE-1-PE-2" and verify the LDP status again, as follows:

```
# on PE-1:
configure {
  router "Base" {
    ldp {
      interface-parameters {
        interface "int-PE-1-PE-2" {
          ipv4 {
            fec-type-capability {
              p2mp-ipv4 false
            }
          }
        }
      }
    }
  }
}
```

```
[/]
A:admin@PE-1# show router ldp status

=====
LDP Status for IPv4 LSR ID 192.0.2.1
IPv6 LSR ID ::
=====
---snip---
Admin State      : Up
IPv4 Oper State  : Up           IPv6 Oper State  : Down
---snip---
-----
Capabilities
-----
```

```
Dynamic          : Enabled          P2MP           : Enabled
IPv4 Prefix Fec  : Enabled          IPv6 Prefix Fec  : Enabled
Service Fec128   : Enabled          Service Fec129   : Enabled
MP MBB           : Enabled          Overload         : Enabled
Unrecognized Notif*: Enabled
```

=====

* indicates that the corresponding row element may have been truncated.

P2MP LDP is still enabled on the router, but it is disabled on interface "int-PE-1-PE-2", which can be verified as follows:

```
[/]
A:admin@PE-1# show router ldp interface "int-PE-1-PE-2" detail

=====
LDP Interfaces
=====

Interface "int-PE-1-PE-2"
=====
-----
BASE
-----
Admin State      : Up              Oper State       : Up
BFDF Status      : Disabled         Load Bal Wt     : None
-----
IPv4
-----
IPv4 Admin State : Up              IPv4 Oper State : Up
---snip---
IPv4 Pfx Fec Cap : Enabled         IPv6 Pfx Fec Cap : Enabled
IPv4 P2MP Fec Cap : Disabled      IPv6 P2MP Fec Cap: Enabled
=====
No. of Interfaces: 1
=====
```

P2MP multicast forwarding can be disabled per LDP interface. Disabling P2MP multicast forwarding does not prevent LDP from exchanging P2MP FEC elements on that interface in the control plane. In the data plane, the forwarding plane is not programmed with the next hop on the outgoing interface that is P2MP disabled.

Configure tunnel interface on the root and leaf nodes

Multicast LDP can be applied in different scenarios. In the following example, a tunnel interface is created on the root and leaf nodes. Other examples are Multicast Virtual Private Network (MVPN) with mLDP and dynamic PIM-mLDP mapping. In several ACG chapters on MVPN, mLDP is chosen; for example, in *Multicast VPN: Use of Wildcard Selective PMSI*.

A tunnel interface needs to be created on the root node, as follows:

```
# on PE-1:
configure {
  router "Base" {
    tunnel-interface {
      ldp-p2mp-root 5000 sender-address 192.0.2.1 { }
    }
  }
}
```


In this example, the tunnel interface gets interface index 73728, as follows:

```
[/]
A:admin@PE-1# show router tunnel-interface

=====
P2MP-RSVP P2MP-LDP Tunnel-Interfaces
=====
LSP/LDP      Type      SenderAddr      IfIndex      RootNode
-----
5000         ldp       192.0.2.1       73728       Yes
-----
Interfaces : 1
=====
```

A similar command is launched on the leaf nodes, but with the keyword **ldp-p2mp-leaf**, as follows:

```
# on PE-3 and PE-4:
configure {
  router "Base" {
    tunnel-interface {
      ldp-p2mp-leaf 5000 sender-address 192.0.2.1 { }
    }
  }
}
```

```
[/]
A:admin@PE-3# show router tunnel-interface

=====
P2MP-RSVP P2MP-LDP Tunnel-Interfaces
=====
LSP/LDP      Type      SenderAddr      IfIndex      RootNode
-----
5000         ldp       192.0.2.1       73728       No
-----
Interfaces : 1
=====
```

A P2MP LSP ping can be sent to verify the P2MP LSP. The options for P2MP LSP ping are as follows:

```
(in classic CLI context)
*A:PE-1# oam p2mp-lsp-ping ?
- p2mp-lsp-ping <lsp-name> [p2mp-instance <instance-name>
  [s2l-dest-address <ipv4-address> [... up to 5]]]
  [ttl <label-ttl>]
- p2mp-lsp-ping ldp <p2mp-identifier> [vpn-recursive-fec]
  [sender-addr <ipv4-address>]
  [leaf-addr <ipv4-address> [... up to 5]]
- p2mp-lsp-ping ldp-ssm source <ip-address> group <ip-address>
  [router <router-instance>|service-name <service-name>]
  [sender-addr <ipv4-address>] [leaf-addr <ipv4-address>]
  [... up to 5]]
- options common to all p2mp-lsp-ping cases: [fc <fc-name> [profile {in|out}]]
  [size <octets>] [timeout <timeout>] [detail]

<lsp-name>          : [64 chars max]
<instance-name>    : [32 chars max]
<ipv4-address>     : a.b.c.d
<in|out>           : in|out - Default: out
<fc-name>          : be|l2|af|l1|h2|ef|h1|nc - Default: be
<octets>           : [1..9786] - Default: 1
<label-ttl>       : [1..255] - Default: 255
```

```

<timeout>          : [1..120] seconds - Default: 10
<detail>           : keyword - displays detailed information
<p2mp-identifier>  : [1..4294967295]
<ldp-ssm>          : keyword - Label Distribution Protocol, Source-Specific
                    Multicast
<ip-address>       : ipv4-address   - a.b.c.d
                    ipv6-address   - x:x:x:x:x:x:x (eight 16-bit pieces)
                                   x:x:x:x:x:d.d.d.d
                                   x - [0..FFFF]H
                                   d - [0..255]D
<router-instance>  : <router-name>|<vprn-svc-id>
                    router-name    - "Base" Default - Base
                    vprn-svc-id    - [1..2147483647]
<service-name>     : [64 chars max]
<vpn-recursive-fec> : keyword - add a VPN Recursive FEC element to the launched packet
                    (useful for pinging a VPN BGP inter-AS Option B leaf)
    
```

Verify the P2MP LSP with the following ping command:

```

A:PE-1# oam p2mp-lsp-ping ldp 5000
P2MP identifier 5000: | 88 bytes MPLS payload

Total Leafs responded = 2
    round-trip min/avg/max = 1.41 / 1.75 / 2.10 ms

Responses based on return code:
EgressRtr(3)=2
    
```

Both leaf nodes have sent a reply. The return code 3 indicates that the replying router is an egress for the Forwarding Equivalence Class (FEC).

For a detailed output per leaf, use the following command:

```

A:PE-1# oam p2mp-lsp-ping ldp 5000 detail
P2MP identifier 5000: | 88 bytes MPLS payload

=====
Leaf Information
=====
From           RTT           Return Code
-----
192.0.2.4      =1.56ms      EgressRtr(3)
192.0.2.3      =2.47ms      EgressRtr(3)
=====

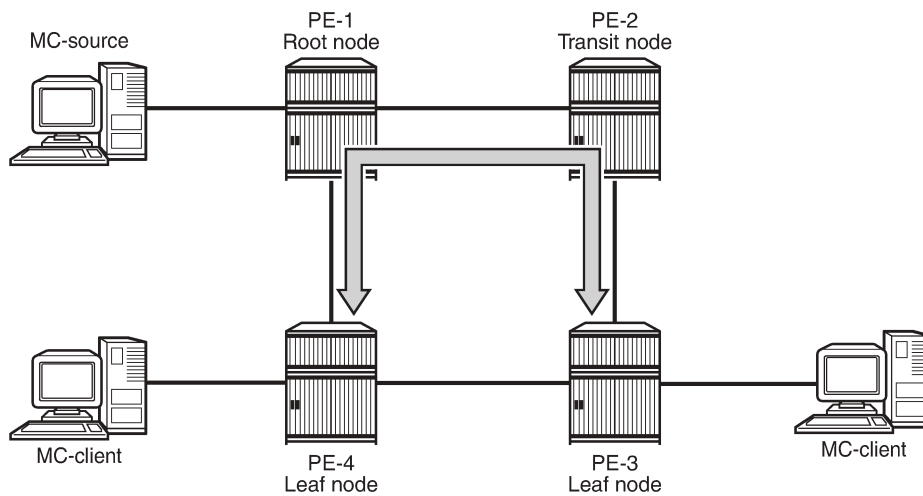
Total Leafs responded = 2
    round-trip min/avg/max = 1.56 / 2.01 / 2.47 ms

Responses based on return code:
EgressRtr(3)=2
    
```

Verify LDP P2MP bindings

The example LDP P2MP LSP is shown in [Figure 241: LDP P2MP LSP](#). In this case, PE-4 is only an egress node and not a bud node.

Figure 241: LDP P2MP LSP



25515

Verify the LDP P2MP bindings on the leaf node PE-4, as follows.
The leaf node programs a pop entry for the label sent upstream.

```
[/]
A:admin@PE-4# show router ldp bindings active p2mp opaque-type generic ipv4

=====
LDP Bindings (IPv4 LSR ID 192.0.2.4)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP Generic IPv4 P2MP Bindings (Active)
=====
P2MP-Id          Interface
RootAddr         Op
IngLbl           EgrLbl
EgrNH            EgrIf/LspId
-----
5000             73728
192.0.2.1       Pop
524283          --
--              --
-----
No. of Generic IPv4 P2MP Active Bindings: 1
=====
```

```
[/]
A:admin@PE-4# show router ldp bindings p2mp opaque-type generic ipv4 detail
```

```

=====
LDP Bindings (IPv4 LSR ID 192.0.2.4)
      (IPv6 LSR ID ::)
=====
Label Status:
      U - Label In Use, N - Label Not In Use, W - Label Withdrawn
      WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
      e - Label ELC
FEC Flags:
      LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
      BA - ASBR Backup FEC
=====
LDP Generic IPv4 P2MP Bindings
=====
-----
P2MP Type      : 1                P2MP-Id       : 5000
Root-Addr     : 192.0.2.1
-----
Peer          : 192.0.2.1:0
Ing Lbl       : 524283U
Egr Lbl       : --
Egr Int/LspId : --
EgrNextHop    : --
Egr. Flags    : None                Ing. Flags : None
=====
No. of Generic IPv4 P2MP Bindings: 1
=====

```

PE-4 is only an egress node and not a transit node. There is no next hop.

Verify the LDP P2MP bindings on the leaf node PE-3, as follows:

```

[/]
A:admin@PE-3# show router ldp bindings active p2mp opaque-type generic ipv4
=====
LDP Bindings (IPv4 LSR ID 192.0.2.3)
      (IPv6 LSR ID ::)
=====
Label Status:
      U - Label In Use, N - Label Not In Use, W - Label Withdrawn
      WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
      e - Label ELC
FEC Flags:
      LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
      BA - ASBR Backup FEC
=====
LDP Generic IPv4 P2MP Bindings (Active)
=====
-----
P2MP-Id      Interface
RootAddr     Op
IngLbl       EgrLbl
EgrNH        EgrIf/LspId
-----
5000         73728
192.0.2.1    Pop
524283       --
--           --
-----
No. of Generic IPv4 P2MP Active Bindings: 1
=====

```

Because PE-3 is an egress node, there is no next hop. The traffic toward PE-3 is sent via transit PE-2 and not via PE-4, as can be verified as follows:

```
[/]
A:admin@PE-3# show router ldp bindings p2mp opaque-type generic ipv4 detail

=====
LDP Bindings (IPv4 LSR ID 192.0.2.3)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP Generic IPv4 P2MP Bindings
=====
P2MP Type      : 1                P2MP-Id      : 5000
Root-Addr     : 192.0.2.1
-----
Peer          : 192.0.2.2:0
Ing Lbl       : 524283U
Egr Lbl       : --
Egr Int/LspId : --
EgrNextHop    : --
Egr. Flags    : None                Ing. Flags : None
=====
No. of Generic IPv4 P2MP Bindings: 1
=====
```

PE-2 has programmed a swap entry for the label it sent to its upstream node PE-1 with the next-hop address and the label it received from the downstream node, as follows:

```
[/]
A:admin@PE-2# show router ldp bindings active p2mp opaque-type generic ipv4

=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP Generic IPv4 P2MP Bindings (Active)
=====
P2MP-Id      Interface
RootAddr     Op
IngLbl       EgrLbl
EgrNH        EgrIf/LspId
-----
5000          Unknw
192.0.2.1    Swap
524283       524283
```

```

192.168.23.2                               1/1/c1/1
-----
No. of Generic IPv4 P2MP Active Bindings: 1
=====

[/]
A:admin@PE-2# show router ldp bindings p2mp opaque-type generic ipv4 detail

=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP Generic IPv4 P2MP Bindings
=====
-----
P2MP Type      : 1                P2MP-Id       : 5000
Root-Addr      : 192.0.2.1
-----
Peer           : 192.0.2.1:0
Ing Lbl        : 524283U
Egr Lbl        : --
Egr Int/LspId  : --
EgrNextHop     : --
Egr. Flags     : None              Ing. Flags    : None
-----
P2MP Type      : 1                P2MP-Id       : 5000
Root-Addr      : 192.0.2.1
-----
Peer           : 192.0.2.3:0
Ing Lbl        : --
Egr Lbl        : 524283
Egr Int/LspId  : 1/1/c1/1
EgrNextHop     : 192.168.23.2
Egr. Flags     : None              Ing. Flags    : None
Egr If Name    : int-PE-2-PE-3
Metric         : 1                 Mtu           : 8922
=====
No. of Generic IPv4 P2MP Bindings: 2
=====

```

The egress next hop is PE-3.

On the root node PE-1, there is MPLS multicast replication. One traffic stream goes via transit node PE-2 toward leaf node PE-3 and the other traffic stream goes directly toward leaf node PE-4. There are two push entries with the corresponding next-hop address, as follows:

```

[/]
A:admin@PE-1# show router ldp bindings active p2mp opaque-type generic ipv4

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====

```

```

Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP Generic IPv4 P2MP Bindings (Active)
=====
P2MP-Id      Interface
RootAddr     Op
IngLbl       EgrLbl
EgrNH        EgrIf/LspId
-----
5000         73728
192.0.2.1    Push
--          524283
192.168.12.2 1/1/c1/1

5000         73728
192.0.2.1    Push
--          524283
192.168.14.2 1/1/c2/1
-----
No. of Generic IPv4 P2MP Active Bindings: 2
=====

```

```

[/]
A:admin@PE-1# show router ldp bindings p2mp opaque-type generic ipv4 detail
=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP Generic IPv4 P2MP Bindings
=====
P2MP Type      : 1          P2MP-Id      : 5000
Root-Addr      : 192.0.2.1
-----
Peer           : 192.0.2.2:0
Ing Lbl        : --
Egr Lbl        : 524283
Egr Int/LspId  : 1/1/c1/1
EgrNextHop     : 192.168.12.2
Egr. Flags     : None      Ing. Flags   : None
Egr If Name    : int-PE-1-PE-2
Metric         : 1          Mtu          : 8922
-----
P2MP Type      : 1          P2MP-Id      : 5000
Root-Addr      : 192.0.2.1
-----
Peer           : 192.0.2.4:0

```

```

Ing Lbl      :  --
Egr Lbl      : 524283
Egr Int/LspId : 1/1/c2/1
EgrNextHop   : 192.168.14.2
Egr. Flags   : None           Ing. Flags : None
Egr If Name   : int-PE-1-PE-4
Metric        : 1             Mtu         : 8922
=====
No. of Generic IPv4 P2MP Bindings: 2
=====

```

Tools command

The following tools command can be launched on any of the nodes in the P2MP LSP.

For the ingress node PE-1, where one branch goes to transit node PE-2 (192.0.2.2) and another branch to leaf node PE-4 (192.0.2.4), the output is as follows:

```

[/]
A:admin@PE-1# tools dump router ldp fec p2mp-id 5000 root 192.0.2.1
P2MP: root: 192.0.2.1, T: 1, L: 4, TunnelId: 5000
  Create Time   : 09/06/23 21:43:41.278 (elapsed: 0d 00:11:16)
  Last Mod. Time: 09/06/23 21:43:51.708 (elapsed: 0d 00:11:06)
  FEC Flags     : Push Mttm
  FEC typedFlags: none
  TunlIfId     : 73728 (OperState : up)
  mttmTunnAttr : None
  LSP ID       : 65540      LSP ID Acct. : 4
  useMcastRteTbl: No      mttmRd       : 0x0
  isIngressMttm : Yes    HasLeaf     : No
  isIngrItermdte: No      CanProgIngress: Yes
  InPhopFrr    : No      InAsbrFrr    : No
  isStitchedUpr : No      isLwrMbbCplte : No
  stitchRteType : Rtm     detctBkpAsbrLp: No
  bkpLoop      : No      bkpAsbrLoop  : No
  sysIpAddr    : 0.0.0.0
  RslvdPhop(p) : 0.0.0.0:0 (seqNum 0 isBgp No isTtm No)
  RslvdPhop(b) : 0.0.0.0:0 (seqNum 0 isBgp No isTtm No)
  pri Upstream : None
  mbb Upstream : None
  bkp Upstream : None
  AdvInLabel(p) : 0
  AdvInLabel(b) : 0
  RslvdAsbr(p) : 0.0.0.0 (seqNum 0 recursive: No vpn: No)
  RslvdAsbr(b) : 0.0.0.0 (seqNum 0 recursive: No vpn: No)
  Num Resolved Nhops : 2
  Num MBB Req. Nhops : 0
  Num Programmed Nhops : 2
  Num MBB Req. Lwr Fecs : 0
  Programmed Nhops[01] : 192.0.2.2:0, OutLabel 524283
  Programmed Nhops[02] : 192.0.2.4:0, OutLabel 524283
  Metric        : 1             Mtu         : 8922

  Num of Peers : 2

  FEC Peer: 192.0.2.2:0
  Peer Flags: MPush (0x800)
  ModTime   : 09/06/23 21:43:50.310 (elapsed.: 0d 00:11:07)

  ->Num Egress Labels:
  -> (Label: 524283   Status: UsePush)

```



```

Flow Label Tx: no, Rx: no

<-Num Ingress Labels:
None

<Resolved as Next Hop>
Next Hop Info :
  metric: 1 mtu: 8922
  [01]: Next Hop: 192.168.12.2   Interface: 2 Inner Label: 0

FEC Peer: 192.0.2.4:0
Peer Flags: MPush (0x800)
ModTime   : 09/06/23 21:43:51.702 (elapsed.: 0d 00:11:06)

->Num Egress Labels:
-> (Label: 524283   Status: UsePush)
Flow Label Tx: no, Rx: no

<-Num Ingress Labels:
None

<Resolved as Next Hop>
Next Hop Info :
  metric: 1 mtu: 8922
  [01]: Next Hop: 192.168.14.2   Interface: 3 Inner Label: 0

```

The labels that are pushed at PE-1 are 524283 for traffic to PE-2 and 524283 for traffic to PE-4.

On transit node PE-2, the incoming label 524283 is swapped to outgoing label 524283 toward PE-3, as follows:

```

[/]
A:admin@PE-2# tools dump router ldp fec p2mp-id 5000 root 192.0.2.1
P2MP: root: 192.0.2.1, T: 1, L: 4, TunnelId: 5000
  Create Time   : 09/06/23 21:43:50.304 (elapsed: 0d 00:11:09)
  Last Mod. Time: 09/06/23 21:43:50.304 (elapsed: 0d 00:11:09)
  FEC Flags     : Swap
  FEC typedFlags: none
  TunlIfId     : 0           (OperState : dn)
  mttmTunnAttr : None
  LSP ID       : 0           LSP ID Acct. : 0
  useMcastRteTbl: No         mttmRd      : 0x0
  isIngressMttm : No         HasLeaf      : No
  isIngrItermdte: No         CanProgIngress: No
  InPhopFrr    : No         InAsbrFrr   : No
  isStitchedUprr : No         isLwrMbbCplte: No
  stitchRteType : None       detctBkpAsbrLp: No
  bkpLoop      : No         bkpAsbrLoop : No
  sysIpAddr    : 0.0.0.0
  RslvdPhop(p) : 192.0.2.1:0 (seqNum 2 isBgp No isTtm No)
  RslvdPhop(b) : 0.0.0.0:0 (seqNum 0 isBgp No isTtm No)
  pri Upstream  : 192.0.2.1:0, AdvLabel 524283
  mbb Upstream  : None
  bkp Upstream  : None
  AdvInLabel(p) : 524283
  AdvInLabel(b) : 0
  RslvdAsbr(p)  : 0.0.0.0 (seqNum 0 recursive: No vpn: No)
  RslvdAsbr(b)  : 0.0.0.0 (seqNum 0 recursive: No vpn: No)
  PrgInLabel(b) : 1
  Num Resolved  Nhops   : 1
  Num MBB Req.  Nhops   : 0
  Num Programmed Nhops   : 1

```

```
Num MBB Req. Lwr Fecs : 0
  Programmed Nhop[01] : 192.0.2.3:0, OutLabel 524283
Metric      : 1      Mtu      : 8922

Num of Peers : 2

FEC Peer: 192.0.2.1:0
Peer Flags: none (0x0)
ModTime   : 09/06/23 21:43:50.309 (elapsed.: 0d 00:11:09)

->Num Egress Labels:
  None

<-Num Ingress Labels:
  <- (Label: 524283   Status: UseSwap)
  Rej Status: OK
  Flow Label Tx: no, Rx: no
  Flow Label Tx Sent: no, Rx Sent: no

<Resolved as CUR Upstream>

FEC Peer: 192.0.2.3:0
Peer Flags: MSwap (0x1000)
ModTime   : 09/06/23 21:43:50.301 (elapsed.: 0d 00:11:09)

->Num Egress Labels:
  -> (Label: 524283   Status: UseSwap)
  Flow Label Tx: no, Rx: no

<-Num Ingress Labels:
  None

<Resolved as Next Hop>
Next Hop Info :
  metric: 1 mtu: 8922
  [01]: Next Hop: 192.168.23.2   Interface: 3 Inner Label: 0
```

On leaf node PE-3, the incoming label from PE-2 (524283) is popped. There is no next hop.

```
[/]
A:admin@PE-3# tools dump router ldp fec p2mp-id 5000 root 192.0.2.1
P2MP: root: 192.0.2.1, T: 1, L: 4, TunnelId: 5000
  Create Time   : 09/06/23 21:43:50.305 (elapsed: 0d 00:11:11)
  Last Mod. Time: 09/06/23 21:43:50.305 (elapsed: 0d 00:11:11)
  FEC Flags     : Pop Mtm
  FEC typedFlags: none
  TunlIfId     : 73728 (OperState : up)
  mttmTunnAttr : None
  LSP ID       : 0      LSP ID Acct. : 0
  useMcastRteTbl: No    mttmRd      : 0x0
  isIngressMtm : No    HasLeaf      : Yes
  isIngrItermdte: No   CanProgIngress: No
  InPhopFrr    : No    InAsbrFrr   : No
  isStitchedUp : No    isLwrMbbCplte : No
  stitchRteType : None  detctBkpAsbrLp: No
  bkpLoop      : No    bkpAsbrLoop : No
  sysIpAddr    : 0.0.0.0
  RslvdPhop(p) : 192.0.2.2:0 (seqNum 2 isBgp No isTtm No)
  RslvdPhop(b) : 0.0.0.0:0 (seqNum 0 isBgp No isTtm No)
  pri Upstream  : 192.0.2.2:0, AdvLabel 524283
  mbb Upstream  : None
  bkp Upstream  : None
  AdvInLabel(p) : 524283
```

```

AdvInLabel(b) : 0
RslvdAsbr(p) : 0.0.0.0 (seqNum 0 recursive: No vpn: No)
RslvdAsbr(b) : 0.0.0.0 (seqNum 0 recursive: No vpn: No)
PrgInLabel(b) : 1
Num Resolved Nhops : 1
Num MBB Req. Nhops : 0
Num Programmed Nhops : 1
Num MBB Req. Lwr Fecs : 0
  Programmed Nhop[01] : 0.0.0.0:0, OutLabel 0 (Leaf)
Metric : 0 Mtu : 0

Num of Peers : 1

FEC Peer: 192.0.2.2:0
Peer Flags: none (0x0)
ModTime : 09/06/23 21:43:50.311 (elapsed.: 0d 00:11:11)

->Num Egress Labels:
None

<-Num Ingress Labels:
<- (Label: 524283 Status: UsePop)
Rej Status: OK
Flow Label Tx: no, Rx: no
Flow Label Tx Sent: no, Rx Sent: no

<Resolved as CUR Upstream>

```

The output for leaf node PE-4 is similar.

Debug commands

Debugging was enabled on the nodes when LDP was configured. To distinguish which messages are being logged for a debug command, the debug configuration is different for the nodes, as follows:

```

# on PE-2:
debug {
  router "Base" {
    ldp {
      peer 192.0.2.1 {
        event {
          bindings { }
        }
      }
    }
  }
}

# on PE-3::
debug {
  router "Base" {
    ldp {
      peer 192.0.2.2 {
        packet {
          label {
            detail true
          }
        }
      }
    }
  }
}

# on PE-4:
debug {
  router "Base" {
    ldp {
      peer 192.0.2.1 {
        packet {
          init {

```

```
        detail true  
    }
```

The following LDP messages are logged. The first two messages correspond to the label mapping messages to establish LDP bindings. The following message is sent from transit node PE-2 to root node PE-1.

```
# on PE-2:  
debug {  
  router "Base" {  
    ldp {  
      peer 192.0.2.1 {  
        event {  
          bindings { }  
        }  
      }  
    }  
  }  
}
```

```
# on PE-2:  
7 2023/09/06 23:43:50.308 CEST MINOR: DEBUG #2001 Base LDP  
"LDP: Binding  
Sending Label mapping label 524283 for P2MP: root = 192.0.2.1, T: 1, L: 4, TunnelId: 5000  
to peer 192.0.2.1:0."
```

The following LDP message is sent by the leaf node PE-3 to the transit node PE-2.

```
# on PE-3::  
debug {  
  router "Base" {  
    ldp {  
      peer 192.0.2.2 {  
        packet {  
          label {  
            detail true  
          }  
        }  
      }  
    }  
  }  
}
```

```
# on PE-3:  
7 2023/09/06 23:43:50.307 CEST MINOR: DEBUG #2001 Base LDP  
"LDP: LDP  
Send Label Mapping packet (msgId 30) to 192.0.2.2:0  
Protocol version = 1  
Label 524283 advertised for the following FECs  
P2MP: root = 192.0.2.1, T: 1, L: 4, TunnelId: 5000  
"
```

The following message shows the negotiation of capabilities when LDP bindings are initialized.

```
# on PE-4:  
debug {  
  router "Base" {  
    ldp {  
      peer 192.0.2.1 {  
        packet {  
          init {  
            detail true  
          }  
        }  
      }  
    }  
  }  
}
```

```
# on PE-4:  
1 2023/09/06 23:41:22.613 CEST MINOR: DEBUG #2001 Base LDP  
"LDP: LDP  
Send Initialization packet (msgId 2) to 192.0.2.1:0"
```

```
Protocol version = 1
Keepalive Timeout = 30 Label Advertisement = downstreamUnsolicited
Loop Detection = Off PathVector Limit = 0 Max Pdu = 4096
P2MP Capability = yes
MP MBB Capability = yes
Overload Capability = yes
Dynamic Capability = yes
Unrecognized Notification Capability = yes
"
```

Configure multicast LDP and verify traffic

Configure PIM and IGMP on the root and leaf nodes

PIM needs to be enabled on the root node on the interface toward the multicast source S-1, as follows:

```
# on PE-1:
configure {
  router "Base" {
    pim {
      interface "int-PE-1-S-1" { }
    }
  }
}
```

On the leaf nodes, PIM needs to be enabled (no shutdown), but no interfaces need to be assigned.

The IGMP configuration for root node PE-1 is needed to forward the incoming traffic for multicast group 232.1.1.1 from source 172.16.11.2 to the tunnel-interface. If IGMP is not configured, the incoming traffic on the interface toward the multicast source is dropped, because no outgoing interface is defined.

```
# on PE-1:
configure {
  router "Base" {
    igmp {
      tunnel-interface {
        ldp-p2mp-root 5000 sender-address 192.0.2.1 {
          static {
            group 232.1.1.1 {
              source 172.16.11.2 { }
            }
          }
        }
      }
    }
  }
}
```

The IGMP configuration for leaf node PE-3 is as follows:

```
# on PE-3:
configure {
  router "Base" {
    igmp {
      interface "int-PE-3-H-3" {
        static {
          group 232.1.1.1 {
            source 172.16.11.2 { }
          }
        }
      }
    }
  }
}
```

The incoming traffic from the tunnel interface is forwarded to the outgoing interface toward the receiving multicast host H-3.

The IGMP configuration for leaf node PE-4 is similar.

At this point, the IGMP/PIM configuration on the root node is complete. This can be verified, as follows:

```
[/]
A:admin@PE-1# show router pim group

=====
Legend:  A = Active   S = Standby
=====
PIM Groups ipv4
=====
Group Address          Type          Spt Bit  Inc Intf      No.Oifs
  Source Address      RP           State    Inc Intf(S)
-----
232.1.1.1              (S,G)                int-PE-1-S-1  1
  172.16.11.2
-----
Groups : 1
=====
```

```
[/]
A:admin@PE-1# show router pim group detail

=====
PIM Source Group ipv4
=====
Group Address          : 232.1.1.1
Source Address         : 172.16.11.2
RP Address             : 0
Advt Router            : 192.0.2.1
Flags                  :
Mode                   : sparse
MRIB Next Hop          : 172.16.11.2
MRIB Src Flags         : direct
Keepalive Timer        : Not Running
Up Time                : 0d 00:03:32      Resolved By          : rtable-u

Up JP State            : Joined           Up JP Expiry          : 0d 00:00:00
Up JP Rpt              : Not Joined StarG  Up JP Rpt Override   : 0d 00:00:00

Register State        : No Info
Reg From Anycast RP: No

Rpf Neighbor           : 172.16.11.2
Incoming Intf       : int-PE-1-S-1
Outgoing Intf List : mpls-if-73728

Curr Fwding Rate      : 9751.560 kbps
Forwarded Packets     : 17930           Discarded Packets    : 0
Forwarded Octets      : 26572260        RPF Mismatches       : 0
Spt threshold         : 0 kbps           ECMP opt threshold   : 7
Admin bandwidth       : 1 kbps
-----
Groups : 1
=====
```

The incoming interface is the interface facing the multicast source S-1. The outgoing interface is a reference to the tunnel interface. The name for the outgoing interface (mpls-if-73728) contains the tunnel interface index 73728 as in previous CLI output. The multicast source S-1 is already sending traffic, but the receivers cannot receive it yet.

The configuration on the leaf nodes is still incomplete. A multicast policy needs to be configured and applied first.

Configure and apply multicast policy on leaf nodes

The leaf nodes need to get multicast traffic off the LDP P2MP LSP. Therefore, a multicast policy needs to be created and applied, as follows:

```
# on PE-3 and PE-4:
configure {
  multicast-management {
    multicast-info-policy "p2mp-pol" {
      bundle "bundle1" {
        primary-tunnel-interface {
          ldp-p2mp 5000
          sender 192.0.2.1
        }
        channel start 232.1.1.1 end 232.1.1.1 { }
      }
    }
  }
}

configure {
  router "Base" {
    multicast-info-policy "p2mp-pol"
  }
}
```

Verify multicast traffic on leaf nodes

Verify the multicast traffic, as follows:

```
[/]
A:admin@PE-3# show router pim group

=====
Legend:  A = Active   S = Standby
=====
PIM Groups ipv4
=====
Group Address          Type          Spt Bit   Inc Intf   No.Oifs
  Source Address      RP           State     Inc Intf(S)
-----
232.1.1.1              (S,G)                mpls-if-73728  1
  172.16.11.2
-----
Groups : 1
=====
```

The multicast source S-1 sends a multicast stream with group address 232.1.1.1. The multicast traffic is received by the leaf nodes, which can be verified as follows:

```
[/]
A:admin@PE-3# show router pim group detail

=====
PIM Source Group ipv4
=====
Group Address       : 232.1.1.1
Source Address      : 172.16.11.2
RP Address          : 0
Advt Router        :
Flags               :
Mode                : sparse
Type                : (S,G)
```

```

MRIB Next Hop      :
MRIB Src Flags     : remote
Keepalive Timer    : Not Running
Up Time           : 0d 00:03:26      Resolved By       : unresolved

Up JP State        : Joined           Up JP Expiry      : 0d 00:00:05
Up JP Rpt          : Not Joined StarG Up JP Rpt Override : 0d 00:00:00

Register State     : No Info
Reg From Anycast RP: No

Rpf Neighbor       :
Incoming Intf    : mpls-if-73728
Outgoing Intf List : int-PE-3-H-3

Curr Fwding Rate   : 9751.560 kbps
Forwarded Packets  : 19246           Discarded Packets : 0
Forwarded Octets   : 28522572       RPF Mismatches    : 0
Spt threshold      : 0 kbps          ECMP opt threshold : 7
Admin bandwidth    : 1 kbps
-----
Groups : 1
=====

```

The incoming interface is from the tunnel interface, whereas the outgoing interface is toward the receiving multicast host H-3.

mLDP fast upstream switchover

mLDP fast upstream switchover allows a downstream node of an mLDP FEC to perform a fast switchover and source the traffic from another upstream node. This switchover is necessary when IGP and LDP are converging after a failure of the upstream LSR, which is the primary next hop of the root LSR for the P2MP FEC. There is traffic duplication toward the node that has the upstream alternate backup (in this case to PE-3), but only one stream is accepted. The multicast stream is sent to the primary next hop as well as to the loopfree alternate backup. As long as there is no failure, the primary next hop accepts the traffic and forwards it. The backup rejects the traffic. When a failure occurs and the primary LDP session goes down, the backup starts accepting packets.

mLDP fast upstream switchover provides an upstream Fast Reroute (FRR) node-protection capability for the mLDP FEC packets. This multicast upstream FRR node protection is at the expense of traffic duplication from two different upstream nodes into the node that performs the fast upstream switchover. This feature is described in *draft-pdutta-mpls-mlbp-up-redundancy*.

Multicast upstream FRR can be configured for mLDP, as follows:

```

# on PE-1:
configure {
  router "Base" {
    ospf 0 {
      loopfree-alternate { }
    }
    ldp {
      mcast-upstream-frr true
    }
  }
}

```


This configuration can be repeated on some or all of the nodes. In this example, it is configured on all nodes. FRR for unicast can be configured in combination with this, but that is not required. FRR for unicast can be enabled as follows:

```
# on all nodes:
configure {
  routing-options {
    ip-fast-reroute true
  }
  router "Base" {
    ldp {
      fast-reroute { }
    }
  }
}
```

In this example, it is assumed that unicast IP and unicast LDP prefixes do not need to be protected. Therefore, unicast FRR remains disabled.

FRR can be verified as disabled for unicast (FRR) and enabled for multicast (Mcast Upstream FRR), as follows:

```
[/]
A:admin@PE-1# show router ldp status

=====
LDP Status for IPv4 LSR ID 192.0.2.1
      IPv6 LSR ID ::
=====
---snip---
Admin State           : Up
IPv4 Oper State       : Up
IPv6 Oper State       : Down
---snip---
FRR                  : Disabled
Mcast Upst ASBR FRR: Disabled
MP MBB Time           : 3
---snip---
-----
Capabilities
-----
Dynamic               : Enabled
IPv4 Prefix Fec       : Enabled
Service Fec128        : Enabled
MP MBB                : Enabled
Unrecognized Notif*   : Enabled
P2MP                  : Enabled
IPv6 Prefix Fec       : Enabled
Service Fec129        : Enabled
Overload              : Enabled
=====
* indicates that the corresponding row element may have been truncated.
```

Of the three nodes in the example topology that have upstream nodes, only PE-3 has an upstream alternate for FRR. PE-4 becomes a transit node for traffic destined for PE-3 (but PE-3 drops it, until the primary LDP session fails). PE-3 sends a label mapping message to PE-4 for label 524282, as in the following trace message.

```
# on PE-3:
8 2023/09/07 00:00:27.538 CEST MINOR: DEBUG #2001 Base LDP
"LDP: LDP
Send Label Mapping packet (msgId 140) to 192.0.2.4:0
Protocol version = 1
Label 524282 advertised for the following FECs
P2MP: root = 192.0.2.1, T: 1, L: 4, TunnelId: 5000
MP Status MBB = REQ
```

"

PE-4 has an additional LDP P2MP binding where the label is swapped, as follows:

```
[/]
A:admin@PE-4# show router ldp bindings active p2mp opaque-type generic ipv4

=====
LDP Bindings (IPv4 LSR ID 192.0.2.4)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP Generic IPv4 P2MP Bindings (Active)
=====
P2MP-Id          Interface
RootAddr         0p
IngLbl           EgrLbl
EgrNH            EgrIf/LspId
-----
5000             73728
192.0.2.1       Pop
524283          --
--              --

5000             73728
192.0.2.1       Swap
524283          524282
192.168.34.1   1/1/c2/1
-----
No. of Generic IPv4 P2MP Active Bindings: 2
=====
```

PE-3 has an additional entry for the FRR backup that is available (BU - Alternate for Fast Re-Route). PE-3 gets duplicated traffic, but rejects all traffic from PE-4 and only accept traffic from PE-2 as long as there is no failover.

```
[/]
A:admin@PE-3# show router ldp bindings active p2mp opaque-type generic ipv4

=====
LDP Bindings (IPv4 LSR ID 192.0.2.3)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP Generic IPv4 P2MP Bindings (Active)
=====
P2MP-Id          Interface
```

RootAddr	Op
IngLbl	EgrLbl
EgrNH	EgrIf/LspId
-----	-----
5000	73728
192.0.2.1	Pop
524283	--
--	--
5000	73728
192.0.2.1	Pop
524282BU	--
--	--
-----	-----
No. of Generic IPv4 P2MP Active Bindings: 2	
=====	

Because LoopFree Alternate (LFA) and ECMP are mutually exclusive, LFA is only useful when ECMP is disabled. When both are enabled, ECMP has preference.

mLDP fast upstream switchover relies on the fast detection of loss of the LDP session to the upstream peer to which the primary Ingress Label Map (ILM) label had been advertised. As a result, Nokia recommends to perform the following:

1. Enable Bidirectional Forwarding Detection (BFD) on all LDP interfaces to upstream LSR nodes. When BFD detects the loss of the last adjacency to the upstream LSR, BFD brings down the LDP session immediately. The backup ILM is activated.
2. If there is a concurrent T-LDP adjacency to the same LSR node, enable BFD on the T-LDP peer as well as on the interface.
3. Enable the **ldp-sync-timer** option on all interfaces to the upstream LSR nodes.

If the LDP session for the primary ILM to the upstream LSR goes down for any other reason than a failure of the interface or of the upstream LSR, routing and LDP go out of sync. The backup ILM remains activated until the Interior Gateway Protocol (IGP) seeks the next Shortest Path First (SPF). By enabling the **ldp-sync-timer**, this process is accelerated because the advertised link metric gets the maximum value as soon as the LDP session goes down. This triggers the IGP to calculate an SPF route. See chapter [LDP-IGP Synchronization](#).

The FRR configuration can be removed, as follows:

```
# on all nodes:
configure {
  router "Base" {
    ldp {
      delete mcast-upstream-frr
    }
    ospf 0 {
      delete loopfree-alternate
    }
  }
}
```

Multipoint make-before-break (MP MBB)

Multipoint MBB is performed when the best path to the root changes, but the existing path can still be used, such as when a link comes up or when the routing metric changes. The goal of MBB is to establish a new P2MP LSP before the old P2MP is removed, so as to avoid traffic loss.

Leaf or transit nodes must allocate a new label and program the ILM with a duplicate set of existing Next-Hop Label Forwarding Entries (NHLFEs) toward the upstream nodes. This may lead to traffic duplication for a short period of time.

Multipoint MBB is enabled by default, as follows:

```
[/]
A:admin@PE-3# show router ldp status

=====
LDP Status for IPv4 LSR ID 192.0.2.3
                IPv6 LSR ID ::
=====
---snip---
Admin State      : Up
IPv4 Oper State  : Up                IPv6 Oper State      : Down
---snip---
MP MBB Time      : 3
---snip---
-----
Capabilities
-----
Dynamic          : Enabled            P2MP                 : Enabled
IPv4 Prefix Fec  : Enabled            IPv6 Prefix Fec      : Enabled
Service Fec128   : Enabled            Service Fec129       : Enabled
MP MBB          : Enabled           Overload              : Enabled
Unrecognized Notif*: Enabled
=====
* indicates that the corresponding row element may have been truncated.
```

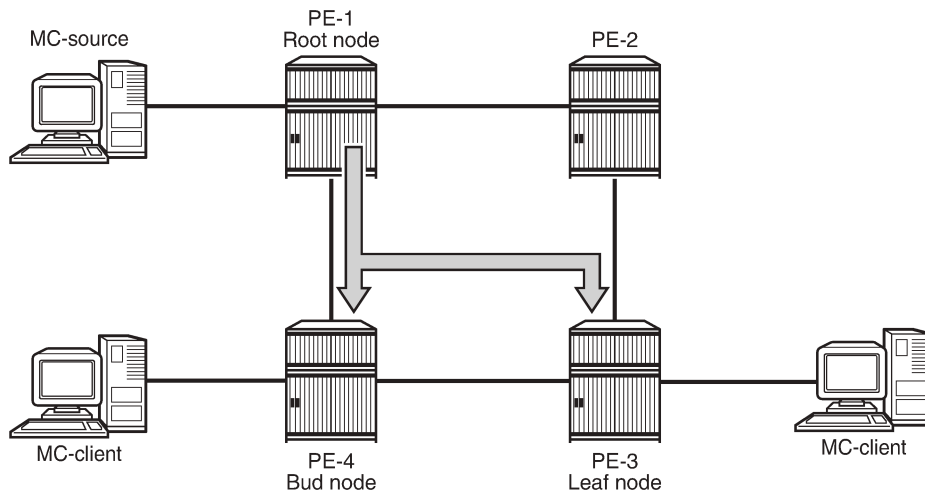
When the metric is increased on the interface (int-PE-3-PE-2) toward the active upstream node, PE-3 sends out an OSPF link status update. Traffic still arrives at PE-3 using the original P2MP LSP. The MBB P2MP LSP is set up.

PE-3 sends a label mapping message toward PE-4, including an MP status TLV carrying an MBB status code indicating that MBB procedures apply to the LSP. PE-4 sends an LDP notification toward PE-3, including an MP status TLV indicating that PE-4 has a state for the existing P2MP LSP.

PE-3 sends an LDP withdrawal message to PE-2. PE-2 replies with an LDP release message.

The multicast traffic arrives at PE-3 using the new LDP P2MP LSP. This way, PE-4 becomes a bud node, and PE-2 is not used for transit anymore; see [Figure 242: New LDP P2MP LSP after metric change](#).

Figure 242: New LDP P2MP LSP after metric change



25516

Originally, leaf node PE-3 preferred the route via PE-2 toward root node PE-1, as follows:

```
[/]
A:admin@PE-3# show router route-table 192.0.2.1

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type  Proto  Age           Pref
  Next Hop[Interface Name]                       Metric
-----
192.0.2.1/32                                     Remote OSPF    00h23m14s    10
  192.168.23.1                                  2
-----
No. of Routes: 1
---snip---
```

Consequently, the label map messages were originally sent to PE-2, not to PE-4. PE-2 is the transit node for traffic destined for PE-3, as follows:

```
[/]
A:admin@PE-2# show router ldp bindings active p2mp opaque-type generic ipv4

=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP Generic IPv4 P2MP Bindings (Active)
=====
```

```

P2MP-Id          Interface
RootAddr         Op
IngLbl           EgrLbl
EgrNH            EgrIf/LspId
-----
5000             Unknw
192.0.2.1        Swap
524283           524283
192.168.23.2   1/1/c1/1
-----
No. of Generic IPv4 P2MP Active Bindings: 1
=====

```

The metric is changed on the interface between PE-3 and PE-2, as follows:

```

# on PE-3:
configure {
  router "Base" {
    ospf 0 {
      area 0.0.0.0 {
        interface "int-PE-3-PE-2" {
          metric 1000
        }
      }
    }
  }
}

```

The preferred route from leaf node PE-3 to root node PE-1 is now via PE-4, as follows:

```

[/]
A:admin@PE-3# show router route-table 192.0.2.1
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type   Proto   Age           Pref
  Next Hop[Interface Name]                               Metric
-----
192.0.2.1/32                Remote OSPF    00h00m23s    10
  192.168.34.2                               2
-----
No. of Routes: 1
---snip---
=====

```

The leaf node PE-3 prefers to set up a path from PE-4 instead of from PE-2. PE-3 sends label mapping messages to PE-4. The old P2MP LSP is used until the new P2MP LSP is set up. There is no traffic interruption.

PE-3 sends a label withdrawal message to PE-2 and PE-2 is no longer a transit node, as follows:

```

[/]
A:admin@PE-2# show router ldp bindings active p2mp opaque-type generic ipv4
=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
  (IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,

```

```

=====
      BA - ASBR Backup FEC
=====
LDP Generic IPv4 P2MP Bindings (Active)
=====
P2MP-Id          Interface
RootAddr         Op
IngLbl           EgrLbl
EgrNH            EgrIf/LspId
-----
No Matching Entries Found
=====

```

PE-4 is the transit node for traffic to PE-3, and also has a local multicast client H-4, so it is a bud node, as follows:

```

[/]
A:admin@PE-4# show router ldp bindings active p2mp opaque-type generic ipv4
=====
LDP Bindings (IPv4 LSR ID 192.0.2.4)
      (IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP Generic IPv4 P2MP Bindings (Active)
=====
P2MP-Id          Interface
RootAddr         Op
IngLbl           EgrLbl
EgrNH            EgrIf/LspId
-----
5000             73728
192.0.2.1        Pop
524283           --
--              --

5000             73728
192.0.2.1        Swap
524283           524282
192.168.34.1    1/1/c2/1

-----
No. of Generic IPv4 P2MP Active Bindings: 2
=====

```

There is no traffic multiplication at the root node PE-1. All traffic goes to PE-4, as follows:

```

[/]
A:admin@PE-1# show router ldp bindings active p2mp opaque-type generic ipv4
=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
      (IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn

```

```

WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP Generic IPv4 P2MP Bindings (Active)
=====
P2MP-Id          Interface
RootAddr         0p
IngLbl           EgrLbl
EgrNH            EgrIf/LspId
-----
5000              73728
192.0.2.1        Push
--              524283
192.168.14.2    1/1/c2/1
-----
No. of Generic IPv4 P2MP Active Bindings: 1
=====

```

The switchover to this new P2MP LSP occurred without traffic loss.

The following debugging was enabled before the metric change:

```

# on PE-3:
debug {
  router "Base" {
    ldp {
      peer 192.0.2.2 {
        packet {
          label {
            detail true
          }
          init {
            detail true
          }
        }
      }
      peer 192.0.2.4 {
        packet {
          label {
            detail true
          }
          init {
            detail true
          }
        }
      }
    }
  }
}

```

The first trace message shows that label 524282 is advertised to PE-4. MBB is requested, as follows:

```

# on PE-3:
12 2023/09/07 00:02:53.483 CEST MINOR: DEBUG #2001 Base LDP
"LDP: LDP
Send Label Mapping packet (msgId 157) to 192.0.2.4:0
Protocol version = 1
Label 524282 advertised for the following FECs
P2MP: root = 192.0.2.1, T: 1, L: 4, TunnelId: 5000
MP Status MBB = REQ

```


"

The next message is a notification from PE-4 confirming that there is no fatal error and MBB can be applied, as follows:

```
# on PE-3:
13 2023/09/07 00:02:53.485 CEST MINOR: DEBUG #2001 Base LDP
"LDP: LDP
Recv Notification packet (msgId 157) from 192.0.2.4:0
Protocol version = 1
Status Code = MPStatus (0x00000040) Non-fatal
Causing message Id = 0
Causing message type = NULL
P2MP: root = 192.0.2.1, T: 1, L: 4, TunnelId: 5000
MP Status MBB = ACK
"
```

The following message is a label withdraw message for label 524283 sent to PE-2:

```
# on PE-3:
14 2023/09/07 00:02:53.485 CEST MINOR: DEBUG #2001 Base LDP
"LDP: LDP
Send Label Withdraw packet (msgId 158) to 192.0.2.2:0
Protocol version = 1
Label 524283 withdrawn for the following FECs
P2MP: root = 192.0.2.1, T: 1, L: 4, TunnelId: 5000
"
```

The last message is a label release message for label 524283 received from PE-2, as follows. This message is only sent after the new P2MP LSP is set up.

```
# on PE-3:
15 2023/09/07 00:02:53.486 CEST MINOR: DEBUG #2001 Base LDP
"LDP: LDP
Recv Label Release packet (msgId 157) from 192.0.2.2:0
Protocol version = 1
Label 524283 released for the following FECs
P2MP: root = 192.0.2.1, T: 1, L: 4, TunnelId: 5000
"
```

Conclusion

Multicast LDP provides extensions to the LDP protocol for the setup of P2MP and MP2MP LSPs in MPLS networks. mLDP is simple to configure compared to RSVP. FRR and MBB are supported for mLDP.

Path MTU Discovery

This chapter provides information about Path MTU discovery.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 14.0.R7, but the CLI in the current edition corresponds to SR OS Release 21.2.R1.

Overview

The Maximum Transmission Unit (MTU) is the largest packet size (in bytes) that a network can transmit. IP datagrams larger than the MTU are fragmented into smaller packets before being sent. [Table 15: MTU types](#) describes the MTU types that are supported in SR OS at both port and service level.

Table 15: MTU types

MTU type	Description
Port MTU	Maximum frame size on a physical wire
Service MTU	Maximum end-to-end frame size sent from the customer across an L2 VPN service
SDP path MTU	Maximum frame size of encapsulated packets sent over the SDP between service endpoints in IP/MPLS VPN
VC MTU	Maximum IP payload size that can be carried inside the tunnel. The VC MTU is derived from the service MTU and negotiated by T-LDP.
LSP path MTU	MTU value negotiated by RSVP path/resv messages
OSPF MTU	Maximum size of the OSPF packet
IP MTU	Used in L3 VPN services (IES or VPRN). Maximum IP packet size that L3 VPN customers can send across the provider network.

[Table 16: MTU values for Ethernet frames](#) lists the values for the MTU types for Ethernet frames. In SR OS, the MTU value never includes the Frame Check Sequence (FCS).

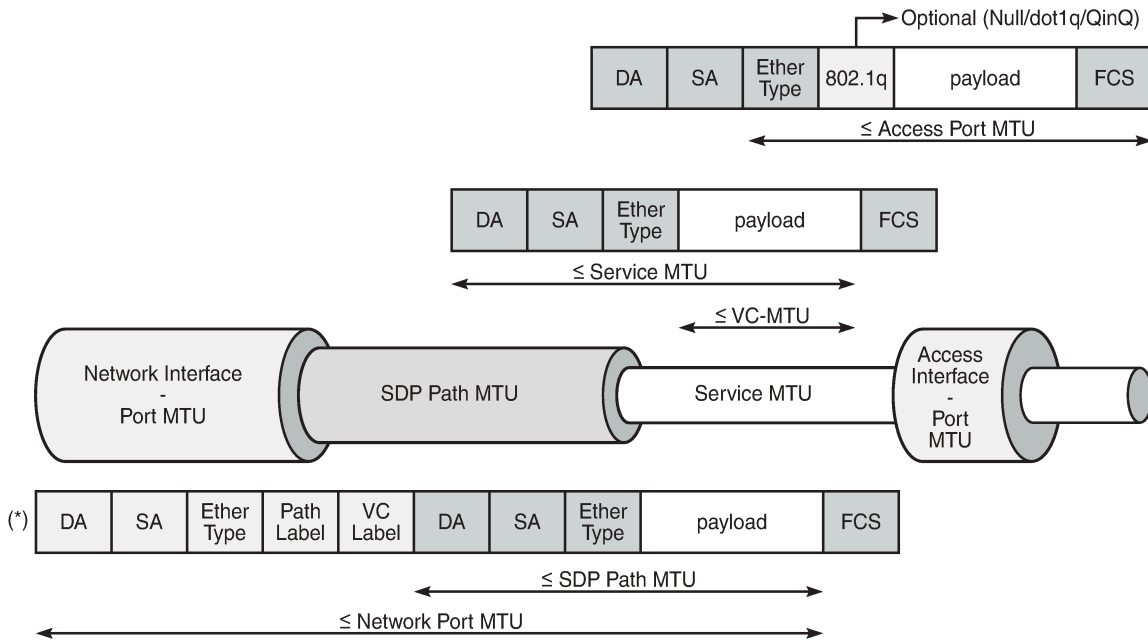
Table 16: MTU values for Ethernet frames

MTU type	Value
Access Port MTU	Configurable in the port context. Value should be greater than or equal to the sum of the service MTU and the port encapsulation overhead (0 for null, 4 for dot1q, 8 for QinQ).
Network Port MTU	Configurable in the port context. Value should be greater than or equal to the sum of the SDP path MTU, the MPLS labels (transport, service, hash (entropy), and OAM labels), and an Ethernet header (possibly with a VLAN tag for dot1q).
Service MTU	Configurable in the service context. Maximum payload (IP + Ethernet) that the service offers to the client. Only used in L2 services.
SDP path MTU	By default, not configured. Derived from the network port MTU. Should be, at a minimum, the value of the service MTU. Should be, at a maximum, the result of {network port MTU - 2 labels - Ethernet header}. However, the service will become operationally up when the SDP path MTU is higher. The SDP path MTU need not match on both sides of the SDP.
VC MTU	Not configurable. Derived from the service MTU and negotiated by T-LDP. The VC MTU value must match the other side. VC MTU = service MTU - 14 bytes (Ethernet header).
LSP path MTU	Derived from the port MTU of the network port
OSPF MTU	MTU negotiated by OSPF and derived from the port MTU or administratively set
IP MTU	Configurable in the L3 routing interfaces

The values of the first five MTU types listed in Table 2 are important in getting L2 services to an operational state of up. For L3 services, the IP MTU is used instead of the service MTU.

[Figure 243: L2 services MTUs for Ethernet frames](#) shows the MTUs used for Ethernet frames in an L2 service, such as an Epipe or a VPLS service.

Figure 243: L2 services MTUs for Ethernet frames



(*) Optionally additional 802.1q header (4 bytes) for dot1q

26371

The VC MTU contains the IP payload. The service MTU contains IP payload and Ethernet header. The SDP path MTU must be greater than or equal to the service MTU. Typically, the VLAN tags are stripped at the service ingress, unless VLAN range SAPs are defined and one VLAN tag is preserved. The physical port MTU on an Ethernet access interface needs to be set to at least 1514 for null encapsulation (1500 + 14 (Ethernet header)), at least 1518 for dot1q (1500 + 14 + 4 (dot1q)), and at least 1522 for QinQ (1500 + 14 + 4 + 4).

Figure 244: Minimum network port MTU for Ethernet frames in MPLS encapsulation shows the minimum physical MTU on network interfaces for a router that needs to support services offering a 1514 byte service payload over MPLS for Ethernet.

Figure 244: Minimum network port MTU for Ethernet frames in MPLS encapsulation

Overhead	Ethernet
Service Payload	1514
MPLS tag used as service ID	4
MPLS tag used for egress LSP	4
Optionally, more MPLS tags	(n*4)
Ethernet Header	14
Total	1536 (+ n*4)

Maximum 12 MPLS labels.
Optionally 1 VLAN tag for dot1q.

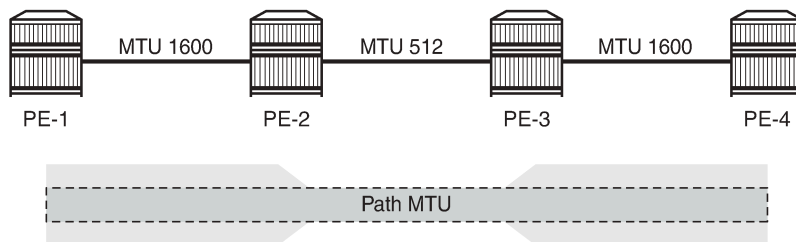
26372

The network port MTU must be at least the maximum service MTU to be supported plus the largest encapsulation type used. The SDP path MTU is at least equal to the service MTU, which is at a minimum 1514 for a service running on a typical Ethernet access interface. This is also valid when the access interface is dot1q or QinQ, because the VLAN tags are stripped at ingress and replaced by the appropriate VLAN tag at egress, unless VLAN range SAPs are defined, in which case one VLAN tag is preserved. The VC tag (service ID) adds a 4 byte service label, the MPLS path adds—at least—one 4 byte transport label, and the Ethernet header adds 14 bytes, for a total of at least 1536 for Ethernet encapsulation. For MPLS, the maximum label stack depth is 12.

The default behavior in SR OS is that the network port MTU is set to its maximum per MDA type, if the network port MTU is not explicitly configured. By default, the SDP path MTU is derived from the network port MTU. For example, when the network port is set to 1600, the SDP path MTU = 1600 (network port MTU) - 4 (MPLS service label) - 4 (MPLS path label) - 14 (Ethernet label) = 1578. However, the SDP path MTU is only accurate when the end-to-end path is considered and the lowest network port MTU in the path is taken.

Figure 245: Path MTU shows that the path MTU is determined by the lowest MTU along the path that the service needs to transit. When IP hosts transmit IP datagrams to each other, the path MTU is the largest size for which no fragmentation is required along the path.

Figure 245: Path MTU



26373

Path MTU discovery (PMTUD)

PMTUD is a technique for dynamically discovering the MTU size on the network path between two IP hosts, to maximize packet efficiency and avoid packet fragmentation. PMTUD is standardized in RFC 1191 and for IPv6 in RFC 1981.

PMTUD can be enabled in LDP and BGP in the following contexts:

```
[/]
A:admin@Dut-A# tree flat detail | match path-mtu-discovery
configure groups group <string> router <string> bgp group <string> path-mtu-discovery <boolean>
configure groups group <string> router <string> bgp neighbor <string | ipv4-address-with-zone |
ipv4-address | ipv6-address-linklocal-with-zone | ipv6-address | ipv6-address-with-zone> path-
mtu-discovery <boolean>
configure groups group <string> router <string> bgp path-mtu-discovery <boolean>
configure groups group <string> router <string> ldp tcp-session-parameters peer. For BGP, -
transport <string | ipv4-address | ipv6-address> path-mtu-discovery <boolean>
configure groups group <string> service vprn <string> bgp group <string> path-mtu-discovery
<boolean>
configure groups group <string> service vprn <string> bgp neighbor <string | ipv4-address-with-
zone | ipv4-address | ipv6-address-linklocal-with-zone | ipv6-address | ipv6-address-with-zone>
path-mtu-discovery <boolean>
configure groups group <string> service vprn <string> bgp path-mtu-discovery <boolean>
configure router <string> bgp group <string> path-mtu-discovery <boolean>
configure router <string> bgp neighbor <ipv4-address-with-zone | ipv4-address | ipv6-address-
linklocal-with-zone | ipv6-address | ipv6-address-with-zone> path-mtu-discovery <boolean>
configure router <string> bgp path-mtu-discovery <boolean>
configure router <string> ldp tcp-session-parameters peer-transport <ipv4-address | ipv6-
address> path-mtu-discovery <boolean>
configure service vprn <string> bgp group <string> path-mtu-discovery <boolean>
configure service vprn <string> bgp neighbor <ipv4-address-with-zone | ipv4-address | ipv6-
address-linklocal-with-zone | ipv6-address | ipv6-address-with-zone> path-mtu-discovery
<boolean>
configure service vprn <string> bgp path-mtu-discovery <boolean>
state system connections statistics tcp path-mtu-discovery-backoff <number>
```

PMTUD can be enabled in BGP at different levels: global, per group, or per neighbor. PMTUD can be enabled in BGP in the base router or in a VPRN. For LDP, PMTUD is enabled per peer.

PMTUD works by setting the Don't Fragment (DF) option bit in the IP header of outgoing packets. The source assumes initially that the path MTU is the MTU of its egress interface. Any device along the path with an MTU smaller than the IPv4 packet will drop the packet and notify the source by sending back an Internet Control Message Protocol (ICMP) "Fragmentation Needed" (type 3, code 4) error message containing its MTU. IPv6 packets larger than the MTU will also be dropped in which case an ICMPv6 error message "Packet Too Big" (type 2, code 0) containing its MTU will be sent back. The source can then reduce its path MTU to this received MTU. The process repeats until the MTU is small enough to traverse the entire path without fragmentation.

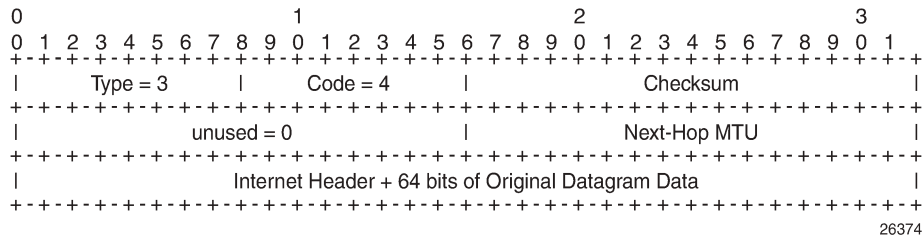
If the path MTU changes to a lower value after the connection is set up, the first larger packet will cause an ICMP error message and the new, lower path MTU will be determined.

PMTUD is used to determine the most efficient packet size for protocols or applications that may send large packets or large data transfers, including BGP updates, LDP, IGP, FTP/TFTP/SCP transfers. With PMTUD enabled, each connection can start with the maximum MTU—based on egress MTU—then allow remote and/or transit routers to lower the effective MTU for the session if the current MTU is too large for one of their next hops. The path MTU is handled and tracked on a per session/connection basis.

All routers along the path must be able to send ICMP error messages of type 3 ("Destination Unreachable") and code 4 ("Fragmentation Needed").

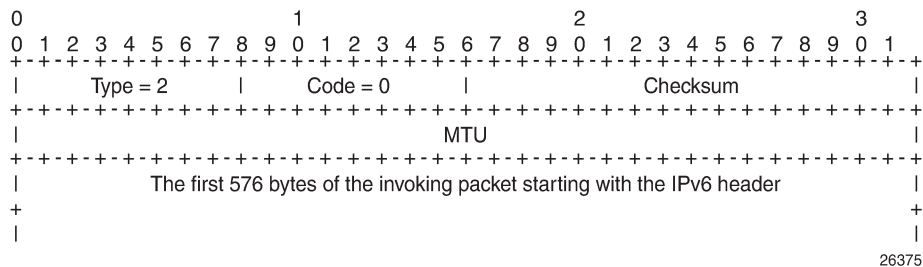
[Figure 246: ICMP "Destination Unreachable" Message - Fragmentation Needed](#) shows the format of such an ICMP message. The next hop MTU is the MTU of the egress interface to the destination of the packet on the router that dropped the packet. The MTU is a count of the octets of the IP header and IP data, without lower-level headers.

Figure 246: ICMP "Destination Unreachable" Message - Fragmentation Needed



The mechanism for IPv6 is similar, but the format of the ICMPv6 message is different. For IPv6, the router will send an ICMPv6 error message of type 2 ("Packet Too Big") and code 0, as shown in [Figure 247: ICMPv6 "Packet Too Big" message](#). The MTU field is populated with the MTU of the egress interface to the destination of the packet on the router that dropped the packet. The MTU is a count of the octets of the IP header and IP data, but no lower-level headers.

Figure 247: ICMPv6 "Packet Too Big" message



When PMTUD is enabled, the IP MTU is initially set to the egress MTU size, based on the source IP interface for that session. When a node along the path is unable to forward a packet due to a smaller MTU, the node drops the packet and sends back an ICMP error message with the MTU of the egress interface. The node that receives the ICMP error message will adjust its MTU accordingly. The IP header and the following bytes of the original IP datagram should be used to determine which connection caused the error.

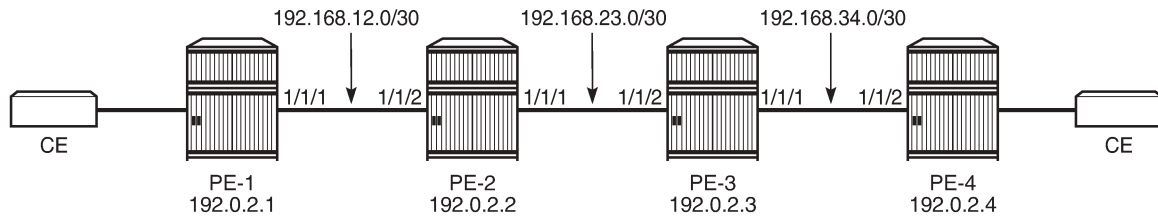
Configuration

The following examples are configured:

- PMTUD in LDP for an IPv4 peer
- PMTUD in LDP for an IPv6 peer
- PMTUD in BGP for an IPv4 peer
- PMTUD in BGP for an IPv6 peer

Figure 248: Example topology shows the example topology with four PE nodes in autonomous system 64496. The interfaces have IPv4 and IPv6 addresses, but in this figure, only the IPv4 addresses are shown.

Figure 248: Example topology



26376

The initial configuration on the nodes includes:

- Cards, MDAs, ports
- Router interfaces with IPv4 and IPv6 address
- IS-IS as IGP on all interfaces between the PEs (alternatively, OSPF can be used)
- LDP enabled on all interfaces between the PEs for IPv4 and IPv6

The initial configuration on PE-1 is as follows:

```
# on PE-1:
configure {
  router "Base" {
    interface "int-PE-1-PE-2" {
      port 1/1/1
      ipv4 {
        primary {
          address 192.168.12.1
          prefix-length 30
        }
      }
      ipv6 {
        address 2001:db8:12:: {
          prefix-length 127
        }
      }
    }
    interface "system" {
      ipv4 {
        primary {
          address 192.0.2.1
          prefix-length 32
        }
      }
      ipv6 {
        address 2001:db8::1 {
          prefix-length 128
        }
      }
    }
  }
  isis 0 {
    admin-state enable
    ipv6-routing native
    area-address [49.0001]
    interface "int-PE-1-PE-2" {
```



```

=====
Ports on Slot B
=====
Port      Admin Link Port  Cfg Oper LAG/ Port Port Port  C/QS/S/XFP/
Id        State      State MTU  MTU  Bndl Mode Encp Type  MDIMDX
-----
B/1      Up      No   Ghost 1514 1514 - netw null faste
B/3      Down   No   Ghost 1514 1514 - netw null faste
B/4      Down   No   Ghost 1514 1514 - netw null faste
=====

```

The network port MTU on the link between PE-2 and PE-3 is configured to 512 for IPv4. For IPv6, this network port MTU on the link between PE-2 and PE-3 is reconfigured with a value of 1300.

The service MTU is 1514, the SAP MTU is 1518 (dot1q encapsulation on access port), and the SDP MTU is 1578 (= 1600 (network port MTU) - 14 (Ethernet) - 8 (2 MPLS labels: service label and transport label)), as shown for an Epipe service on PE-1. The configuration for SDP 14 is shown in section [SDP path MTU for IPv4](#); for Epipe_100_name, in section [PMTUD for LDP IPv4](#). This SDP MTU does not consider the lowest network port MTU in the path, but only the local network MTU.

```

[/]
A:admin@PE-1# show service id 100 base

=====
Service Basic Information
=====
Service Id       : 100                Vpn Id           : 0
Service Type     : Epipe
MACSec enabled   : no
Name             : Epipe_100_name
Description      : (Not Specified)
Customer Id      : 1                Creation Origin   : manual
Last Status Change: 08/05/2021 17:00:44
Last Mgmt Change : 08/05/2021 16:52:43
Test Service     : No
Admin State      : Up                Oper State        : Up
MTU             : 1514
Vc Switching     : False
SAP Count        : 1                SDP Bind Count    : 1
Per Svc Hashing  : Disabled
Vxlan Src Tep Ip : N/A
Force QTag Fwd   : Disabled
Oper Group       : <none>

-----
Service Access & Destination Points
-----
Identifier                Type      AdmMTU  OprMTU  Adm  Opr
-----
sap:1/2/1:100             q-tag    1518    1518    Up   Up
sdp:14:100 S(192.0.2.4)   Spok     0        1578    Up   Up
=====

```

SDP path MTU for IPv4

The network port MTU is configured to 512 on the interfaces between PE-2 and PE-3, as follows:

```

# on PE-2:
configure {

```

```
port 1/1/1 {
    ethernet {
        mtu 512
    }
}
```

```
# on PE-3:
configure {
    port 1/1/2 {
        ethernet {
            mtu 512
        }
    }
}
```

On PE-1, SDP 14 is configured toward PE-4, as follows:

```
# on PE-1:
configure {
    service {
        sdp 14 {
            admin-state enable
            delivery-type mpls
            ldp true
            far-end {
                ip-address 192.0.2.4
            }
        }
    }
}
```

The configuration is similar on PE-4, but with a far end of 192.0.2.1 instead.

The SDP path MTU is derived from the lowest network port MTU in the path: 512 - 14 (Ethernet header) - 4 (MPLS service label) - 4 (MPLS path label) = 490. This can be verified on PE-1 for the end-to-end path with the following OAM command (in classic CLI) that sends packets with an incrementing size: from 400 to 500 bytes in steps of 10 bytes. The packet with size 490 bytes gets a response, whereas the packet with size 500 gets a timeout.

```
A:PE-1# oam sdp-mtu 14 size-inc 400 500 step 10
```

Size	Sent	Response
400	.	Success
410	.	Success
420	.	Success
430	.	Success
440	.	Success
450	.	Success
460	.	Success
470	.	Success
480	.	Success
490	.	Success
500	...	Request Timeout

Maximum Response Size: 490

The next step is to repeat the OAM command to send packets with incrementing size from 490 to 500 in steps of 1:

```
A:PE-1# oam sdp-mtu 14 size-inc 490 500 step 1
Size   Sent   Response
-----
490    .       Success
491    ...     Request Timeout

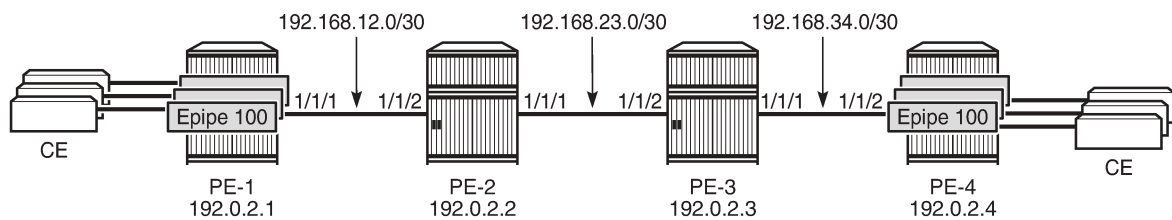
Maximum Response Size: 490
```

The SDP path MTU is 490 bytes.

PMTUD for LDP IPv4

Figure 249: Multiple Epipes Using LDP SDPs shows that multiple Epipe services are configured on PE-1 and PE-4.

Figure 249: Multiple Epipes Using LDP SDPs



26377

The following multiple Epipes are configured on PE-1:

```
# on PE-1:
configure {
  service {
    epipe "Epipe_100_name" {
      admin-state enable
      service-id 100
      customer "1"
      spoke-sdp 14:100 {
      }
      sap 1/2/1:100 {
      }
    }
    epipe "Epipe_101_name" {
      admin-state enable
      service-id 101
      customer "1"
      spoke-sdp 14:101 {
      }
      sap 1/2/1:101 {
      }
    }
  }
  ---snip--- for 102 through 108
  epipe "Epipe_109_name" {
    admin-state enable
    service-id 109
    customer "1"
  }
}
```

```

        spoke-sdp 14:109 {
        }
        sap 1/2/1:109 {
        }
    }
}

```

The following configuration enables PMTUD for LDP IPv4 peer 192.0.2.4 on PE-1. The configuration is similar on PE-4.

```

# on PE-1:
configure {
    router "Base" {
        ldp {
            tcp-session-parameters {
                peer-transport 192.0.2.4 {
                    path-mtu-discovery true
                }
            }
        }
    }
}

```

```
[/]A:admin@PE-1# show router ldp tcp-session-parameters ipv4
```

```

=====
LDP IPv4 TCP Session Parameters
=====
-----
Peer Transport: 192.0.2.4
-----
Authentication Key : Disabled           Path MTU Discovery : Enabled
Auth key chain    :                    Min-TTL           : 0
=====
No. of IPv4 Peers: 1
=====

```

When LDP is disabled and re-enabled on PE-1, all label mappings are signaled again.

```

# on PE-1:
configure {
    router "Base" {
        ldp {
            admin-state disable
            commit
            admin-state enable
            commit
        }
    }
}

```

The size of the LDP label mapping messages may exceed the MTU between PE-2 and PE-3. The DF bit is set, so the packet is discarded at the egress of PE-2 to PE-3. PE-2 sends an ICMP error message of type 3 and code 4 to PE-1. The following ICMP error message is received on PE-1 when debugging is enabled for ICMP (in classic CLI):

```

# on PE-1:
debug
router

```

```

ip
  icmp
exit all

[/]
A:admin@PE-1# show log log-id 2

=====
Event Log 2 log-name 2
=====
Description : (Not Specified)
Memory Log contents [size=100  next event=3  (not wrapped)]

2 2021/08/05 17:17:08.614 CEST MINOR: DEBUG #2001 Base PIP
"PIP: ICMP
instance 1 (Base), interface index 2 (int-PE-1-PE-2),
ICMP ingressing on int-PE-1-PE-2:
 192.168.23.1 -> 192.0.2.1
  type: Destination Unreachable (3)  code: Fragmentation Needed and Don't Fragment was Set (4)
"
---snip---

```

On the egress interface "int-PE-2-PE-3" on PE-2, the network MTU is 512, the IP MTU is 498 (= 512 - 14 (Ethernet header)), and the TCP Maximum Segment Size (OperMss) is 458 (= 498 - 20 (IP header) - 20 (TCP header)), as shown on PE-1:

```

[/]
A:admin@PE-1# show system connections port 646

=====
Connections
=====
Prot   RecvQ   TxmtQ   Local Address           State
      RcvdMss OperMss  Remote Address          vRtrID
-----
TCP    0        0 192.0.2.1.646           LISTEN
      0      1024 0.0.0.0.0               1
TCP    0        0 192.0.2.1.646           ESTABLISH
      1024   1024 192.0.2.2.50025         1
TCP    0        0 192.0.2.1.646           ESTABLISH
      1538   458 192.0.2.4.50933      1
---snip--- for IPv6 addresses
-----
No. of Connections: 5
=====

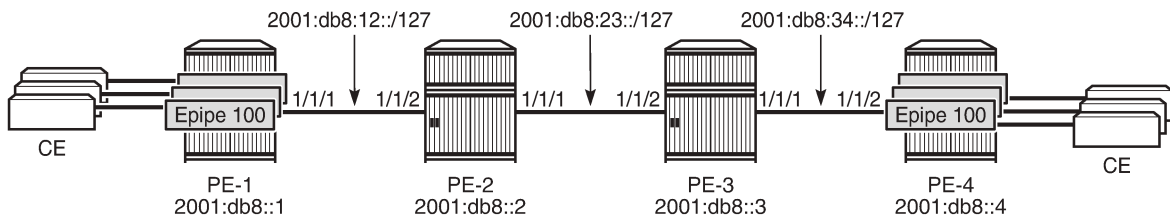
```

TCP port 646 is used for LDP messages. The LDP TCP session with PE-2 keeps the (default) TCP OperMss value of 1024, whereas the LDP TCP session with PE-4 has a reduced TCP OperMss of 458 octets. PE-1 adapts the TCP OperMss size to 458 and retransmits the LDP mapping messages to PE-4. With TCP OperMss set to 458, no fragmentation is required along the path.

PMTUD for LDP IPv6

Multiple Epipes are configured between PE-1 and PE-4. [Figure 250: Multiple Epipes between PE-1 and PE-4 - IPv6](#) shows the IPv6 addresses used.

Figure 250: Multiple Epipes between PE-1 and PE-4 - IPv6



26378

The service configuration is the same as the preceding service configuration, but the far end of the SDP is an IPv6 address instead, as follows:

```
# on PE-1:
configure {
  service {
    sdp 146 {
      admin-state enable
      delivery-type mpls
      ldp true
      far-end {
        ip-address 2001:db8::4
      }
    }
  }
}
```

With the configured network MTU of 512 on the link between PE-2 and PE-3, SDP 146 (for IPv6) is operationally down, whereas SDP 14 (for IPv4) is up, as follows:

```
[/]
A:admin@PE-1# show service sdp

=====
Services: Service Destination Points
=====
SdpId  AdmMTU  OprMTU  Far End          Adm  Opr      Del  LSP  Sig
-----
14     0       1578   192.0.2.4       Up   Up       MPLS L    TLDP
146    0       1578   2001:db8::4     Up   Down     MPLS L    TLDP
-----
Number of SDPs : 2
-----
Legend: R = RSVP, L = LDP, B = BGP, M = MPLS-TP, n/a = Not Applicable
        I = SR-ISIS, 0 = SR-OSPF, T = SR-TE, F = FPE
=====
```

RFC 2460 IPv6 specification states that links with a configurable MTU should have an MTU of at least 1280 octets; preferably 1500 or greater to accommodate possible tunneling encapsulations without the need for fragmentation.

In this example, the network MTU on the link between PE-2 and PE-3 is configured with a value of 1300 and SDP 146 will then be operationally up.

```
# on PE-2:
configure {
```

```
port 1/1/1 {
    ethernet {
        mtu 1300
    }
}
```

```
# on PE-3:
configure {
    port 1/1/2 {
        ethernet {
            mtu 1300
        }
    }
}
```

The SDP path MTU for SDP 146 is 1278 (= 1300 - 14 - 4 - 4). This can be verified on PE-1 with the following OAM command (in classic CLI):

```
A:PE-1# oam sdp-mtu 146 size-inc 1270 1280 step 1
Size    Sent    Response
-----
1270    .        Success
1271    .        Success
1272    .        Success
1273    .        Success
1274    .        Success
1275    .        Success
1276    .        Success
1277    .        Success
1278    .        Success
1279    ...     Request Timeout
```

Maximum Response Size: 1278

PMTUD is enabled for LDP IPv6 peer 2001:db8::4 on PE-1, as follows:

```
# on PE-1:
configure {
    router "Base" {
        ldp {
            tcp-session-parameters {
                peer-transport 2001:db8::4 {
                    path-mtu-discovery true
                }
            }
        }
    }
}
```

```
[/]
A:admin@PE-1# show router ldp tcp-session-parameters ipv6

=====
LDP IPv6 TCP Session Parameters
=====
-----
Peer Transport: 2001:db8::4
-----
Authentication Key : Disabled          Path MTU Discovery : Enabled
Auth key chain     :                   Min-TTL             : 0
```



```
=====
No. of IPv6 Peers: 1
=====
```

With an SDP path MTU of 1280 octets, it is extremely unlikely that LDP packets will exceed this size. An example of an ICMPv6 message that is sent when the packet is too big is shown for BGP in section [PMTUD for BGP IPv6](#).

The TCP OperMss for the IPv6 LDP connection between PE-1 and PE-4 is the default value of 1024 bytes. When the SDP path MTU is big enough for TCP segments with segments of 1024 bytes, the TCP OperMss is set to 1024, unless **tcp-mss** is configured manually on the IPv6 interfaces. This TCP OperMss value may change after an ICMPv6 "Packet Too Big" message is received on PE-1.

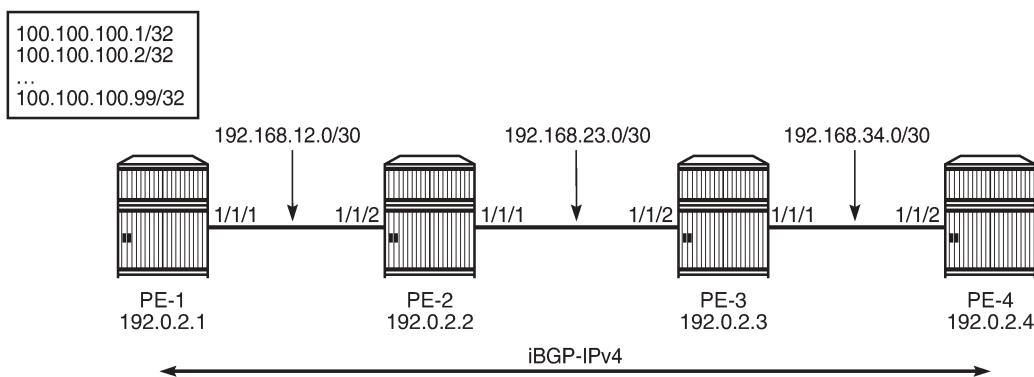
```
[/]
A:admin@PE-1# show system connections address 2001:db8::4 port 646

=====
Connections
=====
Prot  RecvQ  TxmtQ  Local Address  State
  RcvdMss OperMss Remote Address  vRtrID
-----
TCP    0        0 2001:db8::1.646  ESTABLISH
      1518   1518 2001:db8::4.50960
-----
No. of Connections: 1
=====
```

PMTUD for BGP IPv4

[Figure 251: BGP-IPv4](#) shows that a BGP session is established between PE-1 and PE-4 for address family IPv4. Static routes on PE-1 are exported as BGP routes to PE-4.

Figure 251: BGP-IPv4



26379

The network port MTU on the link between PE-2 and PE-3 is set to 512 again:

```
# on PE-2:
configure {
  port 1/1/1 {
    ethernet {
```

```

        mtu 512
    }
}

```

```

# on PE-3:
configure {
    port 1/1/2 {
        ethernet {
            mtu 512
        }
    }
}

```

BGP is configured for address family IPv4 on PE-1, as follows:

```

# on PE-1:
configure {
    policy-options {
        policy-statement "export_static_policy" {
            entry 10 {
                from {
                    protocol {
                        name [static]
                    }
                }
                action {
                    action-type accept
                }
            }
        }
    }
}
router "Base" {
    autonomous-system 64496
    bgp {
        ebgp-default-reject-policy {
            import false
            export false
        }
        group "iBGP_IPv4_name" {
            peer-as 64496
        }
        neighbor "192.0.2.4" {
            group "iBGP_IPv4_name"
            path-mtu-discovery true
            export {
                policy ["export_static_policy"]
            }
        }
    }
}

```

The export policy exports static routes as BGP routes to neighbor 192.0.2.4. PMTUD can be enabled in the global **bgp** context, per **group**, or per **neighbor**. In this example, PMTUD is enabled for neighbor 192.0.2.4. The configuration on PE-4 is similar, but with a neighbor 192.0.2.1 and without any export policy.

Also, a range of static routes is configured on PE-1 to ensure that the size of the BGP update messages will be larger than the SDP path MTU, as follows:

```

# on PE-1:

```

```
configure {
  router "Base" {
    static-routes {
      route 100.100.100.1/32 route-type unicast {
        blackhole {
          admin-state enable
        }
      }
      route 100.100.100.2/32 route-type unicast {
        blackhole {
          admin-state enable
        }
      }
      ---snip--- for 3 through 98
      route 100.100.100.99/32 route-type unicast {
        blackhole {
          admin-state enable
        }
      }
    }
  }
}
```

Debugging is enabled for ICMP (in classic CLI), as follows:

```
# on PE-1:
clear log 2
no debug
debug
  router
    ip
      icmp
      exit all
```

BGP is disabled and re-enabled to ensure that all BGP routes are re-advertised to PE-4. The BGP route update messages exceed the MTU on the egress port of PE-2 to PE-3, and PE-2 should have to fragment them to be able to forward them on the egress interface toward PE-3, but the DF bit is set. Therefore, PE-2 discards the packet and sends an ICMP error message to PE-1 of type 3 ("Destination Unreachable") and code 4 ("Fragmentation Needed and Don't Fragment was Set"). PE-1 receives the following ICMP error message:

```
[/]
A:admin@PE-1# show log log-id 2

=====
Event Log 2 log-name 2
=====
Description : (Not Specified)
Memory Log contents [size=100  next event=6  (not wrapped)]

5 2021/08/05 17:31:46.740 CEST MINOR: DEBUG #2001 Base PIP
"PIP: ICMP
instance 1 (Base), interface index 2 (int-PE-1-PE-2),
ICMP ingressing on int-PE-1-PE-2:
 192.168.23.1 -> 192.0.2.1
  type: Destination Unreachable (3)  code: Fragmentation Needed and Don't Fragment was Set (4)
"
---snip---
```

The following output shows that the TCP OperMss for the BGP connection between PE-1 and PE-4 is 458. TCP destination port 179 is used for BGP traffic.

```
[/]
A:admin@PE-1# show system connections port 179

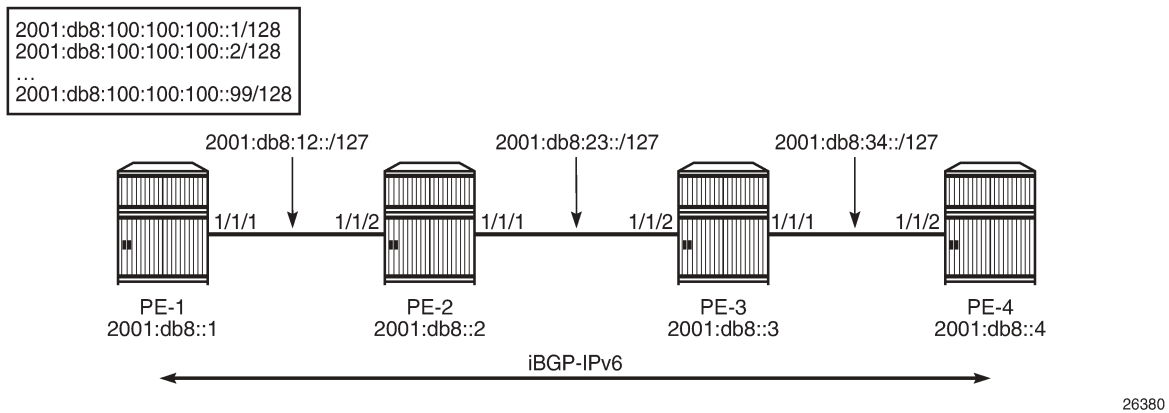
=====
Connections
=====
Prot  RecvQ  TxmtQ  Local Address      State
     RcvdMss OperMss Remote Address      vRtrID
-----
TCP    0       0  0.0.0.0.179      LISTEN
     0      1024  0.0.0.0.0        1
TCP    0       0  192.0.2.1.50595  ESTABLISH
     1538   458  192.0.2.4.179    1
TCP    0       0  ::.179           LISTEN
     0      1024  :::0             1
-----
No. of Connections: 3
=====
```

The TCP OperMss is calculated as follows: $512 - 14 - 20 - 20 = 458$, where 512 is the lowest network port MTU in the path, 14 bytes are used for the Ethernet header, 20 bytes for the IPv4 header, and 20 bytes for the TCP header.

PMTUD for BGP IPv6

Figure 252: BGP-IPv6 shows that a BGP session is established between PE-1 and PE-4 for address family IPv6. PE-1 exports a range of IPv6 routes to PE-4.

Figure 252: BGP-IPv6



The network port MTU on the link between PE-2 and PE-3 is set to 1300 again:

```
# on PE-2:
configure {
  port 1/1/1 {
    ethernet {
      mtu 1300
    }
  }
}
```

```

    }
}

# on PE-3:
configure {
    port 1/1/2 {
        ethernet {
            mtu 1300
        }
    }
}
}

```

The BGP configuration is similar for IPv6 to the configuration for IPv4, only the BGP address family and the neighbor addresses are different. The export policy is identical. PMTUD is enabled in the BGP group "iBGP_IPv6_group". The static routes have now IPv6 addresses.

```

# on PE-1:
configure {
    policy-options {
        policy-statement "export_static_policy" {
            entry 10 {
                from {
                    protocol {
                        name [static]
                    }
                }
                action {
                    action-type accept
                }
            }
        }
    }
}

router "Base" {
    autonomous-system 64496
    bgp {
        group "iBGP_IPv6_name" {
            path-mtu-discovery true
            peer-as 64496
            family {
                ipv6 true
            }
        }
        neighbor "2001:db8::4" {
            group "iBGP_IPv6_name"
            export {
                policy ["export_static_policy"]
            }
        }
    }
    commit
    info
}

static-routes {
    route 2001:db8:100:100:100::1/128 route-type unicast {
        blackhole {
            admin-state enable
        }
    }
    route 2001:db8:100:100:100::2/128 route-type unicast {
        blackhole {
            admin-state enable
        }
    }
}
}

```

```

---snip--- for 3 through 80
route 2001:db8:100:100:100::81/128 route-type unicast {
    blackhole {
        admin-state enable
    }
}
## The trick is to have BGP packets that will not be fragmented at the source PE-1!
## When you have 99 routes, the BGP part contains 1700 bytes and will be fragmented by PE-1.
}
}
}

```

The configuration on PE-4 resembles this configuration, but with a different neighbor address. When the group "iBGP_IPv6_group" is disabled and re-enabled, PE-1 advertises all the IPv6 routes to its peer 2001:db8::4. PE-2 cannot forward the large BGP messages and discards them. PE-2 sends an ICMPv6 error message to PE-1 indicating that the packet is too big (type 2, code 0). PE-1 receives the following ICMPv6 error message:

```

# on PE-1 (debugging is enabled in classic CLI):
clear log 2
no debug
debug
  router "Base"
  ip
    icmp6
  exit all

```

```

[/]A:admin@PE-1# show log log-id 2

=====
Event Log 2 log-name 2
=====
Description : (Not Specified)
Memory Log contents [size=100  next event=10  (not wrapped)]

---snip---
5 2021/08/05 17:37:33.740 CEST MINOR: DEBUG #2001 Base TIP
"TIP: ICMP6_PKT
ICMP6 ingressing on int-PE-1-PE-2 (Base):
  2001:db8:23:: -> 2001:db8::1
  Type: Packet Too Big (2)
  Code: No Code (0)
  MTU : 1286
"
---snip---

```

The MTU is 1286 and includes the IP header and the IP data, but not the Ethernet header. The calculation is as follows: 1300 - 14 = 1286, where 1300 is the lowest network port MTU in the path and 14 bytes are used for the Ethernet header.

On PE-1, the TCP OperMss for BGP traffic with destination address 2001:db8::4 is 1226, as follows:

```

[/]
A:admin@PE-1# show system connections port 179

=====
Connections
=====
Prot  RecvQ  TxmtQ  Local Address                               State

```

	RcvdMss	OperMss	Remote Address	vrtrID
TCP	0	0	0.0.0.0.179	LISTEN
	0	1024	0.0.0.0.0	1
TCP	0	0	192.0.2.1.50595	ESTABLISH
	1538	458	192.0.2.4.179	1
TCP	0	0	:::179	LISTEN
	0	1024	:::0	1
TCP	0	0	2001:db8::1.50650	ESTABLISH
	1518	1226	2001:db8::4.179	1

No. of Connections: 4				
=====				

The TCP OperMss is calculated as follows: $1300 - 14 - 40 - 20 = 1226$, where 1300 is the lowest network port MTU in the path, 14 bytes are used for the Ethernet header, 40 bytes for the IPv6 header, and 20 bytes for the TCP header. This TCP OperMss value is larger than the default value of 1024, so the ICMPv6 "Packet Too Big" message can result in a larger TCP OperMss value.

Conclusion

PMTUD is a technique to determine the MTU size on the network path between two IP hosts, to maximize packet efficiency and avoid packet fragmentation. PMTUD can be enabled for LDP and BGP connections.

Remote Loop-Free Alternate Node Protection

This chapter describes the Remote Loop-Free Alternate Node Protection.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 16.0.R6, but the MD-CLI in the current edition corresponds to SR OS Release 21.2.R1. Remote Loop-Free Alternate (R-LFA) node protection is supported for IS-IS and OSPF in SR OS Release 16.0.R4 and later. There are no prerequisites for this configuration.

Overview

The Loop-Free Alternates (LFAs) computed following the Remote LFA (R-LFA) specifications in RFC 7490 only guarantee point-to-point link protection by using a repair tunnel. The repair tunnel is a Segment Routed (SR) shortest path between the computing router S and the PQ-node, to ensure that the primary protected link SE is avoided. However, the R-LFA link protection algorithm does not guarantee that the repair path toward the PQ node will avoid the primary next hop router E, and that the traffic emerging from the repair tunnel at the PQ node toward the destination router will avoid the primary next hop router E.

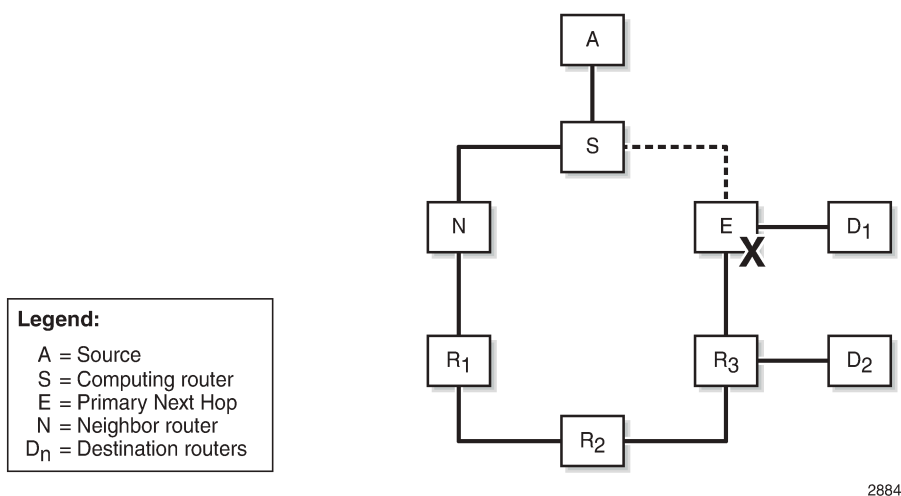
In the remainder of this chapter, SR refers to "Segment Routing", unless specified otherwise. Product and release references, such as 7750 SR and SR OS, continue to refer to "Service Router".

Inequalities for remote LFA node protection

RFC 8102, *Remote LFA node protection*, defines the specifications to protect a path from a source A to a destination D1 or D2, when the primary next hop router E of a computing router S fails; see [Figure 253: LFA node protection - topology & denominations](#). The R-LFA alternate path through a given PQ node to a given destination comprises two path segments:

- path segment from the computing router S to the PQ node (R-LFA alternate next hop)
- path segment from the PQ node to the destination D1 or D2

Figure 253: LFA node protection - topology & denominations



28843

To ensure that an R-LFA alternate path next hop for a given destination provides node protection, none of the path segments may be affected in the event of a failure of the primary next-hop node E. The following four-step algorithm is used to satisfy this requirement:

1. Calculate the node protection extended P-space of router S with respect to the protected node E.
2. Calculate the link protection Q-space of router E with respect to the protected link SE.
Based on the results of step 1 and 2, a list of one or more candidate PQ-routers is compiled.
3. For each candidate PQ-router, perform an additional forward Shortest Path First (SPF) run to ensure that the path from the PQ-router to the destination router does not traverse the protected router E.
4. If more than one candidate PQ-router satisfies the condition from step 3, router S chooses the PQ-router based on criteria that are specified later in this chapter.

The *node protection extended P-space* is the set of routers Y_i that are reachable from the direct neighbor(s) N of S without traversing protected router E. This excludes the direct neighbors for which there is at least one ECMP path from direct neighbor traversing router E. For a router Y_i to be member of a node protection P-space, the following inequality must be true:

$$\text{cost}(N, Y_i) < \text{cost}(N, E) + \text{cost}(E, Y_i)$$

The *link protection Q-space* is the set of routers that can reach E without traversing the protected link SE, as defined in RFC 7490. This excludes equal cost path routes that traverse the SE link. For a router Y_i to be member of the link protection Q-space, the following inequality must be true:

$$\text{cost}(Y_i, E) < \text{cost}(Y_i, S) + \text{cost}(S, E)$$

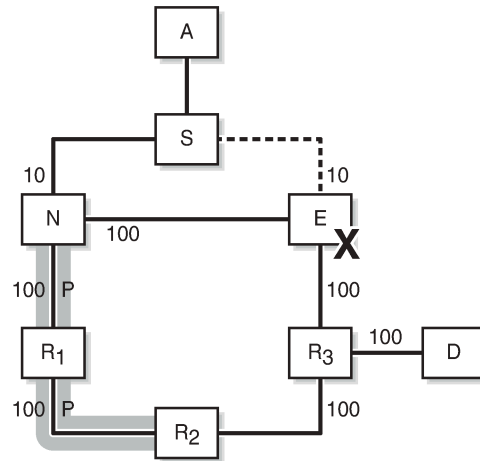
If, with respect to router E, a router Y_i is present in the node protection extended P-space and present in the link protection Q-space, it is a candidate PQ node.

Figure 254: Node protecting extended P-space shows the example topology, with metrics and the calculations in table format to determine the node protection extended P-space of router S with respect to the protected node E. Only routers N, R1, and R2 meet the inequality, and therefore belong to the node protecting extended P-space.

Figure 254: Node protecting extended P-space

Router (Yi)	cost(N,Yi)	cost(N,E)	cost(E,Yi)	Inequality met?
N	0	20	20	Yes (0<20+20)
R ₁	100	20	120	Yes (100<20+120)
R ₂	200	20	200	Yes (200<20+200)
R ₃	120	20	100	No (120<20+100)

Legend:
 A = Source
 S = Computing router
 E = Primary Next Hop
 N = Neighbor router
 D = Destination routers

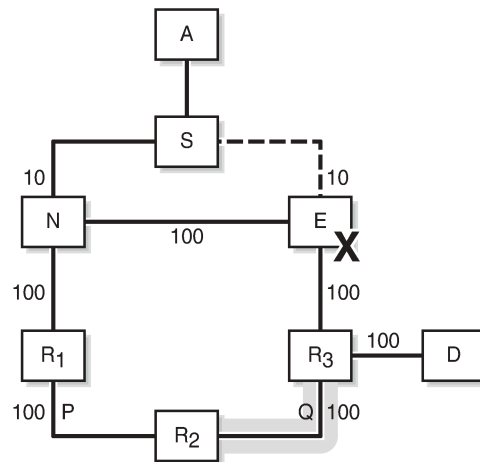


28844

Figure 255: Link protecting Q-space shows the example topology, with metrics and the calculations in table format to determine the link protecting Q-space of router E with respect to the protected link SE. Only routers R2 and R3 meet the inequality, and therefore belong to the link protecting Q-space.

Figure 255: Link protecting Q-space

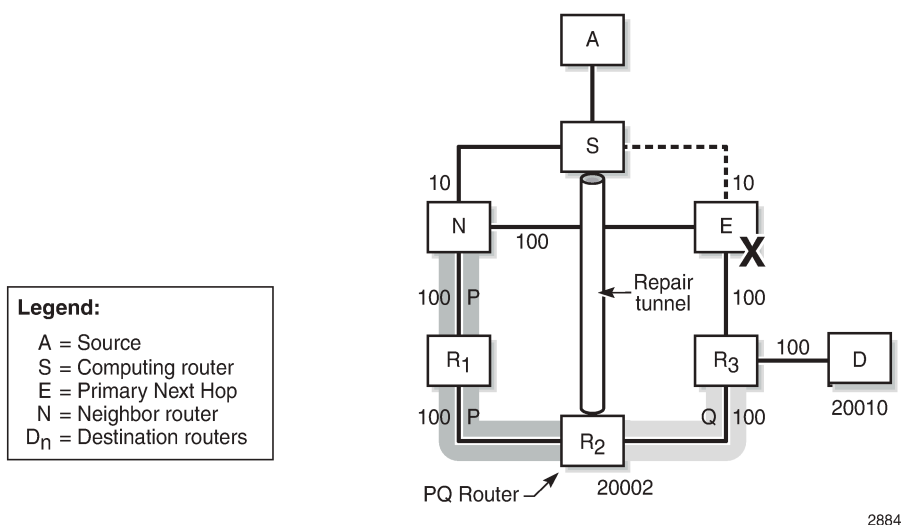
Legend:
 A = Source
 S = Computing router
 E = Primary Next Hop
 N = Neighbor router
 D = Destination routers



28845

Candidate PQ routers are routers that belong to the extended P-space and the Q-space. In this example, only R2 is a candidate PQ node; see Figure 256: One candidate PQ-router – repair tunnel.

Figure 256: One candidate PQ-router – repair tunnel



An additional forward SPF run is required to check that the shortest path from the candidate PQ node R2 toward destination D *does not* traverse protected node E. Therefore, the following inequality must be met:

$$\text{cost}(PQ_i, D) < \text{cost}(PQ_i, E) + \text{cost}(E, D)$$

Applied to this topology, the R2-R3-D path does not go via E; therefore, R2 is a valid R-LFA node protection PQ node. The previous inequality evaluates to true, as follows:

$$200 < 200 + 200 \text{ (True)}$$

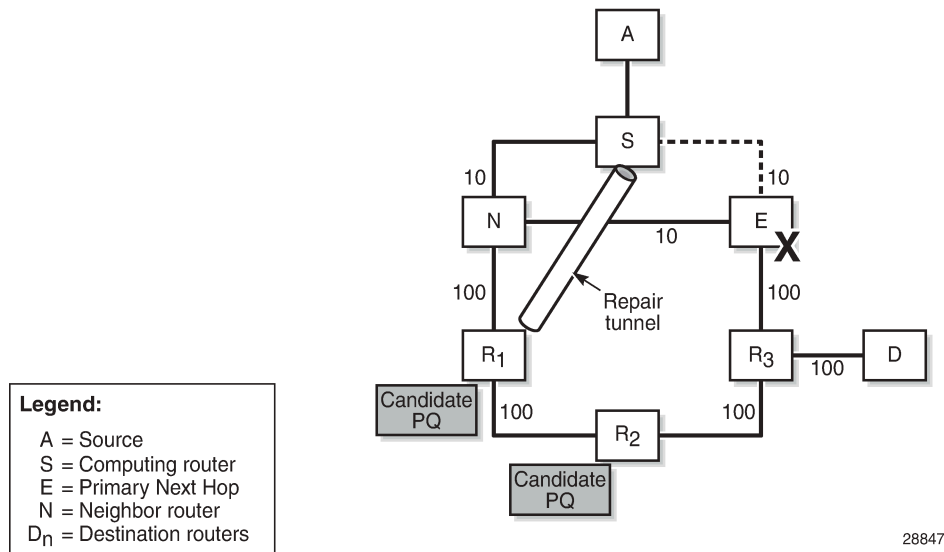
Figure 256: One candidate PQ-router – repair tunnel shows that S constructs a repair tunnel to PQ router R2. To reach destination D from S using the repair tunnel, S pushes a (20002, 20010) label stack, where 20002 and 20010 represent the node SIDs for R2 and D, respectively, while additionally setting the next-hop address to router N. The label 20002 is the first and top label swapped at N and R1, and popped at R2, while 20010 is the second label pushed at R2, swapped at R3, and ultimately popped at D.

In case multiple candidate PQ routers are available, the computing node S selects a PQ router based on the following criteria:

1. lowest IGP path cost from S
2. if multiple PQ routers satisfy (1), S selects the PQ router reachable via the neighbor with the lowest system-ID or router-ID for IS-IS and OSPF, respectively
3. if multiple PQ routers satisfy (1) and (2), S selects the PQ router with the lowest system-ID or router-ID for IS-IS and OSPF, respectively

Figure 257: Two candidate PQ routers – repair tunnel shows an example, with reduced metric between N and E, where R1 and R2 are the candidate PQ routers for protecting router E. In this example, R1 is chosen as the PQ router, because R1 is closer to S than R2. Router S will create an R-LFA repair tunnel for prefixes downstream of R3. To reach those prefixes, the R1 node SID and the D node SID are pushed, with N as the next hop. Prefixes downstream of N, R1, and R2 are unaffected by a failure of E, so they keep using N as their primary next-hop.

Figure 257: Two candidate PQ routers – repair tunnel



28847

LFA and remote LFA interaction

The LFA and remote LFA CLI commands are applied in the OSPF and IS-IS router contexts. The configuration in the IS-IS context is as follows:

```
configure {
  router "Base" {
    isis 0 {
      loopfree-alternate {
        remote-lfa {
          node-protect
        }
      }
    }
  }
}
```

Regular LFA is enabled through the **loopfree-alternate** command. Additionally, the **remote-lfa** and **remote-lfa { node-protect }** command can be configured. In other words, by enabling remote LFA, regular LFA is also enabled.



Note:

A remote LFA repair tunnel is only calculated and created if no regular LFA backup next-hop exists. If this is a concern, Topology Independent LFA (TI-LFA) should be enabled; see the [Topology-Independent Loop-Free Alternate for Link Protection](#) chapter.

The LFA SPF algorithms are run using the following sequence:

1. A regular LFA is computed for each router and prefix, to provide a backup next-hop per prefix.
2. TI-LFA is computed for all routers and prefixes regardless of the outcome of step 1, and the TI-LFA computed next-hops override the regular LFA next-hops, if TI-LFA is enabled.
3. Remote LFA SPF is only run for the prefixes that are not protected after steps 1 and 2.

As a result, remote LFA next-hops, whether link or node protecting, are only computed and installed when no regular LFA next-hops are available for a given next-hop failure, assuming that TI-LFA is not configured. When the **remote-lfa { node-protect }** command is enabled, the router will prefer a node protect over a link-protect repair tunnel for a given prefix if both are found in the Remote LFA SPF computations.

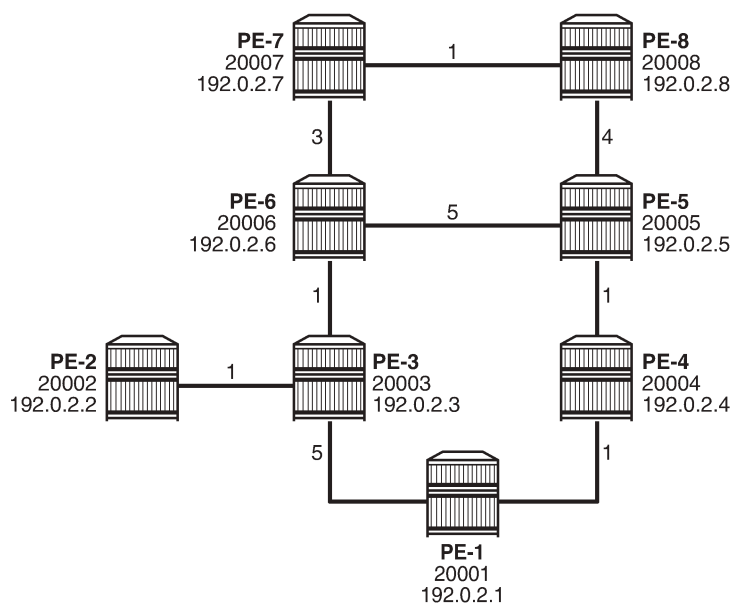
Configuration

Three steps demonstrate the relationship between regular LFA and remote LFA, based on the example topology shown in [Figure 258: Example topology](#). The traffic flow is going from PE-7 to PE-2, and a failure of PE-6 is simulated so that PE-7 is the computing router S, PE-2 is the destination router D, PE-6 is the failing primary next hop E, and PE-8 is the primary backup neighbor N.

The following scenarios are described:

- enable regular LFA on PE-7 — node or link protection cannot be provided for prefixes downstream to PE-6
- enable remote LFA link protection on PE-7 — define the repair tunnel
- enable remote LFA node protection on PE-7 — define the repair tunnel

Figure 258: Example topology



28848

The configuration includes the following:

- Cards, MDAs, ports
- Single stack router interfaces (IPv4 only)
- IS-IS as IGP on the router interfaces. The metrics shown in [Figure 258: Example topology](#) are used.
- Segment routing (SR-ISIS) with node SIDs 2000x

The system addresses and the node SIDs for all routers are also shown in [Figure 258: Example topology](#).

Regular LFA

The Segment Routing Global Block (SRGB) is defined consistently across all nodes in the network, as follows:

```
# on all nodes:
configure {
  router "Base" {
    mpls-labels {
      sr-labels {
        start 20000
        end 20099
      }
    }
  }
}
```

The IS-IS configuration on PE-7 is as follows, and has regular LFA enabled:

```
# on PE-7:
configure {
  router "Base" {
    mpls-labels {
      sr-labels {
        start 20000
        end 20099
      }
    }
  }
  isis 0 {
    admin-state enable
    level-capability 2
    area-address 49.0001.0000
    traffic-engineering true
    advertise-router-capability area
    segment-routing {
      admin-state enable
      prefix-sid-range {
        global
      }
    }
  }
  loopfree-alternate {
  }
  interface "system" {
    admin-state enable
    ipv4-node-sid {
      index 7
    }
  }
  interface "int-PE-7-PE-6" {
    admin-state enable
    interface-type point-to-point
    level 2 {
      metric 3
    }
  }
  interface "int-PE-7-PE-8" {
    admin-state enable
    interface-type point-to-point
    level 2 {
      metric 1
    }
  }
}
```

```
}

```

PE-7 calculates the *regular LFA node protection* for prefixes downstream of PE-6. The shortest path from the primary backup neighbor PE-8 to router PE-2 must be less than the shortest path from the backup neighbor PE-8 node via PE-6, so the inequality becomes:

$$\text{cost}(\text{PE-8,PE-2}) < \text{cost}(\text{PE-8,PE-6}) + \text{cost}(\text{PE-6,PE-2})$$

$$(1 + 3 + 1 + 1) < (1 + 3) + (1 + 1) \text{ (False)}$$

PE-7 calculates the *regular LFA link protection* for the PE-6-PE-7 link for prefixes downstream of PE-6. The shortest path from the primary backup neighbor PE-8 to router PE-2, must be less than the shortest path from the backup neighbor PE-8 via PE-7, so the inequality becomes:

$$\text{cost}(\text{PE-8,PE-2}) < \text{cost}(\text{PE-8,PE-7}) + \text{cost}(\text{PE-7,PE-2})$$

$$(1 + 3 + 1 + 1) < 1 + (3 + 1 + 1) \text{ (False)}$$

Because both inequalities are false, PE-7 cannot provide regular LFA PE-6 node protection or regular LFA PE-6-PE-7 link protection.

Remote LFA with link protection

On PE-7, LFA is reconfigured so that *remote LFA with link protection* applies, as follows:

```
# on PE-7:
configure {
  router "Base" {
    isis 0 {
      loopfree-alternate {
        remote-lfa {
        }
      }
    }
  }
}
```

A repair tunnel will be established, avoiding and protecting the PE-6-PE-7 link, where the endpoint of the repair tunnel is situated on a PQ router.

[Figure 259: Link protection extended P-space calculation](#) provides the calculations in table format, along with a graphical representation, to determine the link protecting extended P-space of router PE-7 with respect to the protected PE-6-PE-7 link. Routers PE-1, PE-4, and PE-5 meet the inequality, and therefore belong to the link protecting extended P-space, meaning that they can be reached from backup neighbor PE-8 using an SPF path excluding the PE-6-PE-7 link.

Figure 259: Link protection extended P-space calculation

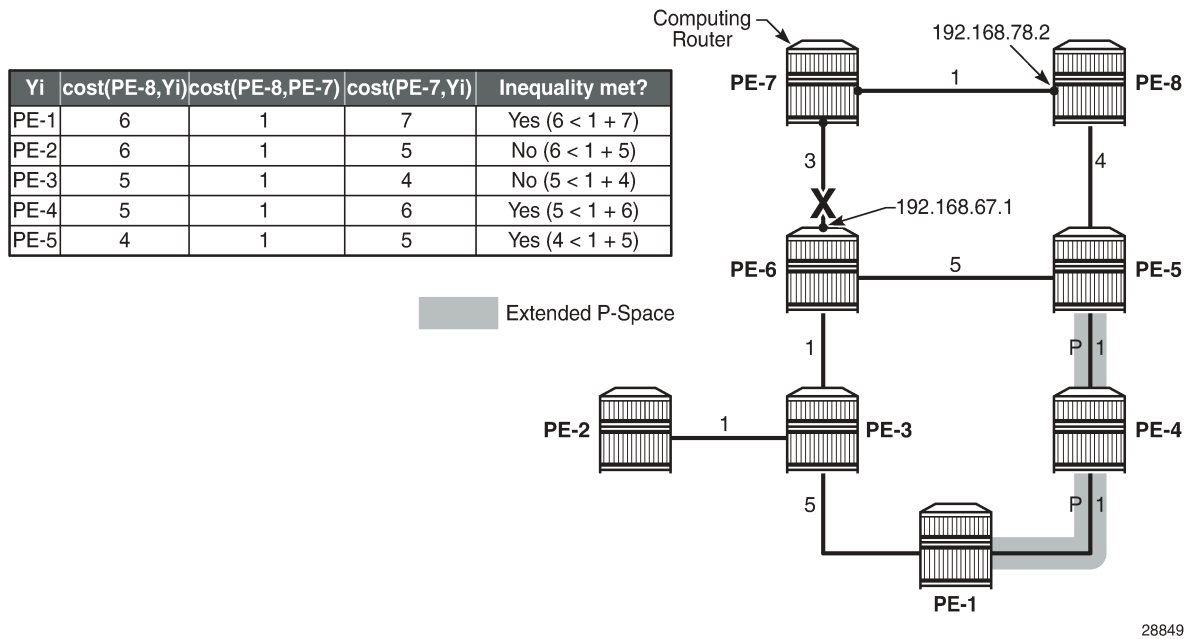


Figure 260: Link protecting Q-space calculation provides the calculations in table format, along with a graphical representation, to determine the link protecting Q-space of router PE-6 with respect to protected PE-7-PE-6 link. Routers PE-1, PE-2, PE-3, PE-4, and PE-5 meet the inequality, and therefore belong to the link protecting Q-space.

Figure 260: Link protecting Q-space calculation

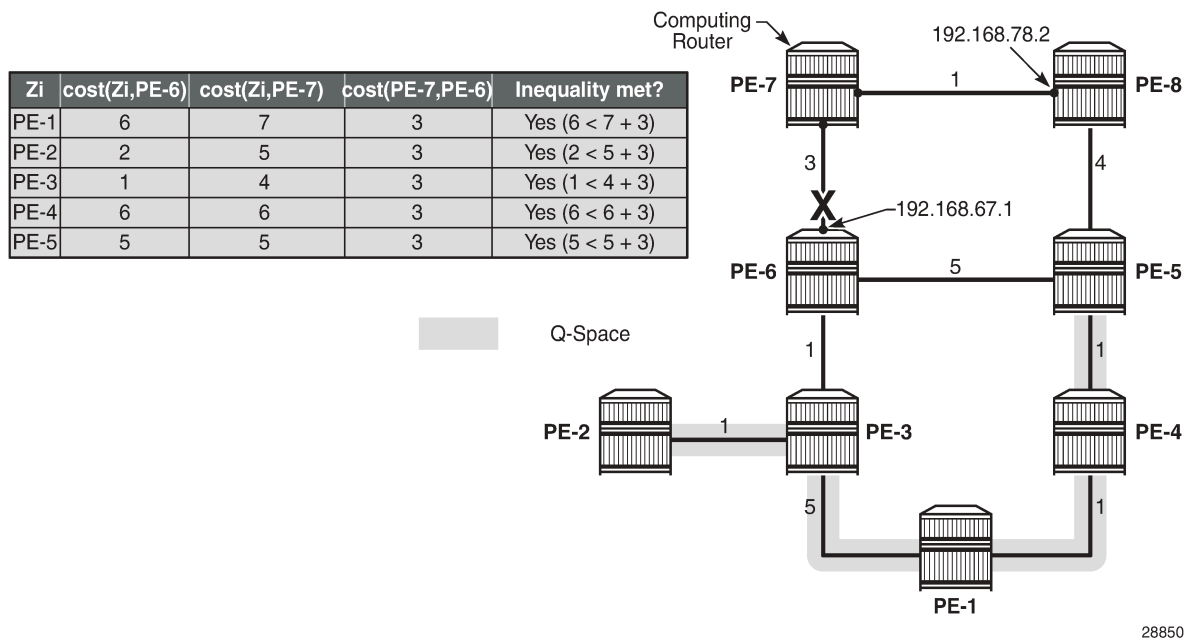
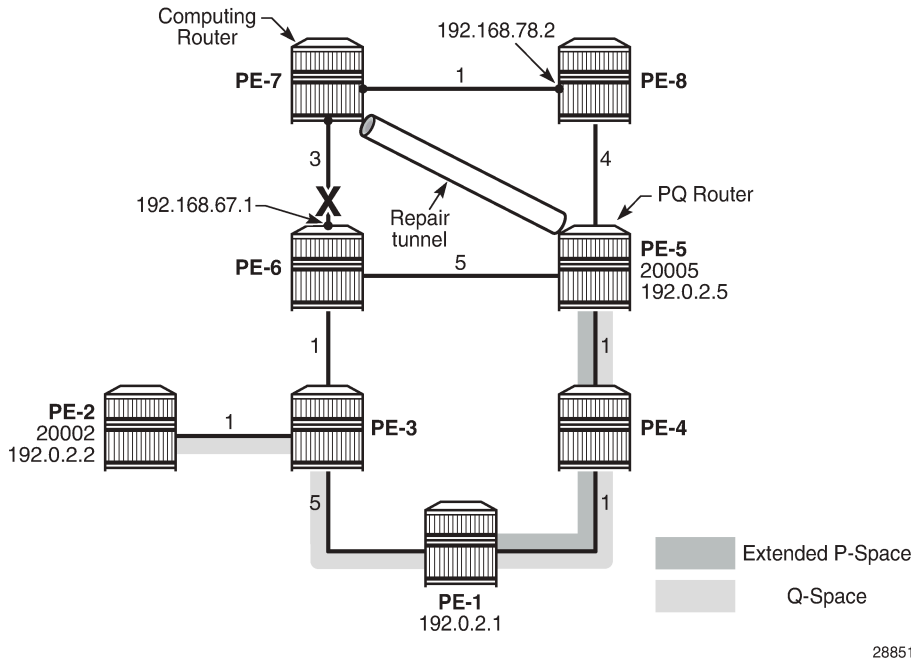


Figure 261: Repair tunnel shows that PE-1, PE-4, and PE-5 are the candidate PQ routers. PE-5 is chosen as the repair tunnel endpoint because of the lowest path cost toward computing node PE-7 (IGP cost from PE-7 to PE-5 = 5). The closest PQ router is chosen to maximize the opportunity for load sharing traffic between the repair tunnel endpoint and the destination router.

Figure 261: Repair tunnel



On the computing node PE-7, the tunnel table for PE-2 destination (192.0.2.2) on IOM 1 shows that 192.168.67.1 is the next hop for the primary path, and that 192.168.78.2 is the next hop for the backup path, as follows. In the normal situation, the PE-7 to PE-2 traffic is routed along the PE-7-PE-6-PE-3-PE-2 path. In case of a PE-7-PE-6 link failure, the traffic on PE-7 node is pushed out with labels 20002 and 20005 to PE-8 (192.168.78.2). The top label is 20005, representing the node SID for PE-5, and 20002 is the label representing the node SID for PE-2.



Note:

Traffic destined for PE-2 and arriving at PE-5 with label 20005 will take the shortest path to PE-2 and therefore will traverse node PE-6.

```
[/]
A:admin@PE-7# show router fp-tunnel-table 1 192.0.2.2/32

=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol          Tunnel-ID
Lbl
NextHop
Lbl      (backup)                            Intf/Tunnel
```

```

NextHop (backup)
-----
192.0.2.2/32                               SR-ISIS-0           524292
20002
  192.168.67.1                             1/1/2
20002/20005
  192.168.78.2(B)                         1/1/1
-----
Total Entries : 1
=====

```

Similar information can be obtained with a **tools dump** command, as follows:

```

[/]
A:admin@PE-7# tools dump router segment-routing tunnel in-label 20002
=====
Legend: (B) - Backup Next-hop for Fast Re-Route
        (D) - Duplicate
label stack is ordered from top-most to bottom-most
=====
-----
----+
Prefix
Sid-Type      Fwd-Type      In-Label  Prot-Inst(algoId)      Out-Label(s) Interface
              Next Hop(s)
              /Tunnel-ID |
-----
----+
192.0.2.2
Node          Orig/Transit   20002     ISIS-0
              192.168.67.1
              (B)192.168.78.2
              20002     int-PE-7-PE-6
              20005     int-PE-7-PE-8
              20002
-----
----+
No. of Entries: 1
-----
----+

```

Another tools command indicates the used LFA type through flags, as follows. Only RLFA link protection applies, and not node protection.

```

[/]
A:admin@PE-7# tools dump router isis sr-database prefix 192.0.2.2 sid 2
=====
Rtr Base ISIS Instance 0 SR Database
=====
SID  Label Prefix      Last-act Lev MT RtmPref TtmPref Metric IpNh SrNh
   Mtu  MtuPrim MtuBk  D xL LT Act AdvSystemId  SrErr
-----
 2    20002 192.0.2.2      LfaNhops 2  0  18    11    5    1    1
   1556  1564  1564  0  0  R  +R  1920.0000.2002 SR_ERR_OK
-----
No. of Entries: 1
-----
Lev = route level
IpNh = number of IP next-hops

```

```
SrNh = number of SR-tunnel next-hops
D = duplicate pending
xL = exclude from LFA
LT = LFA type (L:LFA, R:RLFA, T:TILFA, n:nodeProtection)
Act = tunnel active state (R:reported, F:failed, +:SR-ack)
=====
```

Remote LFA with node protection

On PE-7, LFA is reconfigured so that remote LFA with node protection applies, as follows:

```
# on PE-7:
configure {
  router "Base" {
    isis 0 {
      loopfree-alternate {
        remote-lfa {
          node-protect
        }
      }
    }
  }
}
```

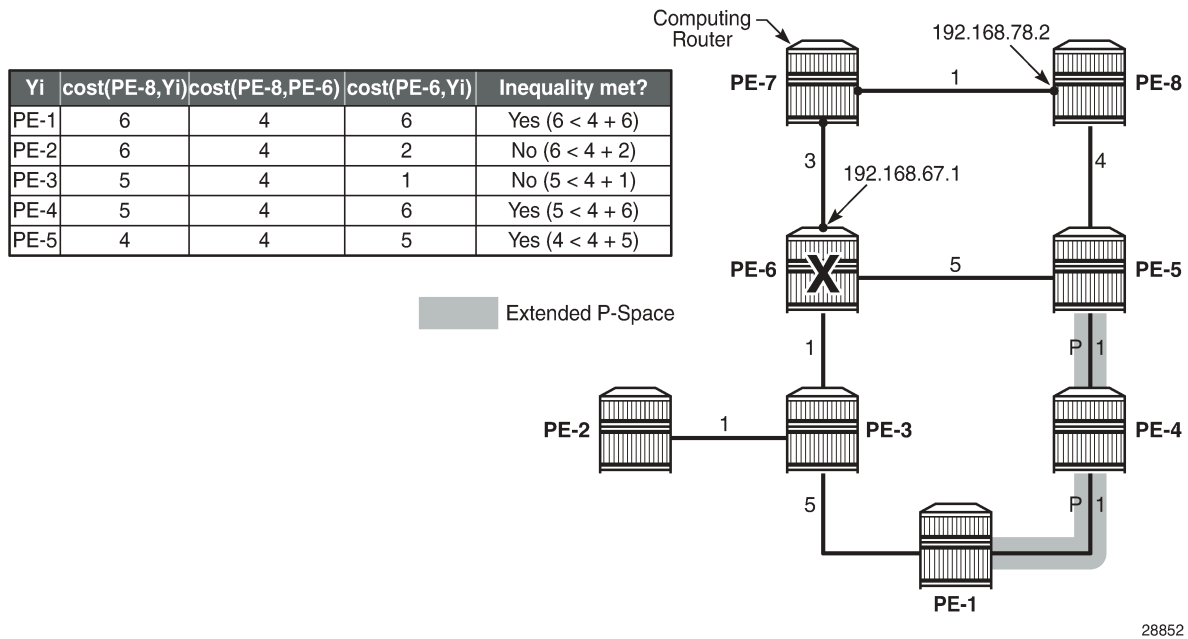
The general node protecting inequality from the [Overview](#) section must be used for defining the node protecting extended P-space. Using the topology from [Figure 258: Example topology](#), the inequality becomes:

$$\text{cost}(N, Y_i) < \text{cost}(N, E) + \text{cost}(E, Y_i)$$

$$\text{cost}(\text{PE-8}, Y_i) < \text{cost}(\text{PE-8}, \text{PE-6}) + \text{cost}(\text{PE-6}, Y_i)$$

[Figure 262: Node protecting extended P-space calculation](#) provides the calculations in table format, along with a graphical representation, to determine the node protecting extended P-space of router PE-7 with respect to protected PE-6 node. Routers PE-1, PE-4, and PE-5 meet the inequality, and therefore belong to the node protecting extended P-space, meaning that they can be reached from backup neighbor PE-8 through an SPF path not passing through node PE-6.

Figure 262: Node protecting extended P-space calculation



The general link protecting inequality from the overview section must be used for defining the Q-space. Using the topology from [Figure 258: Example topology](#), the inequality becomes:

$$\text{cost}(Z_i, E) < \text{cost}(Z_i, S) + \text{cost}(S, E)$$

$$\text{cost}(Z_i, \text{PE-6}) < \text{cost}(Z_i, \text{PE-7}) + \text{cost}(\text{PE-7}, \text{PE-6})$$

[Figure 263: Link protecting Q-space calculation](#) provides the calculations in table format, along with a graphical representation, to determine the link protecting Q-space of router PE-6 with respect to the protected PE-7-PE-6 link. Routers PE-1, PE-2, PE-3, PE-4, and PE-5 meet the inequality, and therefore belong to the link protecting Q-space.

Figure 263: Link protecting Q-space calculation

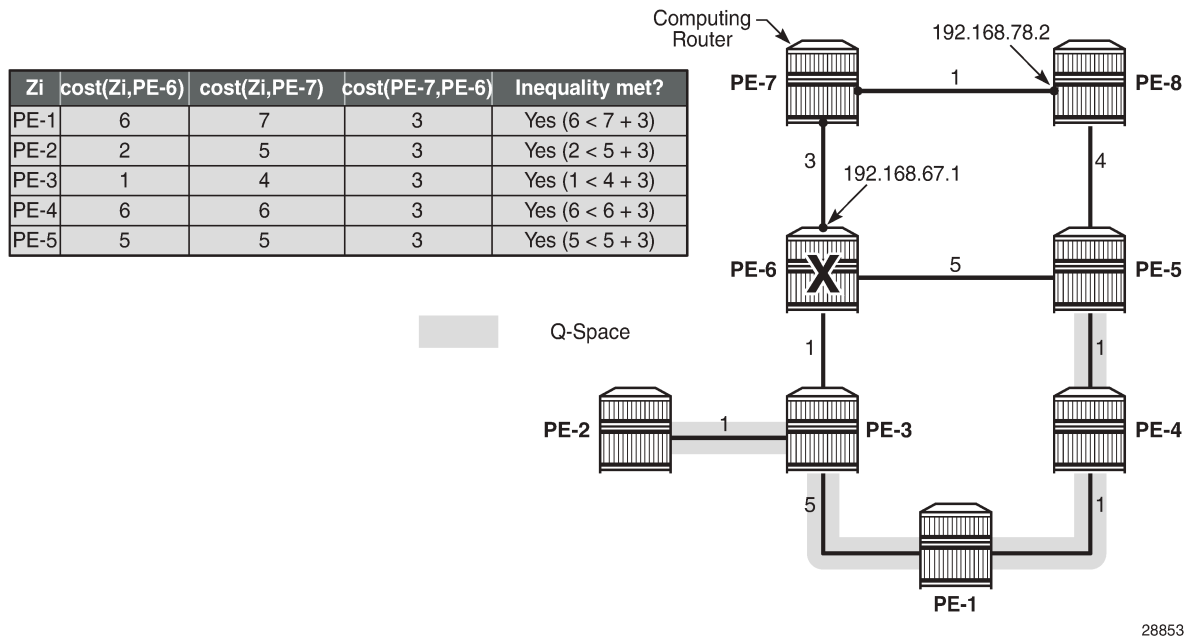
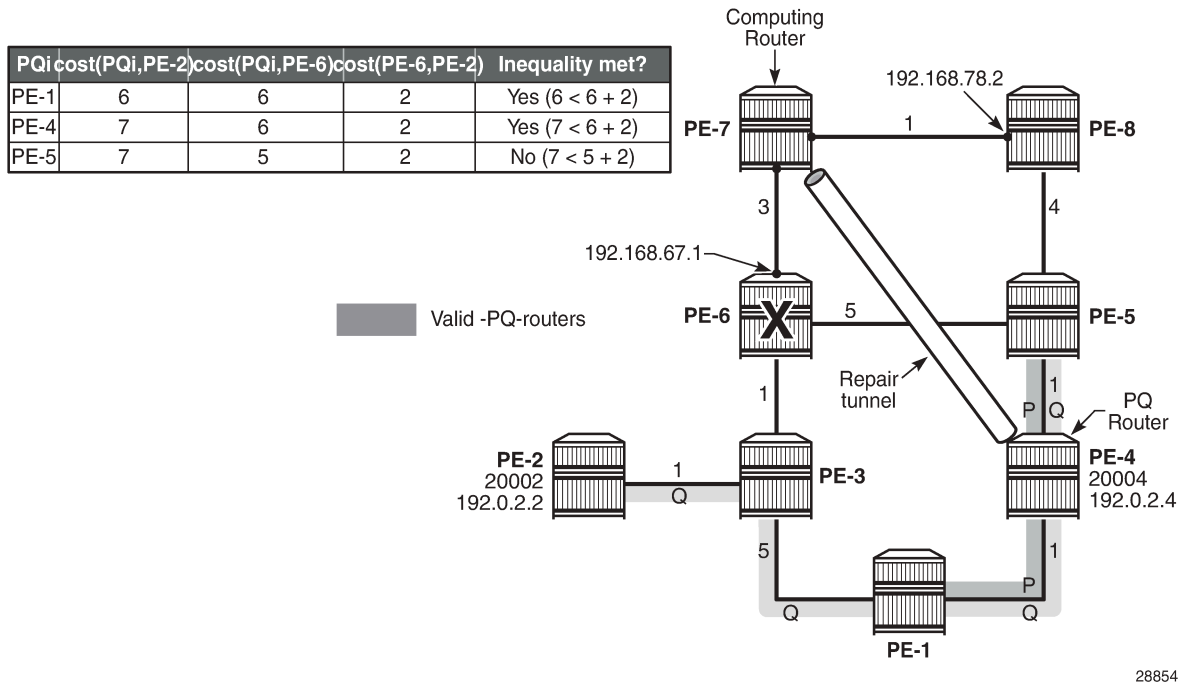


Figure 264: Validating candidate PQ routers - repair tunnel calculation shows that PE-1, PE-4, and PE-5 are the candidate PQ routers for protecting router PE-6. An additional forward SPF run is required for every candidate PQ router, to ensure that the shortest path from that candidate PQ router to destination PE-2 does not traverse the protected router PE-6. The general formula from the Overview section becomes:

$$\text{cost}(PQ_i, D) < \text{cost}(PQ_i, E) + \text{cost}(E, D)$$

$$\text{cost}(PQ_i, PE-2) < \text{cost}(PQ_i, PE-6) + \text{cost}(PE-6, PE-2)$$

Figure 264: Validating candidate PQ routers - repair tunnel calculation



After validating all three candidate PQ routers, only routers PE-1 and PE-4 are valid for terminating a repair tunnel. The tie-breaker for defining the repair tunnel termination is the lowest IGP path cost from the computing node PE-7 point of view. The cost from PE-7 to PE-4 is lower than the cost from PE-7 to PE-1 (6 < 7), so PE-4 becomes the PQ router.

The tunnel table for destination 192.0.2.2 on IOM 1 shows that 192.168.67.1 is the next hop for the primary path, and that 192.168.78.2 is the next hop for the backup path, as follows. In the normal situation, the PE-7 to PE-2 traffic is routed along the PE-7-PE-6-PE-3-PE-2 path. In case of a PE-6 node failure, the traffic from PE-7 is pushed out to PE-8 (192.168.78.2), with two labels. The label 20004 represents the node SID for PQ node PE-4 and is used as the top (first) label, while 20002 represents the node SID for PE-2 and is used as the second label.

```
[/]
A:admin@PE-7# show router fp-tunnel-table 1 192.0.2.2/32

=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol      Tunnel-ID
Lbl                                         NextHop      Intf/Tunnel
Lbl (backup)                               NextHop (backup)
-----
192.0.2.2/32                               SR-ISIS-0    524292
20002                                       192.168.67.1 1/1/2
```

```

20002/20004
192.168.78.2(B) 1/1/1
-----
Total Entries : 1
-----
=====

```

Similar information can be obtained with a **tools dump** command, as follows:

```

[/]
A:admin@PE-7# tools dump router segment-routing tunnel in-label 20002
=====
====
Legend: (B) - Backup Next-hop for Fast Re-Route
        (D) - Duplicate

Label stack is ordered from top-most to bottom-most

=====
====
-----+
Prefix
|
Sid-Type      Fwd-Type      In-Label  Prot-Inst(algoId)
|
|              Next Hop(s)
|
-----+
192.0.2.2
Node          Orig/Transit  20002      ISIS-0
              192.168.67.1
              (B)192.168.78.2
                                20002      int-PE-7-PE-6
                                20004      int-PE-7-PE-8
                                20002
-----+
No. of Entries: 1
-----+

```

Another tools command indicates the used LFA type through flags, as follows. RLFA and node protection applies.

```

[/]
A:admin@PE-7# tools dump router isis sr-database prefix 192.0.2.2 sid 2
=====
Rtr Base ISIS Instance 0 SR Database
=====
SID  Label Prefix          Last-act Lev MT RtmPref TtmPref Metric IpNh SrNh
Mtu  MtuPrim MtuBk  D xL LT Act AdvSystemId  SrErr
-----+
2    20002 192.0.2.2          LfaNhops 2  0  18    11    5    1    1
    1556  1564  1564  0  0  Rn +R  1920.0000.2002 SR_ERR_OK
-----+
No. of Entries: 1
-----+
Lev = route level
IpNh = number of IP next-hops
SrNh = number of SR-tunnel next-hops

```

```
D = duplicate pending
xL = exclude from LFA
LT = LFA type (L:LFA, R:RLFA, T:TILFA, n:nodeProtection)
Act = tunnel active state (R:reported, F:failed, +:SR-ack)
=====
```

The LFA coverage is as follows:

```
[/]
A:admin@PE-7# show router isis sr-lfa-coverage

=====
Rtr Base ISIS Instance 0 SR LFA Coverage
=====
MT-ID  SidType      Level Proto LFA      RLFA      TILFA      Coverage
-----
0      node-sid     L2    ipv4  3(42%)  4(57%)   0(0%)      7/7(100%)
0      adj-sid      L2    ipv4  0(0%)   2(100%)  0(0%)      2/2(100%)
=====
```

Conclusion

Remote LFA Node Protection provides operators the means to create resilient networks, with precalculated backup paths and with improved coverage.

RSVP Point-to-Point LSPs

This chapter provides information about point-to-point label switched paths (LSPs) established using resource reservation protocol (RSVP) with or without traffic engineering (TE).

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 7.0.R5, but the MD-CLI in the current edition is based on SR OS Release 21.2.R1. There are no prerequisites or conditions on the hardware for this configuration.

Overview

Due to the connectionless nature of the network layer protocol IP, packets travel through the network on a hop-by-hop basis with routing decisions made at each node. As a result, hyperaggregation of data on certain links may occur and it may impact the provider's ability to provide guaranteed service levels across the network end-to-end. To address these shortcomings, Multi-Protocol Label Switching (MPLS) was developed. MPLS provides the capability to establish connection-oriented paths, called Label Switched Paths (LSPs), over a connectionless (IP) network.

The LSP offers a mechanism to engineer network traffic independently from the underlying network routing protocol (mostly IP) to improve the network resiliency and recovery options and to permit delivery of new services that are not readily supported by conventional IP routing techniques, such as Layer 2 IP Virtual Private Networks (VPNs). These benefits are essential for today's communication network explaining the wide deployment base of the MPLS technology.

RFC 3031, *Multiprotocol Label Switching Architecture*, specifies the MPLS architecture whereas this document describes the configuration and troubleshooting of RSVP point-to-point LSPs on SR OS. Besides RSVP P2P LSPs, there are also [Static Point-to-Point LSPs](#), [LDP Point-to-Point LSPs](#), and Segment Routing (SR) LSPs (SR-ISIS, SR-OSPF, and SR-TE). For SR-ISIS, see chapter [Segment Routing with IS-IS Control Plane](#).

Packet forwarding

As a packet of a connectionless network layer protocol travels from one router to the next, each router in the network makes an independent forwarding decision by performing the following basic tasks: first analyzing the packet header, then referencing the local routing table to find the longest match based on the destination address in the IP header, and finally sending out the packet on the selected interface. In other words, the first function partitions the entire set of possible packets into a set of Forwarding Equivalence Classes (FECs). All packets associated to a particular FEC will be forwarded along the same logical path

to the same destination. The second function maps each FEC to a next hop destination router. Each router along the data path performs these actions.

In MPLS, the assignment of a packet to a particular FEC is done just once, as the packet enters the network. In turn, the FEC is mapped to an LSP, which is pre-signaled prior to any data flowing. An MPLS label, representing the FEC to which the packet is assigned, is attached to the packet (push operation) and once labeled, the packet is forwarded to the next hop router along that LSP path.

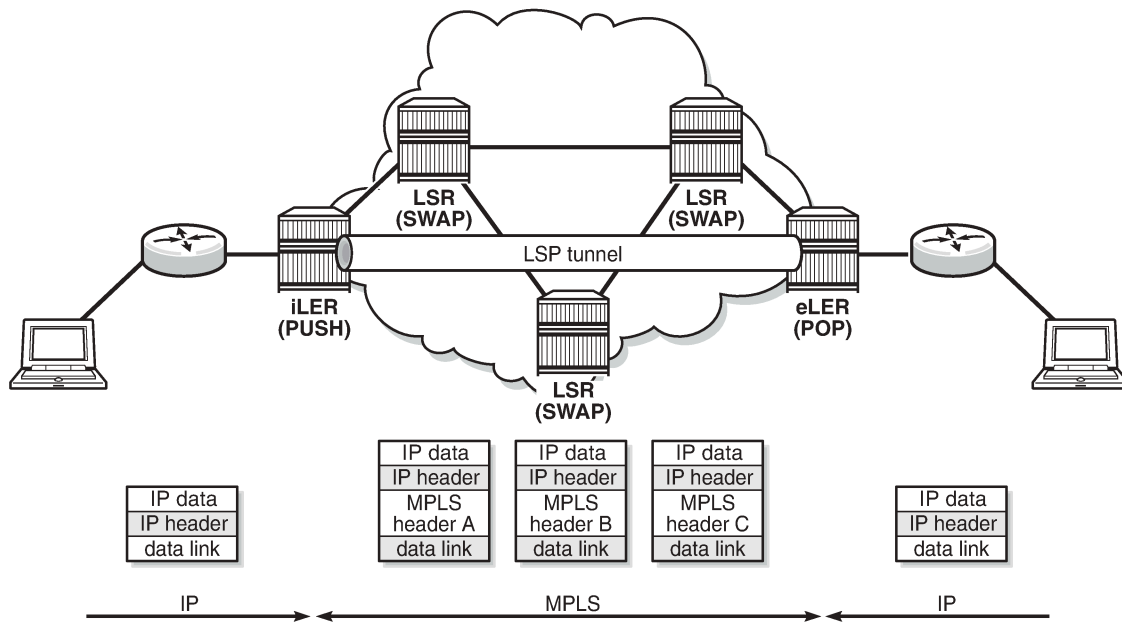
At subsequent hops, there is no further analysis of the packet network layer header. Instead, the label is used as an index into a table which specifies the next hop and a new label. The old label is replaced with the new label (swap operation), and the packet is forwarded to its next hop.

At the MPLS network egress, the label is removed from the packet (pop operation). If this router is the destination (based on the remaining packet), the packet is handed to the receiving application, such as a Virtual Private LAN Service (VPLS). If this router is not the final destination of the packet, the packet will be sent into a new MPLS tunnel or forwarded by conventional IP forwarding toward the Layer 3 destination.

Terminology

[Figure 265: Generic MPLS network, MPLS label operations](#) shows a general network topology clarifying the MPLS-related terms.

Figure 265: Generic MPLS network, MPLS label operations



25762

A Label Edge Router (LER) is a device at the edge of an MPLS network, with at least one interface outside the MPLS domain. A router is usually defined as an LER based on its position relative to a particular LSP. The MPLS router at the head-end of an LSP is called the ingress Label Edge Router (iLER). The MPLS router at the tail-end of an LSP is called the egress Label Edge Router (eLER).

The iLER receives unlabeled packets from outside the MPLS domain, then applies MPLS labels to the packets, and forwards the labeled packets into the MPLS domain. The eLER receives labeled packets from

the MPLS domain, then removes the labels, and forwards unlabeled packets outside the MPLS domain. The eLER can signal an implicit-null label (numeric value 3). This informs the previous hop to send MPLS packets without an outer label and is known as Penultimate Hop Popping (PHP).

A Label Switching Router (LSR) is a device internal to an MPLS network, with all interfaces inside the MPLS domain. These devices switch labeled packets inside the MPLS domain. In the core of the network, LSRs ignore the packet network layer (IP) header and simply forward the packet using the MPLS label swapping mechanism.

A single LSP is unidirectional. In common practice, because the bidirectional nature of most traffic flows is implied, the term LSP often is used to define the pair of LSPs that enable the bidirectional flow. For ease of terminology and discussion however, the LSP in this chapter is referred to as a single entity.

LSP establishment

Prior to packet forwarding, the LSP must be established. In order to do so, labels need to be distributed for the path. Labels are usually distributed by a downstream router in the upstream direction (relative to the data flow). There are a number of ways used for label distribution: static, LDP, and RSVP. For static P2P LSPs, see chapter [Static Point-to-Point LSPs](#); for LDP P2P LSPs, see chapter [LDP Point-to-Point LSPs](#).

RSVP-TE (RFC 3209, *RSVP-TE: Extensions to RSVP for LSP Tunnels*) can be used to signal LSPs across the network. RSVP-TE is used for traffic engineering when the ingress router creates an LSP with specific constraints beyond the best route chosen by the IGP. RSVP-TE identifies the specific path desired for the LSP and may include resource requirements for the path.

The most important benefit of the label swapping mechanism RSVP-TE is its ability to map any type of user traffic to an LSP that has been specifically engineered to satisfy user traffic requirements. Customized LSPs may be created based on hop count, bandwidth requirements, administrative groups, or Shared Risk Link Groups (SRLGs). They can even be routed through a strict path with specific network links or nodes, as specified by the ingress node. This offers service providers precise control over the flow of traffic in their networks and results in a network that operates more efficiently and provides more predictable and scalable services. For information about SRLG, see chapter [Shared Risk Link Groups for RSVP-Based LSPs](#).

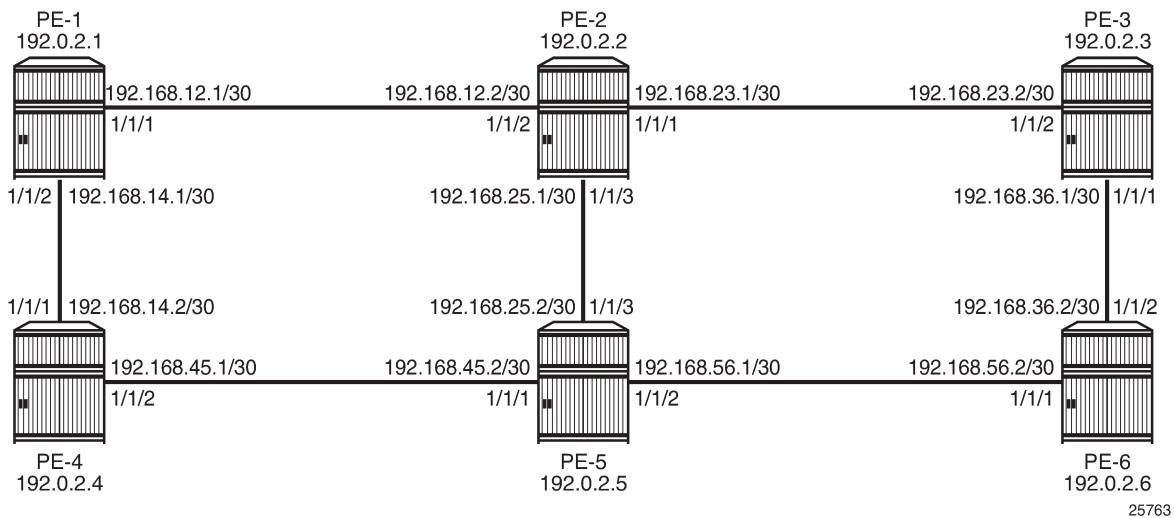
Fast reroute (FRR) allows to signal backup paths before a failure takes place. This allows traffic to flow almost continuously, without waiting for routing protocol convergence. Two different FRR methods exist for RSVP-TE LSPs: one-to-one and facility.

- FRR one-to-one defines detour tunnels toward the eLER for a particular LSP only. The advantage is that the detour tunnel is the best path to the eLER that avoids the node or link at the point of failure. The drawback is that when different LSPs would need the same detour, a dedicated RSVP-TE detour LSP needs to be signaled for each LSP.
- FRR facility defines local repair tunnels avoiding one particular node (the next hop in the data path) or one particular link (the next link in the data path), ignoring the eLER. These bypass tunnels originate in a point of local repair (PLR) and terminate in a merge point (MP) on the LSP. Bypass tunnels are shared between LSPs.

Example topology

[Figure 266: MPLS example topology](#) shows the example topology consisting of six SR OS nodes located in a single autonomous system.

Figure 266: MPLS example topology



Configuration

In this chapter, RSVP LSPs are configured manually, but they can also be configured automatically using LSP templates; see chapter [Automatic Creation of RSVP-TE LSPs](#).

As a general prerequisite for the configuration of MPLS LSPs, a correctly working Interior Gateway Protocol (IGP) is required. Open Shortest Path First (OSPF) or Intermediate System to Intermediate System (IS-IS) can be used as IGP.

RSVP-TE, an extension of the original RSVP protocol, has two major benefits adding to the basic MPLS functionality. The first benefit is traffic engineering, which allows the ingress router to create an LSP with specific constraints beyond the best route chosen by the IGP. The second benefit is improved network resiliency when a link or node fails in the network, using FRR and secondary paths. FRR is also supported for LDP, see chapter [MPLS LDP FRR using ISIS as IGP](#).

In this chapter, several RSVP-TE LSPs are configured:

- A simple LSP with a primary path that has strict hops and no specific TE constraints
- A simple LSP with a dynamic path without any configured hops is created. Initially, there are no constraints and the actual path is calculated based on the IGP best route.
- An LSP configured with constrained shortest path first (CSPF) that will use the TE metric, even though the IGP metric can also be used
- An LSP with fast reroute (FRR) one-to-one enabled
- An LSP with FRR facility enabled
- An LSP including an admin group "blue" and an LSP excluding admin group "red"
- An LSP with a hop limit configured

There is no configuration example with bandwidth constraints configured in this chapter. See chapter "Automatic Bandwidth Adjustment" in P2P LSPs in the MPLS volume of *7450 ESS, 7750 SR, and*

7950 XRS Advanced Configuration Guide — Part I for a configuration with bandwidth constraint with or without automatic adjustment.

Initially, no traffic engineering is enabled in the ISIS context, but it will be enabled when required. For RSVP LSPs, MPLS and RSVP need to be enabled on each router and all network interfaces facing the MPLS domain. For PE-1, the following configuration is required:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      admin-state enable
      interface "int-PE-1-PE-2" {
      }
      interface "int-PE-1-PE-4" {
      }
    }
    rsvp {
      admin-state enable
      interface "int-PE-1-PE-2" {
      }
      interface "int-PE-1-PE-4" {
      }
    }
  }
}
```

Strict or loose path

On the iLER, first the definition of a path is required. A path is a sequence of MPLS routers (hops) through which the LSP using that path has to pass. It is not uniquely bound to a particular LSP; it can be used by any LSP originating in that node. A hop in a path can be strict or loose: strict or loose meaning that the LSP must take either a direct path from the previous hop router to this router (strict) or can traverse through other routers (loose). The hops not explicitly defined in the loose path definition are created by calculating the IGP shortest path. A third possibility is an empty path implying not a single node is required to be present in the LSP path and the shortest path from the IGP is used to define the LSP path. Other techniques, such as the use of admin groups or shared risk link groups, can also be used to influence the decision which hops to include in the path. Three paths will be configured, respectively:

1. Only strict hops
2. Mixed strict and loose hops
3. Empty path

To find a valid path, the last hop in the path sequence needs to be the system IP or an interface address of the terminating router (eLER). The IP addresses in the hop command can be the system IP addresses or the interface addresses of the node. However, it is recommended to use the system IP addresses with keyword **loose** because this allows more flexibility when finding new paths in failover scenarios (because the upstream node could use any of multiple paths to the system address, whereas specifying the interface address would restrict the upstream node to a single entry-point). The recommendation when using the keyword **strict** in the **hop** command context, is to use the physical link addresses.



Note:

The last hop in the path should be a system address to make it appear in the list on the 5620 SAM (service-aware manager).

```
# on PE-1:
```

```
configure {
  router "Base" {
    mpls {
      path "dyn" {
        admin-state enable
      }
      path "path-PE-1-PE-6-semiLoose" {
        admin-state enable
        hop 10 {
          ip-address 192.0.2.5
          type loose
        }
        hop 20 {
          ip-address 192.168.56.2
          type strict
        }
      }
      path "path-PE-1-PE-6-strict" {
        admin-state enable
        hop 10 {
          ip-address 192.168.12.2
          type strict
        }
        hop 20 {
          ip-address 192.168.25.2
          type strict
        }
        hop 30 {
          ip-address 192.168.56.2
          type strict
        }
      }
    }
  }
}
```

The paths can be checked with the **show router mpls path** command.

```
[/]
A:admin@PE-1# show router mpls path

=====
MPLS Path:
=====
Path Name                               Admin PathIdx
Hop Index  IP Address/SID-Label                Strict/Loose
-----
dyn                                               Up    1
no hops      n/a                          n/a

path-PE-1-PE-6-semiLoose                   Up    2
10           192.0.2.5                       Loose
20           192.168.56.2                       Strict

path-PE-1-PE-6-strict                       Up    3
10           192.168.12.2                      Strict
20           192.168.25.2                      Strict
30           192.168.56.2                      Strict

-----
Total Paths : 3
=====
```

Simple RSVP LSP with strict primary path

The configuration of a simple LSP using RSVP signaling contains at least on the iLER:

- System IP address of the terminating node (to)
- Path to the eLER (primary)
- Administratively enabled

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-6" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.6
        primary "path-PE-1-PE-6-strict" {
        }
        secondary "dyn" {
        }
      }
    }
  }
}
```

All the hops in the strict path are already defined and there is no need to look up the IGP best route. The configuration of secondary paths is optional. In case the primary path fails, the secondary path can be signaled to take over the traffic. It can even be signaled as standby while the primary path is operational for a faster switchover when the keyword standby is added, which is not the case here. The secondary path has no hops defined. The hops will be calculated based on the IGP best route. The nodes through which the LSP will pass (LSRs and eLER) require no additional configuration: enabling MPLS and RSVP on their interfaces suffices.

An overview of all LSPs configured on a particular node is given by the **show router mpls lsp** command. More details about a particular LSP can be retrieved by adding the keyword detail to the previous command.

```
[/]
A:admin@PE-1# show router mpls lsp

=====
MPLS LSPs (Originating)
=====
LSP Name          Tun   Fastfail  Adm  Opr
  To              Id     Config
-----
LSP-PE-1-PE-6    1      No        Up   Up
  192.0.2.6
-----
LSPs : 1
=====
```

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-6" detail

=====
MPLS LSPs (Originating) (Detail)
=====
Legend :
  + - Inherited
=====
-----
```

```

Type : Originating
-----
LSP Name      : LSP-PE-1-PE-6
LSP Type     : RegularLsp           LSP Tunnel ID      : 1
LSP Index    : 1                   TTM Tunnel Id     : 1
From         : 192.0.2.1
To           : 192.0.2.6
Adm State    : Up                   Oper State         : Up
LSP Up Time  : 0d 00:00:37         LSP Down Time     : 0d 00:00:00
Transitions  : 1                   Path Changes      : 1
Retry Limit  : 0                   Retry Timer       : 30 sec
Signaling    : RSVP                Resv. Style       : SE
Hop Limit    : 255                 Negotiated MTU    : 1564
Adaptive     : Enabled              ClassType         : 0
FastReroute  : Disabled            Oper FR           : Disabled
PathCompMethod : none              ADSPEC            : Disabled
FallbkPathComp : not-applicable
Metric       : N/A
Load Bal Wt  : N/A                 ClassForwarding   : Disabled
Include Grps :                      Exclude Grps      :
None
Least Fill   : Disabled
BFD Template : None                BFD Ping Intvl   : 60
BFD Enable   : False               BFD Failure-action : None
WaitForUpTimer : 4

Revert Timer : Disabled            Next Revert In   : N/A
Entropy Label : Enabled+           Oper Entropy Label : Enabled
Negotiated EL : Disabled
Auto BW      : Disabled
LdpOverRsvp : Enabled
VprnAutoBind : Enabled
IGP Shortcut : Enabled             BGP Shortcut     : Enabled
IGP LFA      : Disabled            IGP Rel Metric   : Disabled
BGPTransTun : Enabled
Oper Metric  : 16777215
Prop Adm Grp : Disabled
PCE Report   : Disabled+
PCE Control  : Disabled
Path Profile  : None
Admin Tags   : None
Lsp Self Ping : Disabled+          Self Ping Timeouts : 0
SelfPingOAMFail* : 0

Secondary    : dyn                 Down Time        : 0d 00:00:37
Bandwidth    : 0 Mbps
Primary(a)   : path-PE-1-PE-6-strict
Up Time     : 0d 00:00:37
Bandwidth    : 0 Mbps
=====
* indicates that the corresponding row element may have been truncated.

```

In each hop (originating, transit and terminate), the RSVP sessions can be verified as follows:

```

[/]
A:admin@PE-1# show router rsvp session
=====
RSVP Sessions
=====
RSVP Session Name
  From           To           Tunnel ID   LSP ID     State

```



```

-----
LSP-PE-1-PE-6::path-PE-1-PE-6-strict
192.0.2.1          192.0.2.6          1          13824          Up
-----
Sessions : 1
=====

```

The detailed output of this command includes among others the session type (here: originate), the incoming and outgoing labels, the previous and next hop, and - for originating LSPs - also the list of hops):

```

[/]
A:admin@PE-1# show router rsvp session detail

=====
RSVP Sessions (Detailed)
=====
LSP : LSP-PE-1-PE-6::path-PE-1-PE-6-strict
-----
From          : 192.0.2.1          To          : 192.0.2.6
Tunnel ID     : 1              LSP ID     : 13824
Style        : SE              State      : Up
Session Type  : Originate
In Interface  : n/a            Out Interface : 1/1/1
In IF Name   : n/a
Out IF Name  : int-PE-1-PE-2
In Label     : n/a            Out Label   : 524287
Previous Hop : n/a            Next Hop   : 192.168.12.2
Hops         :
              192.168.12.2(S)
              -> 192.168.25.2(S)
              -> 192.168.56.2(S)
SetupPriority : 7              Hold Priority : 0
Class Type   : 0
SubGrpOrig ID : 0              SubGrpOrig Addr:
P2MP ID     : 0
FrrAvailType : N/A
FrrSrlgStrict : N/A          SrlgDisjoint : N/A

Path Recd    : 0              Path Sent   : 4
Resv Recd   : 3              Resv Sent  : 0
Summary msgs :
SPath Recd  : 0              SPath Sent  : 0
SResv Recd  : 0              SResv Sent  : 0
LSP Attr Flags : N/A
=====

```

The following RSVP LSP is in the tunnel table on PE-1:

```

[/]
A:admin@PE-1# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner   Encap TunnelId  Pref  Nexthop      Metric
Color
-----
192.0.2.6/32     rsvp   MPLS  1           7     192.168.12.2  16777215
-----
Flags: B = BGP or MPLS backup hop available

```

```
L = Loop-Free Alternate (LFA) hop available  
E = Inactive best-external BGP route  
k = RIB-API or Forwarding Policy backup hop  
=====
```

In order to signal PHP with RSVP, implicit-null must be configured on the eLER.

```
# on PE-6:  
configure {  
  router "Base" {  
    rsvp {  
      implicit-null-label true
```

The implicit-null is signaled and is shown on PE-5 as an egress label of 3. This label is not actually sent toward PE-6.

```
[/]  
A:admin@PE-5# show router rsvp session detail  
  
=====
```

RSVP Sessions (Detailed)			

LSP : LSP-PE-1-PE-6::path-PE-1-PE-6-strict			

From	: 192.0.2.1	To	: 192.0.2.6
Tunnel ID	: 1	LSP ID	: 13828
Style	: SE	State	: Up
Session Type	: Transit		
In Interface	: 1/1/3	Out Interface	: 1/1/2
In IF Name	: int-PE-5-PE-2		
Out IF Name	: int-PE-5-PE-6		
In Label	: 524287	Out Label	: 3
Previous Hop	: 192.168.25.1	Next Hop	: 192.168.56.2
---snip---			

The use of implicit-null can also be enabled/disabled on a per interface basis:

```
# on PE-6:  
configure {  
  router "Base" {  
    rsvp {  
      interface "int-PE-6-PE-5"  
        implicit-null-label true
```

In the remainder of the chapter, LSPs with empty paths will be used. LSP "LSP-PE-1-PE-6" is administratively disabled.

Simple RSVP LSP with dynamic path

In this section, an LSP is configured from PE-1 to PE-3 with a dynamic path that is empty. There is no secondary path. LSP "LSP-PE-1-PE-3" is configured as follows:

```
# on PE-1:  
configure {  
  router "Base" {  
    mpls {  
      lsp "LSP-PE-1-PE-3" {
```

```

admin-state enable
type p2p-rsvp
to 192.0.2.3
primary "dyn" {
}

```

Interfaces with a lower metric will be preferred over links with a high metric. The default IGP metric in this example is 10. The metric is lower for higher speed links, but can be configured manually; as follows:

```

# on PE-1:
configure {
  router "Base" {
    isis 0 {
      interface "int-PE-1-PE-2" {
        level 1 {
          metric 1000
        }
      }
    }
  }
}

```

The link between PE-1 and PE-2 has a higher metric and will not be selected for forwarding traffic because the route via PE-4 has a lower metric. The routing table shows that the route to prefix 192.0.2.3 has PE-4 as next hop instead of PE-2 and that the metric is 40:

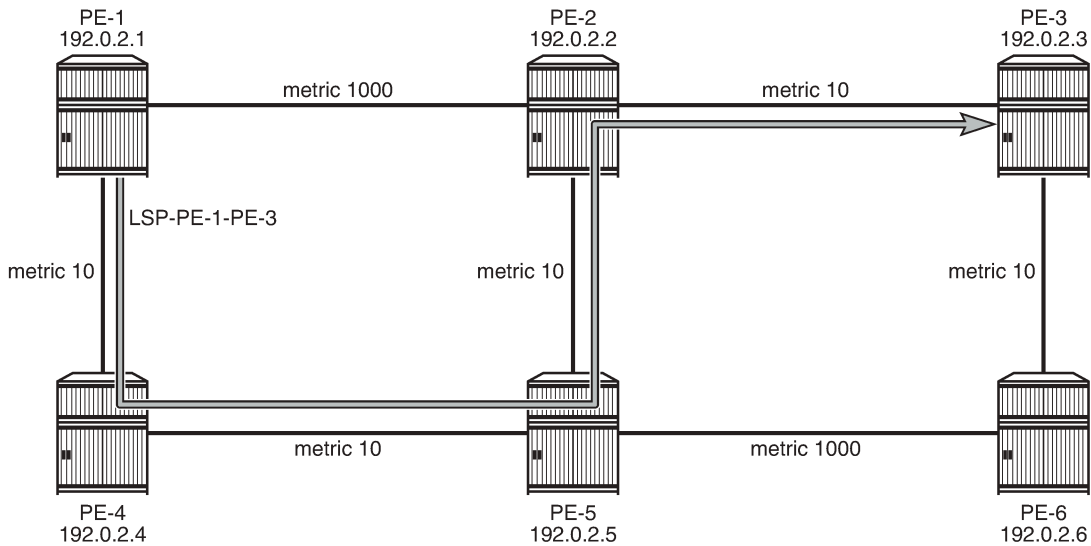
```

[/]
A:admin@PE-1# show router route-table 192.0.2.3
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]                Metric
-----
192.0.2.3/32                      Remote ISIS   00h02m45s  15
  192.168.14.2                      40
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====

```

Figure 267: LSP with dynamic path takes IGP best route shows the path used by the LSP:

Figure 267: LSP with dynamic path takes IGP best route



25765

The actual hops can be verified in the following output. The path is dynamic, therefore, no explicit hops are configured.

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-3" path detail

=====
MPLS LSP LSP-PE-1-PE-3 Path (Detail)
=====
Legend :
@ - Detour Available          # - Detour In Use
b - Bandwidth Protected      n - Node Protected
s - Soft Preemption          L - Loose
S - Strict                   + - Inherited
A - ABR

-----
LSP LSP-PE-1-PE-3
Path dyn
-----
LSP Name      : LSP-PE-1-PE-3
From          : 192.0.2.1
To            : 192.0.2.3
Admin State   : Up
Path Name     : dyn
Path LSP ID   : 45056
Path Admin    : Up
Out Interface : 1/1/2
Oper State    : Up
Path Type     : Primary
Path Oper     : Up
Out Label     : 524287
---snip---

Explicit Hops :
No Hops Specified
Actual Hops   :
192.168.14.1(192.0.2.1) Record Label : N/A
-> 192.168.14.2(192.0.2.4) Record Label : 524287
-> 192.168.45.2(192.0.2.5) Record Label : 524287
-> 192.168.25.1(192.0.2.2) Record Label : 524287
```

```
-> 192.168.23.2(192.0.2.3)          Record Label      : 524287
Resignal Eligible: False
Last Resignal      : n/a           CSPF Metric        : 0
=====
```

The tunnel table shows the RSVP LSP with PE-4 as the next hop and a metric of 40:

```
[/]
A:admin@PE-1# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.3/32         rsvp      MPLS  2          7    192.168.14.2  40
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
```

RSVP-TE LSP with dynamic path

Traffic engineering is enabled in the ISIS context on all nodes; as follows:

```
# on all PEs:
configure {
  router "Base" {
    isis 0 {
      traffic-engineering
    }
  }
}
```

The LSP can be configured with constrained shortest path first (CSPF); as follows:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3" {
        path-computation-method local-cspf
      }
    }
  }
}
```

For this LSP, it will not make any difference. By default, the IGP metrics are used and the LSP path takes the IGP shortest path.

Besides IGP metrics, also TE metrics can be configured; as follows:

```
# on PE-2:
configure {
  router "Base" {
    mpls {
      interface "int-PE-2-PE-1" {
        te-metric 10
      }
      interface "int-PE-2-PE-3" {
        te-metric 500
      }
      interface "int-PE-2-PE-5" {

```

```
te-metric 10
}
```

In this example, all interfaces on all PEs get a TE metric of 10, except for the interfaces between PE-1 and PE-4, which get a TE metric of 100 and the interfaces between PE-2 and PE-3, which get a TE metric of 500. Even with these TE metrics configured, the LSP path will not change, because the IGP metric is used by default, as can be verified as follows:

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-3" path detail

=====
MPLS LSP LSP-PE-1-PE-3 Path (Detail)
=====
Legend :
@ - Detour Available          # - Detour In Use
b - Bandwidth Protected      n - Node Protected
s - Soft Preemption
S - Strict                    L - Loose
A - ABR                       + - Inherited
=====
-----
LSP LSP-PE-1-PE-3
Path dyn
-----
LSP Name      : LSP-PE-1-PE-3
From          : 192.0.2.1
To            : 192.0.2.3
Admin State   : Up
Path Name     : dyn
Path LSP ID   : 45058
Path Admin    : Up
Out Interface : 1/1/2
Out Label     : 524286
---snip---

Adaptive      : Enabled
Preference    : n/a
Path Trans    : 2
Failure Code   : noError
Failure Node  : n/a
Explicit Hops :
  No Hops Specified
Actual Hops   :
  192.168.14.1(192.0.2.1)
  -> 192.168.14.2(192.0.2.4)
  -> 192.168.45.2(192.0.2.5)
  -> 192.168.25.1(192.0.2.2)
  -> 192.168.23.2(192.0.2.3)
Computed Hops :
  192.168.14.1(S)
  -> 192.168.14.2(S)
  -> 192.168.45.2(S)
  -> 192.168.25.1(S)
  -> 192.168.23.2(S)
Resignal Eligible: False
Last Resignal : n/a
CSPF Metric   : 40
---snip---
```

The RSVP LSP in the tunnel table has next hop PE-4 and a metric of 40:

```
[/]
A:admin@PE-1# show router tunnel-table
```

```

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.3/32         rsvp      MPLS  2         7    192.168.14.2  40
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
    
```

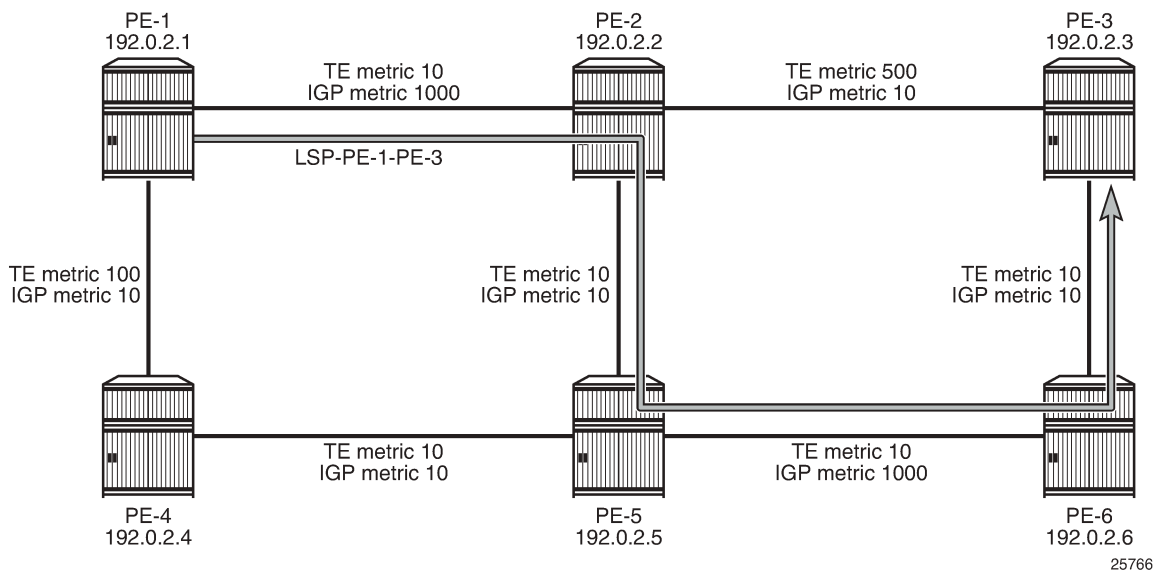
To force the LSP to use the TE metric, the LSP is reconfigured as follows:

```

# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3" {
        path-computation-method local-cspf
        metric-type te
      }
    }
  }
}
    
```

The LSP path is shown in [Figure 268: RSVP-TE LSP with dynamic path using TE metric](#):

Figure 268: RSVP-TE LSP with dynamic path using TE metric



The LSP path goes from PE-1 to PE-2 and via PE-5 and PE-6 to PE-3, as can be seen in the following output. The CSPF metric is 40, which corresponds to the TE metric in this case:

```

[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-3" path detail
    
```

```

=====
MPLS LSP LSP-PE-1-PE-3 Path (Detail)
=====
    
```

```

Legend :
  @ - Detour Available          # - Detour In Use
  b - Bandwidth Protected      n - Node Protected
  s - Soft Preemption
  S - Strict                    L - Loose
  A - ABR                       + - Inherited
=====
-----
LSP LSP-PE-1-PE-3
Path dyn
-----
LSP Name      : LSP-PE-1-PE-3
From          : 192.0.2.1
To            : 192.0.2.3
Admin State   : Up                Oper State    : Up
Path Name     : dyn
Path LSP ID   : 45060             Path Type     : Primary
Path Admin    : Up                Path Oper     : Up
Out Interface : 1/1/1             Out Label     : 524287
---snip---

Adaptive      : Enabled           Oper Metric   : 16777215
Preference    : n/a
Path Trans    : 4                 CSPF Queries  : 2
Failure Code  : noError
Failure Node  : n/a
Explicit Hops :
  No Hops Specified
Actual Hops   :
  192.168.12.1(192.0.2.1)         Record Label  : N/A
  -> 192.168.12.2(192.0.2.2)     Record Label  : 524287
  -> 192.168.25.2(192.0.2.5)     Record Label  : 524287
  -> 192.168.56.2(192.0.2.6)     Record Label  : 524287
  -> 192.168.36.1(192.0.2.3)     Record Label  : 524287
Computed Hops :
  192.168.12.1(S)
  -> 192.168.12.2(S)
  -> 192.168.25.2(S)
  -> 192.168.56.2(S)
  -> 192.168.36.1(S)
Resignal Eligible: False
Last Resignal  : n/a                CSPF Metric   : 40
=====

```

The tunnel table shows the RSVP LSP with next hop PE-2 and a metric of 16777215 (infinity) because the IGP metric is not used:

```

[/]
A:admin@PE-1# show router tunnel-table
=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.3/32     rsvp      MPLS  2          7      192.168.12.2  16777215
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====

```


The IGP metric values are restored to their default value on all interfaces on all PEs. The TE metric is configured with a value of 10 on all interfaces on all PEs.

On PE-1, the configuration is as follows:

```
# on PE-1:
configure {
  router "Base" {
    isis 0 {
      interface "int-PE-1-PE-2" {
        level 1 {
          delete metric
        }
      }
    }
  }
  mpls {
    interface "int-PE-1-PE-4" {
      te-metric 10
    }
  }
}
```

When all metrics have the same value, it does not matter whether CSPF uses the IGP or TE metric. CSPF will use the IGP metric after the following command is executed.

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3" {
        path-computation-method local-cspf
        delete metric-type          # default metric-type = igp
      }
    }
  }
}
```

The primary path will go from PE-1 to PE-2 and then to PE-3 with a CSPF (IGP) metric of 20.

Fast reroute for RSVP-TE LSPs

It is mandatory to have CSPF enabled for FRR.

Fast reroute can be configured on the RSVP LSP in two ways:

1. One-to-one: for each potential point of failure, the best detour tunnel to the eLER is signaled. This detour tunnel is signaled for this particular LSP only and cannot be shared among LSPs
2. Facility: local bypass tunnels are signaled from each point of local repair avoiding the next link or the next node. The bypass tunnels can be shared among LSPs.

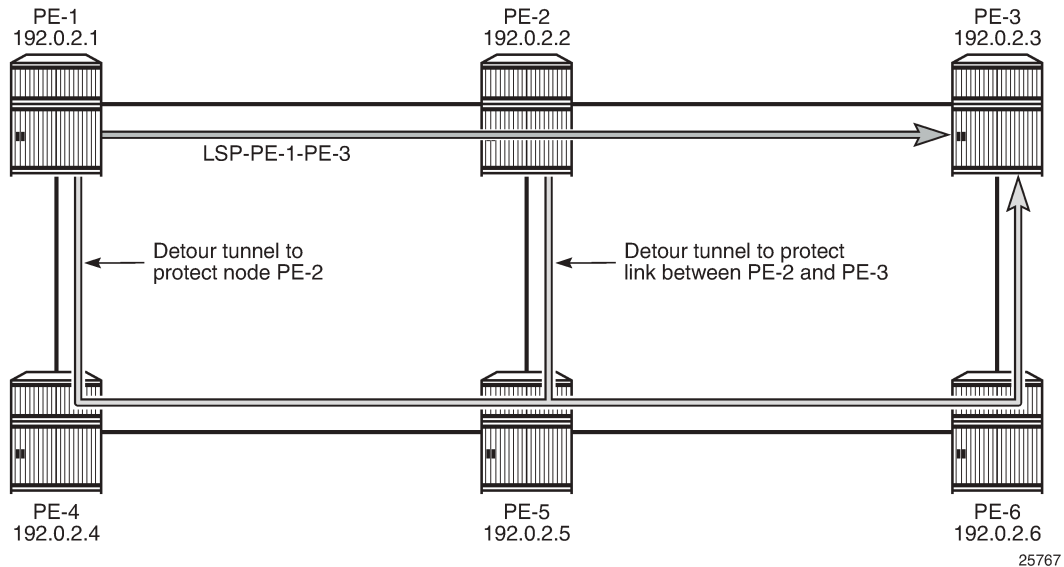
FRR one-to-one

The LSP "LSP-PE-1-PE-3" is configured with FRR one-to-one; as follows:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3" {
        fast-reroute {
          frr-method one-to-one
        }
      }
    }
  }
}
```

The preferred path from PE-1 to PE-3 is via PE-2. There will be two detour tunnels: one originating in PE-1 to protect node PE-2, and a second detour tunnel originating in PE-2 to protect the link between PE-2 and PE-3. Both detour tunnels use the same path from PE-5 to PE-3 and there is no need to signal this path twice. One detour tunnel terminates in PE-5, and the diverted traffic in this tunnel will be sent to PE-6 and PE-3 via the established detour tunnel. Depending on which detour tunnel is established first, the other detour tunnel terminates in PE-5. The preferred tunnel and the detour tunnels are shown in [Figure 269: Fast reroute one-to-one detour tunnels](#):

Figure 269: Fast reroute one-to-one detour tunnels



The protection can be seen in the list of actual hops in the path. In PE-1, a detour tunnel for node protection originates (indicated by @ n; see legend) and in PE-2 a detour tunnel for link protection (indicated by @):

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-3" path detail

=====
MPLS LSP LSP-PE-1-PE-3 Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use
  b - Bandwidth Protected      n - Node Protected
  s - Soft Preemption
  S - Strict                   L - Loose
  A - ABR                      + - Inherited
=====

LSP LSP-PE-1-PE-3
Path dyn
-----
LSP Name      : LSP-PE-1-PE-3
From          : 192.0.2.1
To           : 192.0.2.3
Admin State   : Up
Oper State    : Up
Path Name     : dyn
Path LSP ID   : 45064
Path Type     : Primary
```

```

Path Admin      : Up
Out Interface   : 1/1/1
---snip---

FRR             : Enabled
FRR NodeProtect : Enabled
FR Hop Limit   : 16
---snip---

Path Oper      : Up
Out Label     : 524287

Oper FRR      : Enabled
Oper FRR NP   : Enabled
Oper FRHopLimit : 16

Actual Hops    :
  192.168.12.1(192.0.2.1) @ n
-> 192.168.12.2(192.0.2.2) @
-> 192.168.23.2(192.0.2.3)
---snip---

Record Label   : N/A
Record Label   : 524287
Record Label   : 524287

Detour Status  : Standby
Detour Avoid Nod* : 192.0.2.2
Setup Priority  : 7
Class Type     : 0
Detour Active Ti* : n/a
In Interface   : n/a
Out Interface  : 1/1/2
NextHop       : 192.168.14.2
Explicit Hops  :
  192.168.14.1(S)
-> 192.168.14.2(S)
-> 192.168.45.2(S)
-> 192.168.56.2(S)
-> 192.168.36.1(S)

Detour Type    : Originate
Detour Origin  : 192.0.2.1
Hold Priority   : 0
Detour Up Time : 0d 00:02:55
In Label      : n/a
Out Label     : 524287
=====
* indicates that the corresponding row element may have been truncated.

```

The output also contains information about the detour tunnel originating in PE-1 that protects node PE-2. Because the detour tunnel is dedicated for this LSP, that information can be included in the LSP information.

The RSVP detour sessions can be retrieved in the originating, transit, and terminating nodes. On originating node PE-1:

```

[/]
A:admin@PE-1# show router rsvp session detour

=====
RSVP Sessions
=====
RSVP Session Name
From           To           Tunnel ID   LSP ID     State
-----
LSP-PE-1-PE-3::dyn_detour
192.0.2.1     192.0.2.3     2           45064      Up
-----
Sessions : 1
=====

```

In the transit/terminating node PE-5:

```

[/]
A:admin@PE-5# show router rsvp session detour-transit

=====
RSVP Sessions
=====

```

```

=====
RSVP Session Name
  From           To           Tunnel ID   LSP ID     State
-----
LSP-PE-1-PE-3::dyn_detour
192.0.2.1       192.0.2.3       2           45064      Up
-----
Sessions : 1
=====

```

```

[/]
A:admin@PE-5# show router rsvp session detour-terminate

```

```

=====
RSVP Sessions
=====
RSVP Session Name
  From           To           Tunnel ID   LSP ID     State
-----
LSP-PE-1-PE-3::dyn_detour
192.0.2.1       192.0.2.3       2           45064      Up
-----
Sessions : 1
=====

```

More detailed information can be retrieved as follows:

```

[/]
A:admin@PE-5# show router rsvp session detail
=====
RSVP Sessions (Detailed)
=====
LSP : LSP-PE-1-PE-3::dyn_detour
-----
From           : 192.0.2.1           To           : 192.0.2.3
Tunnel ID      : 2                       LSP ID       : 45064
Style          : SE                       State        : Up
Session Type : Transit (Detour)
In Interface   : 1/1/1             Out Interface : 1/1/2
In IF Name     : int-PE-5-PE-4
Out IF Name    : int-PE-5-PE-6
In Label       : 524287           Out Label    : 524287
Previous Hop : 192.168.45.1   Next Hop   : 192.168.56.2
---snip---

LSP : LSP-PE-1-PE-3::dyn_detour
-----
From           : 192.0.2.1           To           : 192.0.2.3
Tunnel ID      : 2                       LSP ID       : 45064
Style          : SE                       State        : Up
Session Type : Terminate (Detour)
In Interface   : 1/1/3             Out Interface : 1/1/2
In IF Name     : int-PE-5-PE-2
Out IF Name    : int-PE-5-PE-6
In Label       : 524286           Out Label    : 524287
Previous Hop : 192.168.25.1   Next Hop   : 192.168.56.2
---snip---

```

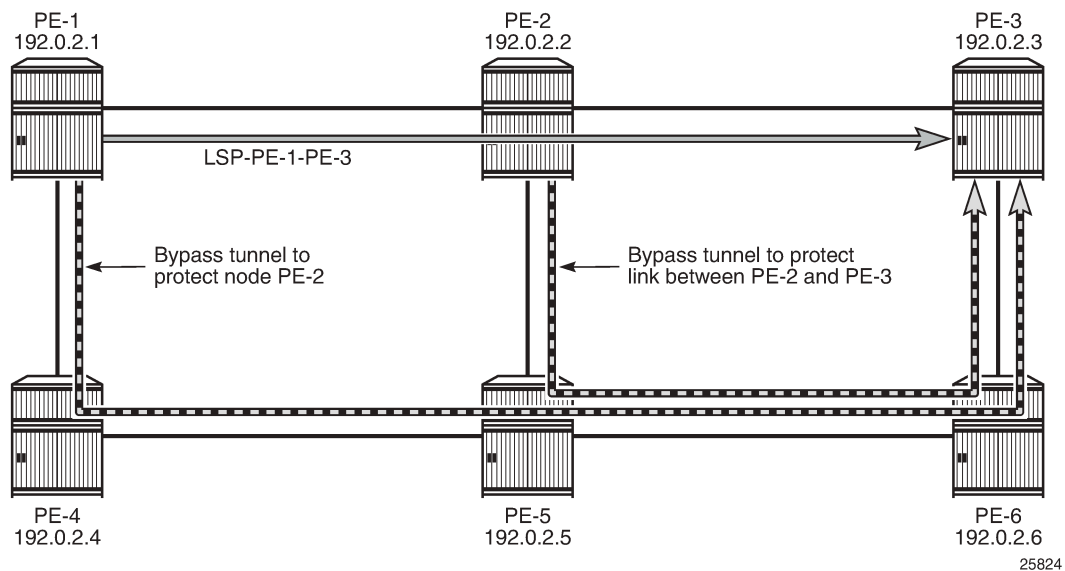
PE-5 is a transit node for the detour tunnel with previous hop PE-4 and a terminating node for the detour tunnel with previous hop PE-2. In both cases, the next hop is PE-6.

FRR facility

The drawback of FRR one-to-one is that each LSP requires its own detour tunnels to be signaled. FRR facility does not have this issue, because it offers local repair for the next node or the next link that uses bypass tunnels that can be shared by LSPs. FRR facility bypass tunnels terminate in the merge point (MP), which is a hop in the primary path. FRR facility bypass tunnels for link protection terminate in the next hop in the primary path and FRR facility bypass tunnels for node protection terminate in the next hop of that next hop. FRR bypass tunnels are unaware of the final destination of the LSP and need not terminate in the final destination, but in this case they do, because the number of hops in the primary path is limited.

[Figure 270: FRR facility bypass tunnels](#) shows the FRR facility bypass tunnels for LSP "LSP-PE-1-PE-3":

Figure 270: FRR facility bypass tunnels



Fast reroute facility is enabled on LSP "LSP-PE-1-PE-3" as follows:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-3" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.3
        path-computation-method local-cspf
        fast-reroute {
          frr-method facility
        }
        primary "dyn" {
        }
      }
    }
  }
}
```

The LSP path detail output shows that there is a bypass tunnel available in PE-1 that offers node protection for the next node in the primary path: PE-2. In PE-2, there is a bypass tunnel offering link protection for the next link, which is the link between PE-2 and PE-3; as follows:

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-3" path detail

=====
MPLS LSP LSP-PE-1-PE-3 Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use
  b - Bandwidth Protected      n - Node Protected
  s - Soft Preemption
  S - Strict                    L - Loose
  A - ABR                       + - Inherited
=====

-----
LSP LSP-PE-1-PE-3
Path dyn
-----
LSP Name      : LSP-PE-1-PE-3
From          : 192.0.2.1
To            : 192.0.2.3
Admin State   : Up                Oper State    : Up
Path Name     : dyn
Path LSP ID   : 45064             Path Type     : Primary
Path Admin    : Up                Path Oper     : Up
Out Interface : 1/1/1             Out Label     : 524287
---snip---

FRR           : Enabled           Oper FRR      : Enabled
FRR NodeProtect : Enabled         Oper FRR NP   : Enabled
FR Hop Limit  : 16                Oper FRHopLimit : 16
---snip---

Actual Hops   :
  192.168.12.1(192.0.2.1) @ n      Record Label  : N/A
  -> 192.168.12.2(192.0.2.2) @     Record Label  : 524287
  -> 192.168.23.2(192.0.2.3)      Record Label  : 524287
---snip---

FRR           : Enabled           Oper FRR      : Enabled
FRR NodeProtect : Enabled         Oper FRR NP   : Enabled
FR Hop Limit  : 16                Oper FRHopLimit : 16
---snip---
```

In FRR facility mode, the bypass tunnels are shared. They are not included in the LSP information. The bypass tunnels can be retrieved as follows:

```
[/]
A:admin@PE-1# show router mpls bypass-tunnel protected-lsp detail

=====
MPLS Bypass Tunnels (Detail)
=====
-----
bypass-node192.0.2.2-61441
-----
To          : 192.168.36.1        State         : Up
Out I/F     : 1/1/2              Out Label     : 524287
Up Time     : 0d 00:02:54        Active Time   : n/a
```

```

Reserved BW      : 0 Kbps          Protected LSP Count : 1
Type            : Dynamic         Bypass Path Cost   : 40
Setup Priority   : 7              Hold Priority       : 0
Class Type      : 0
Exclude Node    : None           Inter-Area          : False
Computed Hops   :
  192.168.14.1(S)                Egress Admin Groups : None
-> 192.168.14.2(S)                Egress Admin Groups : None
-> 192.168.45.2(S)                Egress Admin Groups : None
-> 192.168.56.2(S)                Egress Admin Groups : None
-> 192.168.36.1(S)                Egress Admin Groups : None
Actual Hops     :
  192.168.14.1(192.0.2.1)         Record Label        : N/A
-> 192.168.14.2(192.0.2.4)         Record Label        : 524287
-> 192.168.45.2(192.0.2.5)         Record Label        : 524286
-> 192.168.56.2(192.0.2.6)         Record Label        : 524286
-> 192.168.36.1(192.0.2.3)         Record Label        : 524285
Last Resignal   :
Attempted At    : n/a             Resignal Reason     : n/a
Resignal Status: n/a             Reason              : n/a

Protected LSPs -
LSP Name       : LSP-PE-1-PE-3::dyn
From           : 192.0.2.1        To                  : 192.0.2.3
Avoid Node/Hop : 192.0.2.2        Downstream Label    : 524287
Bandwidth      : 0 Kbps
=====

```

This is the bypass tunnel that originates in PE-1 to protect (avoid) PE-2. In this example, there is only one LSP protected by this bypass tunnel, but the list of protected LSPs can be longer. The same command can be launched on PE-2, where a bypass tunnel originates that protects the link between PE-2 and PE-3.

The RSVP sessions can be displayed as follows:

```

[/]
A:admin@PE-3# show router rsvp session

=====
RSVP Sessions
=====
RSVP Session Name
From           To           Tunnel ID  LSP ID    State
-----
LSP-PE-1-PE-3::dyn
192.0.2.1      192.0.2.3    2          45066     Up

bypass-link192.168.23.2-61441
192.0.2.2      192.168.36.1 61441      2          Up

bypass-node192.0.2.2-61441
192.0.2.1      192.168.36.1 61441      2          Up

-----
Sessions : 3
=====

```

In PE-3, there is an RSVP session for the regular LSP and two bypass tunnels. In this case, the bypass tunnels all go to PE-3, which is the terminating node for the LSP, but that need not be the case. All bypass tunnels are signaled from the point of local repair to the merge point on the LSP path.

To force a FRR facility switchover to a bypass tunnel, a failure is simulated by disabling port 1/1/1 on PE-2, as follows:

```
# on PE-2:
configure {
  port 1/1/1 {
    admin-state disable
```

The detailed output for the LSP path on PE-1 shows that the tunnel is locally repaired.

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-3" path detail

=====
MPLS LSP LSP-PE-1-PE-3 Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use
  b - Bandwidth Protected      n - Node Protected
  s - Soft Preemption
  S - Strict                    L - Loose
  A - ABR                       + - Inherited
=====

LSP LSP-PE-1-PE-3
Path dyn
-----
LSP Name      : LSP-PE-1-PE-3
From          : 192.0.2.1
To           : 192.0.2.3
Admin State   : Up                Oper State      : Up
Path Name     : dyn
Path LSP ID   : 45066              Path Type       : Primary
Path Admin    : Up                Path Oper       : Up
Out Interface : 1/1/1              Out Label       : 524287
---snip---

Adaptive      : Enabled            Oper Metric     : 20
Preference    : n/a
Path Trans    : 8                  CSPF Queries    : 6
Failure Code : tunnelLocallyRepaired
Failure Node  : 192.0.2.2
Explicit Hops :
  No Hops Specified
Actual Hops   :
  192.168.12.1(192.0.2.1) @ n      Record Label    : N/A
  -> 192.168.12.2(192.0.2.2) @ #   Record Label    : 524287
  -> 192.168.36.1(192.0.2.3)      Record Label    : 524287
---snip---
```

The failure code is tunnelLocallyRepaired and next to the actual hop 192.168.12.2 (PE-2), the symbol # indicates that the detour is in use.

FRR facility without node protection

Node protection is by default enabled, but can be disabled as follows:

```
# on PE-1:
configure {
```



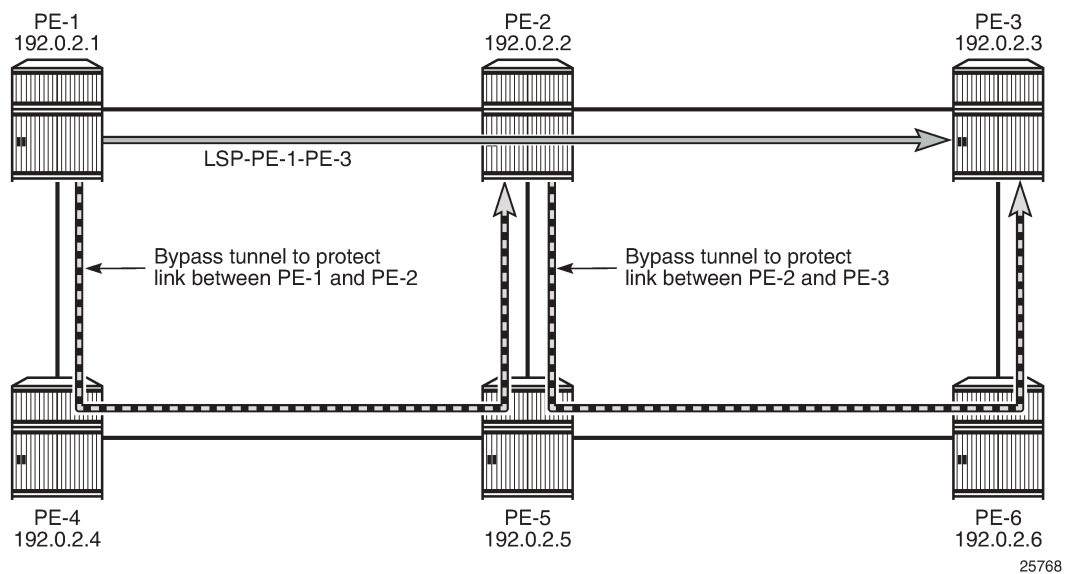
```

router "Base" {
  mpls {
    lsp "LSP-PE-1-PE-3" {
      admin-state enable
      type p2p-rsvp
      to 192.0.2.3
      fast-reroute {
        frr-method facility
        node-protect false
      }
      primary "dyn" {
      }
    }
  }
}

```

As a result, there is only link protection. The bypass tunnels from PE-1 and PE-2 terminate in the next hop in the primary path, as shown in [Figure 271: FRR facility without node protection](#):

Figure 271: FRR facility without node protection



The LSP path detail output shows that there is no node protection:

```

[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-3" path detail

=====
MPLS LSP LSP-PE-1-PE-3 Path (Detail)
=====
Legend :
@ - Detour Available          # - Detour In Use
b - Bandwidth Protected      n - Node Protected
s - Soft Preemption
S - Strict                   L - Loose
A - ABR                      + - Inherited

-----
LSP LSP-PE-1-PE-3
Path dyn
-----
LSP Name      : LSP-PE-1-PE-3

```

```

From          : 192.0.2.1
To            : 192.0.2.3
Admin State   : Up                Oper State    : Up
Path Name     : dyn
Path LSP ID   : 45070             Path Type     : Primary
Path Admin    : Up                Path Oper     : Up
Out Interface : 1/1/1             Out Label     : 524283
---snip---

FRR           : Enabled           Oper FRR      : Enabled
FRR NodeProtect : Disabled       Oper FRR NP   : Disabled
---snip---

Actual Hops   :
  192.168.12.1(192.0.2.1) @      Record Label  : N/A
  -> 192.168.12.2(192.0.2.2) @    Record Label  : 524283
  -> 192.168.23.2(192.0.2.3)     Record Label  : 524285
---snip---

```

The bypass tunnel originating in PE-1 is now terminating in PE-2 instead of PE-3; as follows:

```

[/]
A:admin@PE-1# show router mpls bypass-tunnel protected-lsp detail

=====
MPLS Bypass Tunnels (Detail)
=====
-----
bypass-link192.168.12.2-61443
-----
To          : 192.168.25.1          State         : Up
Out I/F     : 1/1/2                Out Label    : 524287
Up Time     : 0d 00:02:14          Active Time   : n/a
Reserved BW : 0 Kbps               Protected LSP Count : 1
Type        : Dynamic              Bypass Path Cost : 30
Setup Priority : 7                  Hold Priority  : 0
Class Type  : 0
Exclude Node : None                Inter-Area    : False
Computed Hops :
  192.168.14.1(S)                  Egress Admin Groups : None
  -> 192.168.14.2(S)                Egress Admin Groups : None
  -> 192.168.45.2(S)                Egress Admin Groups : None
  -> 192.168.25.1(S)                Egress Admin Groups : None
Actual Hops :
  192.168.14.1(192.0.2.1)          Record Label  : N/A
  -> 192.168.14.2(192.0.2.4)        Record Label  : 524287
  -> 192.168.45.2(192.0.2.5)        Record Label  : 524282
  -> 192.168.25.1(192.0.2.2)        Record Label  : 524282
Last Resignal :
Attempted At : n/a                  Resignal Reason : n/a
Resignal Status: n/a                Reason          : n/a
Protected LSPs -
LSP Name     : LSP-PE-1-PE-3::dyn
From          : 192.0.2.1            To            : 192.0.2.3
Avoid Node/Hop : 192.168.12.2      Downstream Label : 524283
Bandwidth     : 0 Kbps
=====

```

In the remainder of this chapter, this LSP is no longer used. Therefore, the LSP is disabled, as follows:

```

# on PE-1:
configure {

```

```
router "Base" {  
  mpls {  
    lsp "LSP-PE-1-PE-3" {  
      admin-state disable  
    }  
  }  
}
```

Administrative groups for RSVP-TE LSPs

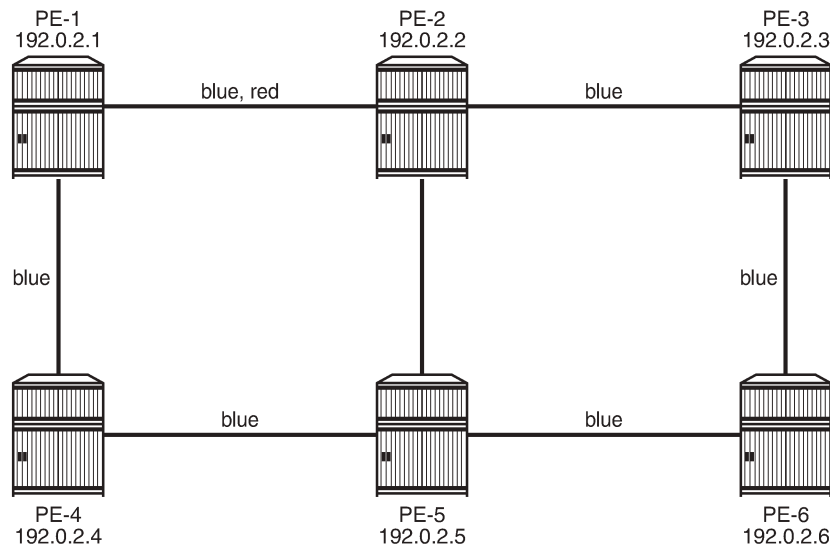
Administrative groups (link-coloring) can be used to calculate a path with the restriction to only include links of a particular admin group (color) or to exclude links of a particular admin group. Paths can be disjointed from each other, without the need for an explicit hops list.

Two admin groups are configured on all nodes; as follows:

```
# on all nodes:  
configure {  
  routing-options {  
    if-attribute {  
      admin-group "blue" {  
        value 1  
      }  
      admin-group "red" {  
        value 0  
      }  
    }  
  }  
}
```

Admin group "blue" is assigned to all MPLS interfaces, except for the link between PE-2 and PE-5 while admin group "red" is only assigned to the link between PE-1 and PE-2; see [Figure 272: Admin groups 'blue' and 'red'](#):

Figure 272: Admin groups 'blue' and 'red'



25825

The admin groups are assigned to the MPLS interfaces as follows:

```
# on PE-1:  
configure {
```

```
router "Base" {
  mpls {
    interface "int-PE-1-PE-2" {
      admin-group ["blue" "red"]
    }
    interface "int-PE-1-PE-4" {
      admin-group ["blue"]
    }
  }
}
```

The configuration on the other nodes is similar.

To ensure that FRR bypass tunnels will adhere to the same admin group constraints as defined in the LSP, the following is configured on all nodes. It is required on all Points of Local Repair (PLRs):

```
# on all nodes (at least on all PLRs):
configure {
  router "Base" {
    mpls {
      admin-group-frr true
    }
  }
}
```

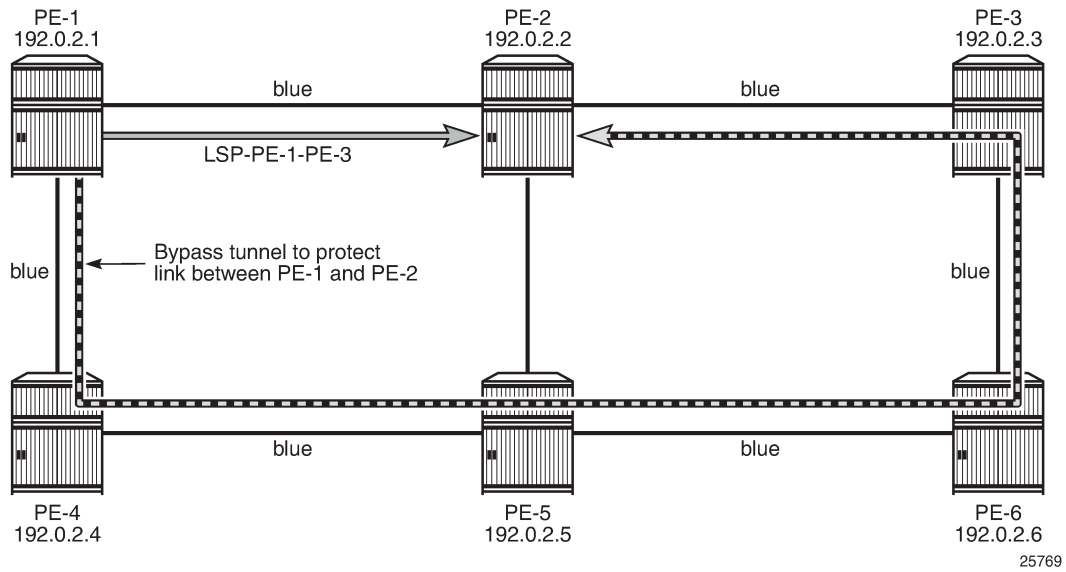
LSP includes admin group 'blue'

LSP "LSP-PE-1-PE-2" is created on PE-1 with a dynamic primary path. FRR facility is enabled. The LSP includes admin group blue and both the primary path as the bypass tunnel must use links in admin group "blue" (**propagate-admin-group**). **Admin-group-frr** is enabled in the **mpls** context, to ensure that the admin group restriction is respected for FRR.

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      admin-group-frr true
      lsp "LSP-PE-1-PE-2" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.2
        propagate-admin-group true
        path-computation-method local-cspf
        include-admin-group ["blue"]
        fast-reroute {
          propagate-admin-group true
          frr-method facility
        }
        primary "dyn" {
        }
      }
    }
  }
}
```

The bypass tunnel cannot include the link between PE-2 and PE-5, because that link does not belong to admin group "blue". The LSP and its bypass tunnel are shown in [Figure 273: LSP and bypass within admin group 'blue'](#):

Figure 273: LSP and bypass within admin group 'blue'



The LSP path detailed information is as follows:

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-2" path detail

=====
MPLS LSP LSP-PE-1-PE-2 Path (Detail)
=====
Legend :
@ - Detour Available          # - Detour In Use
b - Bandwidth Protected      n - Node Protected
s - Soft Preemption          L - Loose
S - Strict                   + - Inherited
A - ABR

-----
LSP LSP-PE-1-PE-2
Path dyn
-----
LSP Name      : LSP-PE-1-PE-2
From          : 192.0.2.1
To            : 192.0.2.2
Admin State   : Up           Oper State    : Up
Path Name     : dyn
Path LSP ID   : 64000        Path Type     : Primary
Path Admin    : Up           Path Oper     : Up
Out Interface : 1/1/1        Out Label    : 524287
---snip---

FRR           : Enabled      Oper FRR      : Enabled
FRR NodeProtect : Enabled    Oper FRR NP   : Enabled
FR Hop Limit  : 16           Oper FRHopLimit : 16
FR Prop Admin Gr*: Enabled  Oper FRPropAdmGrp : Enabled
Propagate Adm Grp: Enabled  Oper Prop Adm Grp : Enabled
---snip---

Include Groups :              Oper IncludeGroups:
```

```

blue
Exclude Groups   :
None
---snip---

Actual Hops      :
  192.168.12.1(192.0.2.1) @
-> 192.168.12.2(192.0.2.2)
---snip---

blue
Oper ExcludeGroups:
None

Record Label    : N/A
Record Label    : 524287
    
```

There is a bypass tunnel originating in PE-1 that offers protection for the link between PE-1 and PE-2. More information about this bypass tunnel can be retrieved as follows:

```

[/]
A:admin@PE-1# show router mpls bypass-tunnel protected-lsp detail

=====
MPLS Bypass Tunnels (Detail)
=====
-----
bypass-link192.168.12.2-61444
-----
To           : 192.168.23.1      State           : Up
Out I/F      : 1/1/2           Out Label       : 524287
Up Time      : 0d 00:01:20     Active Time     : n/a
Reserved BW  : 0 Kbps         Protected LSP Count : 1
Type         : Dynamic        Bypass Path Cost : 50
Setup Priority : 7             Hold Priority    : 0
Class Type   : 0
Exclude Node : None           Inter-Area      : False
Computed Hops :
  192.168.14.1(S)           Egress Admin Groups :
-> 192.168.14.2(S)           blue
-> 192.168.45.2(S)          Egress Admin Groups :
-> 192.168.56.2(S)          blue
-> 192.168.36.1(S)          Egress Admin Groups :
-> 192.168.23.1(S)          blue
Actual Hops   :               Egress Admin Groups : None
  192.168.14.1(192.0.2.1)   Record Label       : N/A
-> 192.168.14.2(192.0.2.4)   Record Label       : 524287
-> 192.168.45.2(192.0.2.5)   Record Label       : 524287
-> 192.168.56.2(192.0.2.6)   Record Label       : 524287
-> 192.168.36.1(192.0.2.3)   Record Label       : 524287
-> 192.168.23.1(192.0.2.2)   Record Label       : 524286
Last Resignal :
Attempted At  : n/a           Resignal Reason   : n/a
Resignal Status: n/a         Reason            : n/a

Protected LSPs -
LSP Name      : LSP-PE-1-PE-2::dyn
From          : 192.0.2.1     To                : 192.0.2.2
Avoid Node/Hop : 192.168.12.2 Downstream Label  : 524287
Bandwidth     : 0 Kbps
    
```

All egress links are in admin group blue on the originating and transit nodes.

LSP excludes admin group 'red'

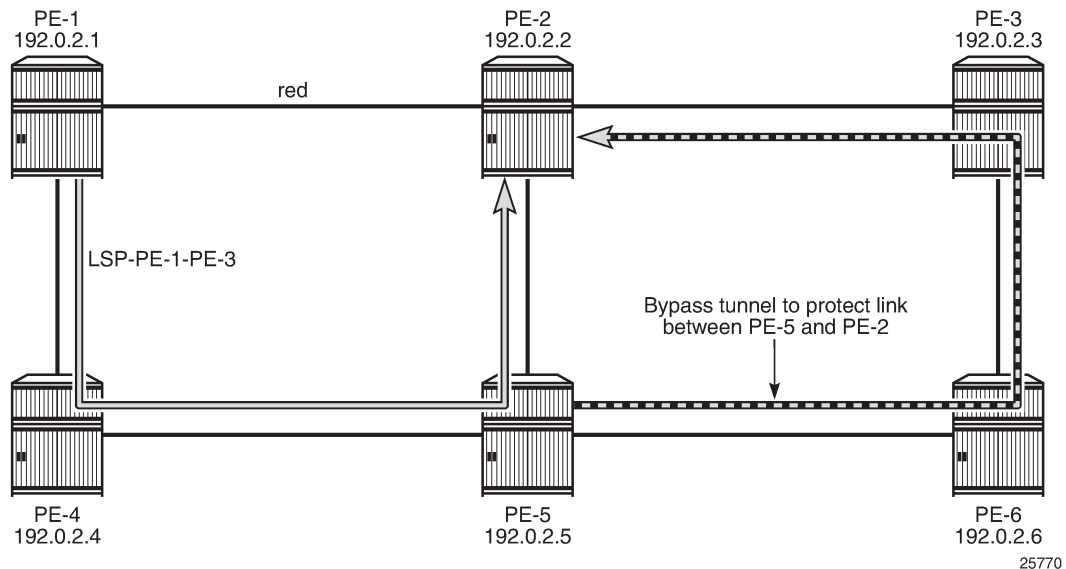
The LSP is reconfigured: instead of including admin group 'blue', it will exclude admin group 'red'. Nothing is changed to the configuration of FRR.

The MPLS configuration is modified as follows:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-2" {
        delete include-admin-group ["blue"]
        exclude-admin-group ["red"]
      }
    }
  }
}
```

The LSP cannot use the red link between PE-1 and PE-2. The path that avoids the red link, is from PE-1 via PE-4 and PE-5 to PE-2. On all PLRs, **admin-group-frr** is configured, which implies that the originating FRR bypass tunnels need to respect the admin-group constraint of the LSP. There can be no node protection for PE-4 or PE-5 without using the red link between PE-1 and PE-2. The only link that can be protected without using the red link between PE-1 and PE-2, is the link between PE-5 and PE-2. The LSP and the FRR bypass tunnel are shown in [Figure 274: LSP and FRR bypass tunnel excluding admin group 'red'](#):

Figure 274: LSP and FRR bypass tunnel excluding admin group 'red'



The LSP path can be verified as follows:

```
[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-2" path detail

=====
MPLS LSP LSP-PE-1-PE-2 Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use
```

```

b - Bandwidth Protected      n - Node Protected
s - Soft Preemption
S - Strict                  L - Loose
A - ABR                     + - Inherited
=====
-----
LSP LSP-PE-1-PE-2
Path dyn
-----
LSP Name      : LSP-PE-1-PE-2
From          : 192.0.2.1
To            : 192.0.2.2
Admin State   : Up
Path Name     : dyn
Path LSP ID   : 64002
Path Admin    : Up
Out Interface : 1/1/2
Oper State    : Up
Path Type     : Primary
Path Oper     : Up
Out Label     : 524286
---snip---

Include Groups :
None
Oper IncludeGroups:
None
Exclude Groups :
red
Oper ExcludeGroups:
red
---snip---

Actual Hops :
  192.168.14.1(192.0.2.1)      Record Label : N/A
-> 192.168.14.2(192.0.2.4)      Record Label : 524286
-> 192.168.45.2(192.0.2.5) @    Record Label : 524286
-> 192.168.25.1(192.0.2.2)      Record Label : 524285
---snip---

```

There is only link protection for the link from PE-5 to PE-2. The bypass tunnel originates in PE-5 and has no links belonging to admin group 'red':

```

[/]
A:admin@PE-5# show router mpls bypass-tunnel protected-lsp detail
=====
MPLS Bypass Tunnels (Detail)
=====
-----
bypass-link192.168.25.1-62074
-----
To          : 192.168.23.1      State          : Up
Out I/F     : 1/1/2            Out Label     : 524286
Up Time     : 0d 00:04:38      Active Time    : n/a
Reserved BW : 0 Kbps           Protected LSP Count : 1
Type        : Dynamic          Bypass Path Cost : 30
Setup Priority : 7              Hold Priority   : 0
Class Type  : 0
Exclude Node : None            Inter-Area     : False
Computed Hops :
  192.168.56.1(S)              Egress Admin Groups :
                                blue
-> 192.168.56.2(S)              Egress Admin Groups :
                                blue
-> 192.168.36.1(S)              Egress Admin Groups :
                                blue
-> 192.168.23.1(S)              Egress Admin Groups : None
Actual Hops :
  192.168.56.1(192.0.2.5)      Record Label   : N/A
-> 192.168.56.2(192.0.2.6)      Record Label   : 524286

```



```

-> 192.168.36.1(192.0.2.3)      Record Label      : 524286
-> 192.168.23.1(192.0.2.2)    Record Label      : 524284
Last Resignal      :
Attempted At      : n/a
Resignal Status   : n/a
Resignal Reason   : n/a
Reason            : n/a

Protected LSPs -
LSP Name          : LSP-PE-1-PE-2::dyn
From              : 192.0.2.1
To                : 192.0.2.2
Avoid Node/Hop    : 192.168.25.1
Bandwidth         : 0 Kbps
Downstream Label  : 524287

```

This configuration is preserved for the following example.

Hop limit for RSVP-TE LSPs

Another constraint to influence the path selection, is hop limit. This can be configured on the LSP, on a secondary path, or on FRR in case the path should not contain too many hops. In this example, it will be configured on the LSP and later also for FRR on that LSP. By default, the LSP hop limit is 255, but it can be configured as follows:

```

# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-2" {
        hop-limit 5
      }
    }
  }
}

```

This hop limit of 5 is enough for the path via PE-4 and PE-5, but it will not be sufficient when the link between PE-2 and PE-5 is down:

```

# on PE-5:
configure {
  port 1/1/3 {
    admin-state disable
  }
}

```

In this case, the only possible path that excludes the 'red' link between PE-1 and PE-2, has to go to PE-2 via PE-4, PE-5, PE-6, and PE-3. There are too many hops. The FRR bypass tunnel can do a local repair, but no new LSP path can be signaled, with failure code: noCspfRouteToDestination:

```

[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-2" path detail

```

```

=====
MPLS LSP LSP-PE-1-PE-2 Path (Detail)
=====

```

```

Legend :
@ - Detour Available          # - Detour In Use
b - Bandwidth Protected      n - Node Protected
s - Soft Preemption
S - Strict                   L - Loose
A - ABR                      + - Inherited

```

```

-----
LSP LSP-PE-1-PE-2
Path dyn

```

```

-----
LSP Name      : LSP-PE-1-PE-2
From          : 192.0.2.1
To           : 192.0.2.2
Admin State   : Up                               Oper State    : Up
Path Name    : dyn
Path LSP ID   : 64004                             Path Type     : Primary
Path Admin    : Up                               Path Oper     : Up
Out Interface : 1/1/2                             Out Label    : 524287
---snip---

Include Groups :                               Oper IncludeGroups:
None                                                  None
Exclude Groups :                               Oper ExcludeGroups:
red                                                  red

Adaptive       : Enabled                         Oper Metric   : 30
Preference     : n/a
Path Trans     : 3                               CSPF Queries  : 4
Failure Code   : tunnelLocallyRepaired
Failure Node   : 192.0.2.5
Explicit Hops  :
  No Hops Specified
Actual Hops    :
  192.168.14.1(192.0.2.1)                       Record Label  : N/A
  -> 192.168.14.2(192.0.2.4)                     Record Label  : 524287
  -> 192.168.45.2(192.0.2.5) @ #                 Record Label  : 524287
  -> 192.168.25.1(192.0.2.2)                     Record Label  : 524287
---snip---

In Prog MBB :
MBB Type      : GlobalRevert                     Next Retry In : 24 sec
Started At    : 02/13/2021 17:32:35             Retry Attempt  : 1
Failure Code  : noCspfRouteToDestination        Failure Node   : 192.0.2.1
              n
Signaled BW   : 0 Mbps
=====
* indicates that the corresponding row element may have been truncated.

```

FRR tunnels also have a hop limit. The FRR hop limit is by default 16, but can be configured as follows:

```

# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-2" {
        fast-reroute {
          hop-limit 3
        }
      }
    }
  }
}

```

When the LSP is recalculated, it is impossible to establish the primary path with a hop limit of 5 and it is also impossible to establish a bypass tunnel protecting the link between PE-5 and PE-2 when the FRR hop limit is 3. The LSP will remain operationally down with failure code: noCspfRouteToDestination:

```

[/]
A:admin@PE-1# show router mpls lsp "LSP-PE-1-PE-2" path detail

=====
MPLS LSP LSP-PE-1-PE-2 Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use

```

```

b - Bandwidth Protected      n - Node Protected
s - Soft Preemption
S - Strict                  L - Loose
A - ABR                     + - Inherited
=====
-----
LSP LSP-PE-1-PE-2
Path dyn
-----
LSP Name      : LSP-PE-1-PE-2
From          : 192.0.2.1
To            : 192.0.2.2
Admin State : Up                Oper State      : Down
Path Name     : dyn
Path LSP ID   : 64008              Path Type      : Primary
Path Admin  : Up                Path Oper    : Down
Out Interface : n/a                Out Label     : n/a
Path Up Time  : 0d 00:00:00        Path Down Time : 0d 00:00:56
---snip---

FRR           : Enabled            Oper FRR       : N/A
FRR NodeProtect : Enabled          Oper FRR NP    : N/A
FR Hop Limit  : 3                  Oper FRHopLimit : N/A
FR Prop Admin Gr* : Enabled         Oper FRPropAdmGrp : N/A
Propagate Adm Grp : Enabled         Oper Prop Adm Grp : N/A
---snip---

Neg MTU       : 0                  Oper MTU       : N/A
Bandwidth     : No Reservation     Oper Bandwidth : N/A
Hop Limit   : 5                Oper HopLimit  : N/A
---snip---

Include Groups :                    Oper IncludeGroups:
None                                                  N/A
Exclude Groups :                    Oper ExcludeGroups:
red                                                  N/A
---snip---

Failure Code      : noCspfRouteToDestination
Failure Node      : 192.0.2.1
Explicit Hops     :
  No Hops Specified
Actual Hops       :
  No Hops Specified
---snip---

```

For the remainder of the examples, FRR is disabled and the hop limit is restored to the default value, which is 255:

```

# on PE-1:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-1-PE-2" {
        delete hop-limit
        delete fast-reroute
      }
    }
  }
}

```

On PE-5, port 1/1/3 is enabled, as follows:

```

# on PE-5:
configure {
  port 1/1/3 {

```

```
admin-state enable
```

Manual resignal

Instead of waiting for the resignal timer to expire, one can manually trigger the resignal process.

The command to resignal the path "dyn" of LSP "LSP-PE-1-PE-2":

```
[/]  
A:admin@PE-1# tools perform router mpls resignal lsp "LSP-PE-1-PE-2" path "dyn"
```

The command to resignal all RSVP LSPs originating at node PE-1:

```
[/]  
A:admin@PE-1# tools perform router mpls resignal delay 0  
WARNING: CLI #2006: Warning while processing command - WARNING: CLI Delay will not be in  
effect, configure resignal-timer at config>router>mpls>resignal-timer .
```

The preceding command overrules the resignal timer in the **mpls** context, so it can only be launched after the resignal timer is configured.

```
[ex:configure router "Base" mpls]  
A:admin@PE-1# resignal-timer ?  
  
resignal-timer <number>  
<number> - <30..10080> - minutes  
  
Resignal timer for RSVP LSPs
```

The configuration timer is configured to 30 minutes as follows:

```
# on PE-1:  
configure {  
  router "Base" {  
    mpls {  
      resignal-timer 30
```

Whenever an LSP is resigned, the resignal timer is restarted.

LSP OAM

The LSP diagnostics are modeled after ICMP echo request/reply which provides a mechanism to detect data plane failures in MPLS LSPs. For a given FEC, LSP ping verifies whether the packet reaches the egress label edge router (LER). The following OAM commands are in classic CLI.

```
[/]  
A:admin@PE-1# //  
INFO: CLI #2051: Switching to the classic CLI engine  
A:PE-1# oam lsp-ping "LSP-PE-1-PE-2"  
LSP-PING LSP-PE-1-PE-2: 92 bytes MPLS payload  
Seq=1, send from intf int-PE-1-PE-4, reply from 192.0.2.2  
  udp-data-len=32 ttl=255 rtt=3.71ms rc=3 (EgressRtr)  
  
---- LSP LSP-PE-1-PE-2 PING Statistics ----  
1 packets sent, 1 packets received, 0.00% packet loss
```

```
round-trip min = 3.71ms, avg = 3.71ms, max = 3.71ms, stddev = 0.000ms
```

In LSP traceroute mode, the packet is sent to the control plane of each transit label switched router (LSR) which performs various checks to see if it is actually a transit LSR for the path.

```
A:PE-1# oam lsp-trace "LSP-PE-1-PE-2"
lsp-trace to LSP-PE-1-PE-2: 0 hops min, 0 hops max, 116 byte packets
1 192.0.2.4 rtt=2.60ms rc=8(DSRtrMatchLabel) rsc=1
2 192.0.2.5 rtt=3.82ms rc=8(DSRtrMatchLabel) rsc=1
3 192.0.2.2 rtt=3.54ms rc=3(EgressRtr) rsc=1
```

```
A:PE-1# oam lsp-trace "LSP-PE-1-PE-2" detail
lsp-trace to LSP-PE-1-PE-2: 0 hops min, 0 hops max, 116 byte packets
1 192.0.2.4 rtt=2.30ms rc=8(DSRtrMatchLabel) rsc=1
   DS 1: ipaddr=192.168.45.2 ifaddr=192.168.45.2 iftype=ipv4Numbered MRU=1564
         label[1]=524286 protocol=4(RSVP-TE)
2 192.0.2.5 rtt=3.70ms rc=8(DSRtrMatchLabel) rsc=1
   DS 1: ipaddr=192.168.25.1 ifaddr=192.168.25.1 iftype=ipv4Numbered MRU=1564
         label[1]=524286 protocol=4(RSVP-TE)
3 192.0.2.2 rtt=3.84ms rc=3(EgressRtr) rsc=1
```

RSVP LSP statistics

Statistics can be collected for RSVP LSPs. For each accounting record, a file ID is configured; as follows:

```
# on PE-1:
configure {
  log {
    file 2 {
      rollover 5
      retention 1
      compact-flash-location {
        primary cfl
      }
    }
  }
}
```

An accounting policy is configured for each record type; as follows:

```
# on PE-1:
configure {
  log {
    accounting-policy 2 {
      admin-state enable
      record combined-mpls-lsp-ingress
      destination {
        file 2
      }
    }
  }
}
```

The collection of statistics is enabled in the **mpls** context as follows:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      ingress-statistics {
        lsp sender 192.0.2.1 lsp-name "LSP-PE-1-PE-2" {
          admin-state enable
        }
      }
    }
  }
}
```

```

        collect-stats true
        accounting-policy 2
    }
}

```

To display the statistics, the following options are available for `lsp-ingress-stats`:

```

[/]
A:admin@PE-1# show router mpls lsp-ingress-stats ?

lsp-ingress-stats [type <keyword>] [active]
lsp-ingress-stats lsp <string> sender <ipv4 address>
lsp-ingress-stats [sender <ipv4 address>] [type <keyword>] [active] template-match
  SessionNameString <string>

SessionNameString    - [Max 64 chars]
active                - match on all stats enabled lsp
lsp                  - max 64 chars
sender                - ipv4 address '<d.d.d.d>'
template-match       - match on p2p/p2mp stats template
type                  - p2p|p2mp

```

The following command retrieves the LSP ingress statistics for LSP "LPS-PE-1-PE-2" with sender 192.0.2.1:

```

[/]
A:admin@ PE-1# show router mpls lsp-ingress-stats lsp "LSP-PE-1-PE-2"
                                                    sender 192.0.2.1

=====
MPLS LSP Ingress Statistics
=====
-----
LSP Name      : LSP-PE-1-PE-2
Sender       : 192.0.2.1
-----
Collect Stats : Enabled          Accting Plcy. : 2
Adm State    : Up                PSB Match     : False
FC BE
InProf Pkts : 0                  OutProf Pkts  : 0
InProf Octets : 0                OutProf Octets : 0
FC L2
InProf Pkts : 0                  OutProf Pkts  : 0
InProf Octets : 0                OutProf Octets : 0
FC AF
InProf Pkts : 0                  OutProf Pkts  : 0
InProf Octets : 0                OutProf Octets : 0
FC L1
InProf Pkts : 0                  OutProf Pkts  : 0
InProf Octets : 0                OutProf Octets : 0
FC H2
InProf Pkts : 0                  OutProf Pkts  : 0
InProf Octets : 0                OutProf Octets : 0
FC EF
InProf Pkts : 0                  OutProf Pkts  : 0
InProf Octets : 0                OutProf Octets : 0
FC H1
InProf Pkts : 0                  OutProf Pkts  : 0
InProf Octets : 0                OutProf Octets : 0
FC NC
InProf Pkts : 0                  OutProf Pkts  : 0
InProf Octets : 0                OutProf Octets : 0

```

```
Aggregate Pkts   : 0
Aggregate Octets : 0
=====
```

Statistics can be cleared as follows:

```
[clear router mpls]
A:admin@PE-1# lsp-ingress-stats 192.0.2.1 lsp "LSP-PE-1-PE-2"
```

```
[clear router mpls]
A:admin@PE-1# lsp-ingress-stats
```

Debug

A wide range of debug tools are available which can be tuned to the specific information of importance for a certain troubleshooting task. These tools are in classic CLI. In the **debug router mpls** context, the LSP object to trace or monitor can be selected by the following parameters:

- LSP name
- Source address of the LSP (the from parameter in the LSP definition)
- Termination point of the LSP (the to parameter in the LSP definition)
- Tunnel ID of the LSP
- LSP ID

```
A:PE-1# debug router rsvp ?
- no rsvp
- rsvp [lsp name>] [sender <sender-address>] [endpoint <endpoint-address>]
      [tunnel-id <tunnel-id>] [lsp-id <lsp-id>] [interface <ip-int-name>]

<name>           : [160 chars max]
<sender-address> : a.b.c.d
<endpoint-address> : a.b.c.d
<tunnel-id>      : [0..4294967295]
<lsp-id>         : [1..65535]
<ip-int-name>    : [32 chars max]

[no] event       + Enable/disable debugging for specific RSVP events
[no] packet      + Enable/disable debugging for specific RSVP packets
```

```
A:PE-1# debug router mpls ?
- mpls [lsp <name>] [sender <ip-address|ipv6-address>] [endpoint <ip-address|
      ipv6-address>] [tunnel-id <tunnel-id>] [lsp-id <lsp-id>]
- no mpls

<name>           : [160 chars max]
<ip-address|ipv6-a*> : [64 chars max]
<ip-address|ipv6-a*> : [64 chars max]
<tunnel-id>      : [0..4294967295]
<lsp-id>         : [1..65535]

[no] event       + Enable/disable debugging for specific MPLS events
[no] forwarding-pol* + Enable/disable debugging for MPLS Forwarding-Policies
```

In the **debug** command tree, the MPLS event type can be selected (tracing must be enabled):

```
A:PE-1# debug router mpls lsp "LSP-PE-1-PE-2" event ?
- event
- no event

[no] all          - Enable/disable debugging for MPLS all
[no] frr          - Enable/disable debugging for MPLS frr
[no] iom          - Enable/disable debugging for MPLS iom
[no] lsp-setup    - Enable/disable debugging for MPLS lsp setup
[no] mbb          - Enable/disable debugging for MPLS mbb
[no] misc         - Enable/disable debugging for MPLS misc
[no] pcc          - Enable/disable debugging for MPLS PCC
[no] te           - Enable/disable debugging for MPLS TE
[no] xc           - Enable/disable debugging for MPLS xc
```

As an example, the all keyword is entered, logging all MPLS events related to the selected LSP:

```
# on PE-1:
debug
  router "Base"
    mpls lsp "LSP-PE-1-PE-2"
      event
        all
```

```
A:PE-1# show debug
debug
  router "Base"
    mpls lsp "LSP-PE-1-PE-2"
      event
        iom
        lsp-setup
        xc
        frr
        mbb
        misc
        pcc
        te
      exit
    exit
  exit
exit
```

The last step is to create a log container which will gather all MPLS debugging information according to the criteria set in the debug context. The **from debug-trace** parameter must be configured but there are several options where the different captured entries will be stored: console, a syslog server, SNMP, local file on the compact flash card, a temporary circular memory buffer, or the telnet/SSH session from which you are logged into the node.

The ID of the log container is a local number without any other significance.

```
[ex:configure log log-id 2]
A:admin@PE-1# destination ?

destination

Note: 'configure log log-id 2' and all other elements in this context support
single-model management only.

Immutable fields      - file, syslog, console
```



```
Choice: log-id-destination
cli          :+ Enable the cli context
console     :- Log events to send to the system console port
file        :- Log events to send to the specified file ID
memory      :+ Enable the memory context
netconf     :+ Enable the netconf context
snmp        :+ Enable the snmp context
syslog      :- Log events to send to the specified syslog ID
```

For this example, the temporary buffer (with adjustable size) is chosen, as follows:

```
# on PE-1:
configure {
  log {
    log-id 2 {
      source {
        debug true
      }
      destination {
        memory {
        }
      }
    }
  }
}
```

All MPLS events related to the selected LSP are stored in the location (memory) specified. The content of this log container can be viewed through the **show log log-id 2** command. The following output is a subset of messages shown after port 1/1/2 on PE-2 is disabled, which causes LSP "LSP-PE-1-PE-2" to go down.

```
[/]
A:admin@PE-1# show log log-id 2 ascending

=====
Event Log 2 log-name 2
=====
Description : (Not Specified)
Memory Log contents [size=100  next event=19  (not wrapped)]

1 2021/02/13 20:16:29.034 CET MINOR: DEBUG #2001 Base MPLS
"MPLS: LSP Path
Signalling failure for LspPath LSP-PE-1-PE-2::dyn(LspId 64022)"

2 2021/02/13 20:16:29.034 CET MINOR: DEBUG #2001 Base MPLS
"MPLS: CSPF
Delete CSPF Hop list 18"

3 2021/02/13 20:16:29.034 CET MINOR: DEBUG #2001 Base MPLS
"MPLS: LSP Path
Set operational state for LspPath LSP-PE-1-PE-2::dyn(LspId 64022) to Down, previous state is
Up"

4 2021/02/13 20:16:29.034 CET MINOR: DEBUG #2001 Base MPLS
"MPLS: LSP Path
Set operational MTU for LspPath LSP-PE-1-PE-2::dyn(LspId 64022) to 0"

5 2021/02/13 20:16:29.034 CET MINOR: DEBUG #2001 Base MPLS
"MPLS: LSP Path
Set operational metric for LspPath LSP-PE-1-PE-2::dyn(LspId 64022) to 30"
---snip---
```

Conclusion

MPLS provides the capability to establish connection oriented paths over a connectionless network. The LSP offers a mechanism to engineer network traffic on constraint-based paths rather than the IGP shortest path. This can greatly improve network resiliency. In this chapter, the configuration of several RSVP LSP features is given together with the associated show output which can be used to verify and troubleshoot.

Seamless MPLS: Isolated IGP/LDP Domains and Labeled BGP

This chapter provides information about Seamless MPLS: Isolated IGP/LDP domains and Labeled BGP.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter is applicable to SR OS routers and was initially written for SR OS Release 13.0.R7. The CLI in the current edition is based on SR OS Release 23.3.R1.

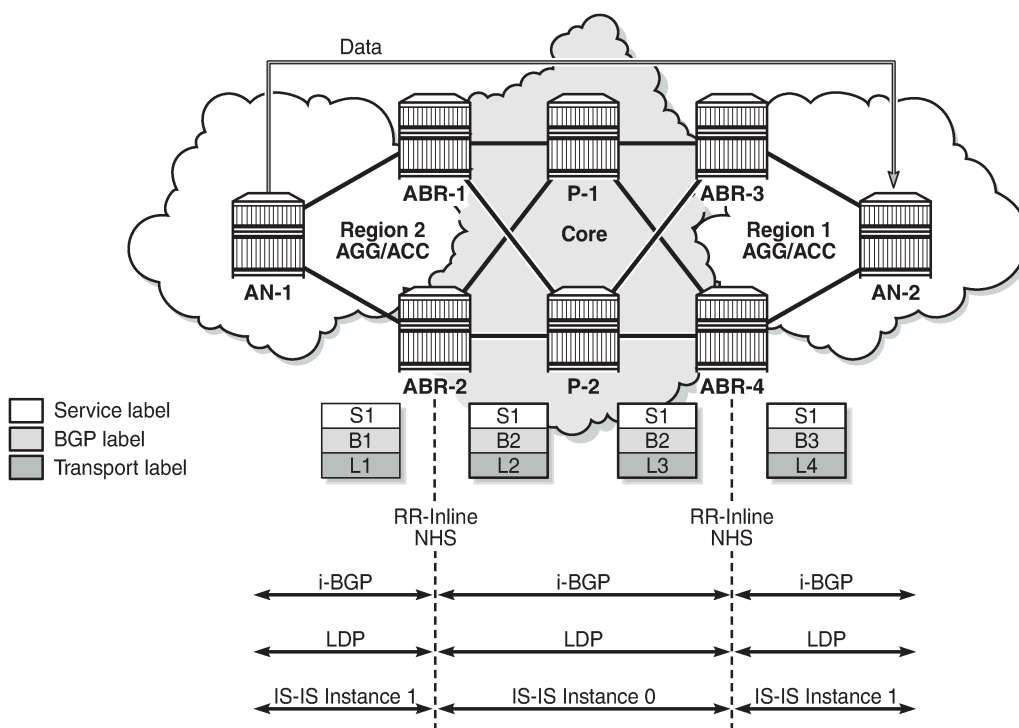
Overview

Seamless Multi-Protocol Label Switching (MPLS) is a network architecture that extends MPLS networks to integrate access and aggregation networks into a single MPLS domain, to solve the scaling problems in flat MPLS-based deployments. The Seamless MPLS transport concept described in this chapter partitions the core, aggregation, and access networks into isolated IGP/LDP domains. Seamless MPLS does not define any new protocols or technologies and is based on existing and well-known ones. Seamless MPLS provides end-to-end service-independent transport, separating the service and transport plane. Therefore, it removes the need for service-specific configurations in network transport nodes. Service provisioning is restricted only at the points of the network where it is required.

When BGP is used to distribute a route, it can also distribute an MPLS label that is mapped to that route. The label mapping information is appended to the BGP update message that is used to distribute the route. This is described in RFC 3107, *Carrying Label Information in BGP-4*.

[Figure 275: Seamless MPLS - network topology, control and data plane](#) shows a network with a core area and regional areas. [Figure 275: Seamless MPLS - network topology, control and data plane](#) also shows the control plane used in this Seamless MPLS implementation. For simplification, the control plane is displayed from right to left and the data plane from left to right. In this example, LDP is used as the underlying transport inside each IGP domain. Alternatively, RSVP-TE could be used.

Figure 275: Seamless MPLS - network topology, control and data plane



25635

In typical Seamless MPLS solutions, multiple ABRs are in place that result in some specific BGP configurations to send/receive multiple paths, such as the add-path feature. Because of this, ANs and ABRs have several next hops for the same prefix, allowing the use of redundancy mechanisms such as BGP Prefix Independent Convergence (PIC) edge, also known as BGP Fast ReRoute (FRR). These mechanisms are beyond the scope of this chapter.

AN routers in a regional area learn the reachability of AN routers in other regional areas through BGP labeled routes redistributed by the local ABRs (RFC 3107).

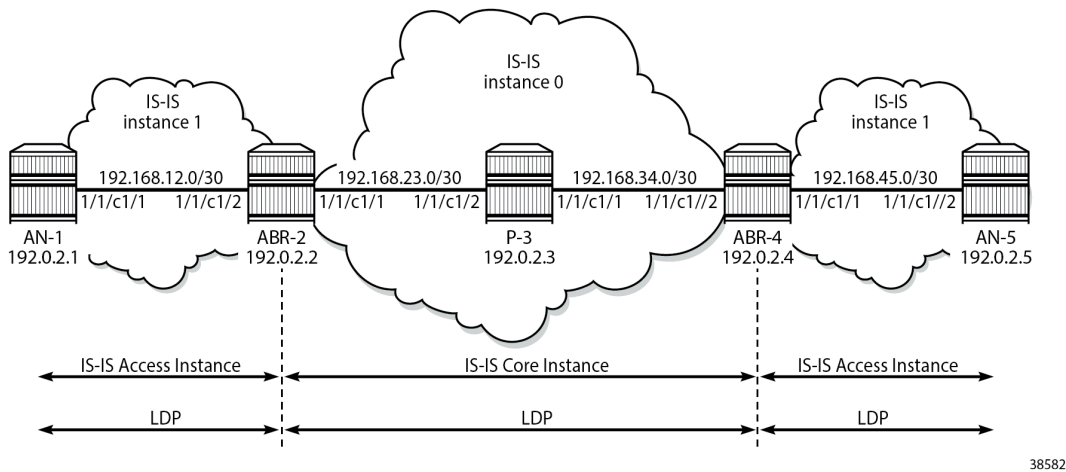
The label stack contains three labels for packets sent in a VPN service between the access nodes:

- The ANs push a service label to the packets sent in the VPN service. The service label remains unchanged end-to-end between ANs. The service label is popped by the remote AN and is the inner label of the label stack.
- The BGP label is the middle label of the label stack and should be regarded as a transport label. The transport label stack is increased to two labels: BGP and LDP transport labels. The BGP label is pushed by the iLER AN and is swapped at the BGP next hop, which can be one of the two local ABRs. Both ABRs are configured with next-hop-self. The BGP label is also swapped by the remote ABR.
- The iLER AN pushes an LDP transport label to the packets sent to the remote AN to reach the BGP next hop. At the local ABR, the LDP transport label is popped and a new LDP transport label is pushed to reach the BGP next hop (remote ABR). The LDP transport label is swapped in every label switching router (LSR) and popped by the ABR nearest to the remote AN. That ABR pops the LDP transport label, swaps the BGP label, and pushes an LDP transport label to reach the remote eLER AN.

Configuration

Figure 276: Seamless MPLS - IGP/LDP domains shows the example topology that is used in this chapter. An Epipe and VPRN are established between the access nodes AN-1 and AN-5. In the regional areas, and in the core area, IS-IS L2 capability is used.

Figure 276: Seamless MPLS - IGP/LDP domains



38582

Initial configuration

All nodes have the following initial configuration:

- Cards, media dependent adapters (MDAs), ports
 - Router interfaces:

```
# on ABR-2:
configure exclusive
router "Base" {
    interface "int-ABR-2-AN-1" {
        port 1/1/c1/2
        ipv4 {
            primary {
                address 192.168.12.2
                prefix-length 30
            }
        }
    }
    interface "int-ABR-2-P-3" {
        port 1/1/c1/1
        ipv4 {
            primary {
                address 192.168.23.1
                prefix-length 30
            }
        }
    }
}
interface "system" {
    ipv4 {
        primary {
```

```

        address 192.0.2.2
        prefix-length 32
    }
}

```

- IS-IS (alternatively, OSPF could be used). Core area and regional areas run an isolated IS-IS instance. ABRs run two IS-IS instances: instance 0 belongs to the core and instance 1 belongs to the access network.

- **Core instance.** All ABRs and Ps have level 2 (L2) capability, as follows:

```

# on ABR-2:
configure exclusive
  router "Base" {
    isis 0 {
      admin-state enable
      level-capability 2
      area-address [49.0001]
      interface "int-ABR-2-P-3" {
        interface-type point-to-point
      }
      interface "system" {
      }
    }
  }

```

- **Access instance.** All ABRs and ANs have L2 capability, as follows:

```

# on ABR-2:
configure exclusive
  router "Base" {
    isis 1 {
      admin-state enable
      level-capability 2
      interface "int-ABR-2-AN-1" {
        interface-type point-to-point
      }
      interface "system" {
      }
    }
  }

```

- LDP

Link LDP is enabled on all router interfaces on all nodes, as follows:

```

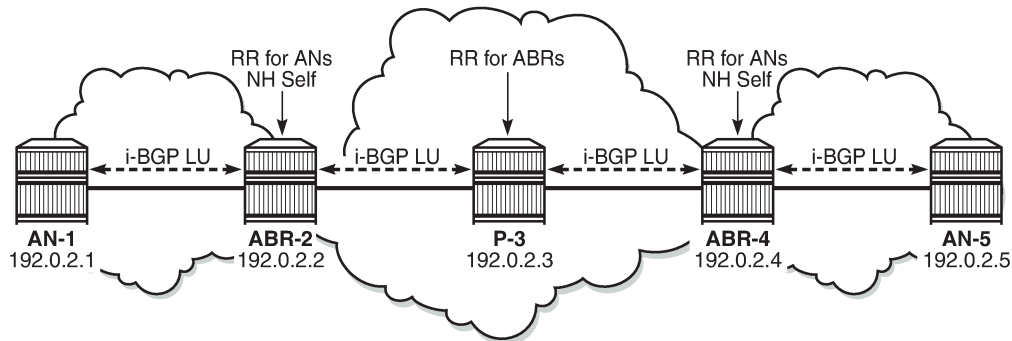
# on ABR-2:
configure exclusive
  router "Base" {
    ldp {
      interface-parameters {
        interface "int-ABR-2-AN-1" {
          ipv4 {
          }
        }
        interface "int-ABR-2-P-3" {
          ipv4 {
          }
        }
      }
    }
  }

```

Configure BGP

BGP is configured on all ABRs and all ANs. P-3 acts as a core Route Reflector (RR). To allow for separation of core/access IGP domains, the ABRs become RRs inline and implement next-hop-self on labeled IPv4 BGP prefixes. [Figure 277: Seamless MPLS - BGP](#) shows the exchange of iBGP Labeled Unicast (LU) routes.

Figure 277: Seamless MPLS - BGP



25637

BGP configuration on ABRs

There are two BGP groups on the ABRs: one group toward the core RR and another group toward the AN, as follows:

```
# on ABR-2:
configure exclusive
router "Base" {
  autonomous-system 64496
  bgp {
    ebgp-default-reject-policy {
      import false
      export false
    }
    group "core" {
      peer-as 64496
      advertise-inactive true
      family {
        vpn-ipv4 true
        label-ipv4 true
      }
    }
  }
  neighbor "192.0.2.3" {
    description "coreRR_P-3"
    group "core"
    next-hop-self true
  }
}
```

Advertise-inactive must be enabled on the BGP group toward the core. The /32 system IP addresses, learned in labeled BGP, are also learned in IS-IS. Because IS-IS has a lower preference compared to iBGP, the IS-IS routes are installed in the routing table. BGP default behavior only advertises those prefixes that were elected by RTM and used. The VPN IPv4 address family is also included, along with labeled

IPv4, to allow setting up L3 VPN services, as shown in next sections. The next-hop attribute of VPN IPv4 prefixes remains unchanged.

```
# on ABR-2:
configure exclusive
router "Base" {
  bgp {
    group "ANs_Label_IPv4+VPN_IPv4" {
      peer-as 64496
      family {
        vpn-ipv4 true
        label-ipv4 true
      }
      cluster {
        cluster-id 2.2.2.2
      }
    }
    neighbor "192.0.2.1" {
      description "AN-1"
      group "ANs_Label_IPv4+VPN_IPv4"
      next-hop-self true
    }
  }
}
```

BGP configuration on the core RR

```
# on P-3:
configure exclusive
router "Base" {
  autonomous-system 64496
  bgp {
    ebgp-default-reject-policy {
      import false
      export false
    }
    group "core" {
      peer-as 64496
      advertise-inactive true
      family {
        vpn-ipv4 true
        label-ipv4 true
      }
      cluster {
        cluster-id 3.3.3.3
      }
    }
    neighbor "192.0.2.2" {
      description "ABR-2"
      group "core"
    }
    neighbor "192.0.2.4" {
      description "ABR-4"
      group "core"
    }
  }
}
```

BGP configuration on ANs toward ABRs

```
# on AN-1:
```



```
configure exclusive
router "Base" {
  autonomous-system 64496
  bgp {
    ebgp-default-reject-policy {
      import false
      export false
    }
    group "ABRs_Label_IPv4+VPN_IPv4" {
      peer-as 64496
      family {
        vpn-ipv4 true
        label-ipv4 true
      }
    }
    neighbor "192.0.2.2" {
      group "ABRs_Label_IPv4+VPN_IPv4"
    }
  }
}
```

Configuring address family **label-ipv4** implies that all advertised IPv4 prefixes are sent to the remote BGP peer as an RFC 3107 formatted label. The **next-hop-self** command only applies to labeled IPv4 prefixes, not to VPN-IPv4.

The BGP sessions can be shown with the following command:

```
[/]
A:admin@P-3# show router bgp summary all

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId          AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
                   PktSent OutQ
-----
192.0.2.2
ABR-2
Def. Inst          64496      6   0 00h00m32s 0/0/0 (VpnIPv4)
                   4   0           0/0/0 (Lbl-IPv4)
192.0.2.4
ABR-4
Def. Inst          64496      3   0 00h00m23s 0/0/0 (VpnIPv4)
                   3   0           0/0/0 (Lbl-IPv4)
-----
```

```
[/]
A:admin@AN-1# show router bgp summary all

=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
ServiceId          AS PktRcvd InQ  Up/Down  State|Rcv/Act/Sent (Addr Family)
                   PktSent OutQ
-----
```

```
192.0.2.2
Def. Inst      64496      6      0 00h00m38s 0/0/0 (VpnIPv4)
                6      0                0/0/0 (Lbl-IPv4)
-----
```

Export policies for BGP

A policy is required on the ANs to advertise the system IP address in labeled BGP toward the ABRs. The same policy is required on the ABRs to advertise their system IP address in labeled BGP toward the core and the AN.

Policy configuration on ANs and ABRs

```
# on AN-1 and ABR-2:
configure exclusive
  policy-options {
    prefix-list "system" {
      prefix 192.0.2.1/32 type exact {
      }
    }
    policy-statement "export-system" {
      entry 10 {
        from {
          prefix-list ["system"]
          protocol {
            name [direct]
          }
        }
        action {
          action-type accept
        }
      }
    }
  }
}
```

This export policy must be applied in the **bgp** context on AN-1: either in the general settings or per **group** or per **neighbor**, as follows:

```
# on AN-1:
configure exclusive
  router "Base" {
    bgp {
      group "ABRs_Label_IPv4+VPN_IPv4" {
        export {
          policy ["export-system"]
        }
      }
    }
  }
```

The same export policy is applied in the group "core" on ABR-2, as follows:

```
# on ABR-2:
configure exclusive
  router "Base" {
    bgp {
      group "core" {
        export {
          policy ["export-system"]
        }
      }
    }
  }
```

```
}
}
```

A similar export policy is defined to export prefix 192.0.2.5 from AN-5 to ABR-4 and from ABR-4 to the RR in the core network, P-3.

The prefix of the remote AN is added to the routing table in AN-1 and services can be configured in the ANs. No service configuration is required in the transit nodes.

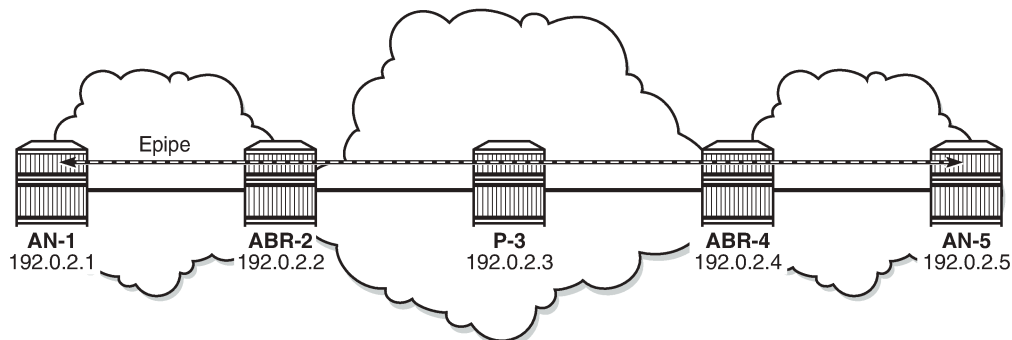
```
[/]
A:admin@AN-1# show router route-table

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]          Type   Proto   Age           Pref
                                   Metric
-----
192.0.2.1/32
  system                          Local  Local   00h05m09s    0
192.0.2.2/32
  192.168.12.2                    Remote ISIS(1) 00h04m57s  18
192.0.2.5/32
  192.0.2.2 (tunneled)            Remote BGP_LABEL 00h00m25s  170
192.168.12.0/30
  int-AN-1-ABR-2                  Local  Local   00h05m09s    0
-----
No. of Routes: 4
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```

Configure SDP and Epipe

An end-to-end Epipe service is established between AN-1 and AN-5, as shown in [Figure 278: End-to-End Epipe service](#).

Figure 278: End-to-End Epipe service



25638

The SDP is configured on AN-1 and AN-5, as follows:

```
# on AN-1:
configure exclusive
service {
  sdp 15 {
    admin-state enable
    delivery-type mpls
    bgp-tunnel true
    far-end {
      ip-address 192.0.2.5
    }
  }
}
```

```
# on AN-5:
configure exclusive
service {
  sdp 51 {
    admin-state enable
    delivery-type mpls
    bgp-tunnel true
    far-end {
      ip-address 192.0.2.1
    }
  }
}
```

Epipe 1 is configured on AN-1 and AN-5, as follows:

```
# on AN-1:
configure exclusive
service {
  epipe "Epipe 1" {
    admin-state enable
    service-id 1
    customer "1"
    spoke-sdp 15:1 {
    }
    sap 1/1/c1/3:1 {
    }
  }
}
```

```
# on AN-5:
configure exclusive
service {
  epipe "Epipe 1" {
    admin-state enable
    service-id 1
    customer "1"
    spoke-sdp 51:1 {
    }
    sap 1/1/c1/3:1 {
    }
  }
}
```

The state of the SDP and of the Epipe service can be verified on AN-1, as follows:

```
[/]
A:admin@AN-1# show service sdp
```

=====

```

Services: Service Destination Points
=====
SdpId  AdmMTU  OprMTU  Far End          Adm  Opr          Del  LSP  Sig
-----
15     0       1552   192.0.2.5       Up  Up           MPLS B    TLDP
-----
Number of SDPs : 1
-----
Legend: R = RSVP, L = LDP, B = BGP, M = MPLS-TP, n/a = Not Applicable
        I = SR-ISIS, 0 = SR-OSPF, T = SR-TE, F = FPE
=====

```

```

[/]
A:admin@AN-1# show service id 1 base
=====
Service Basic Information
=====
Service Id       : 1                Vpn Id          : 0
Service Type     : Epipe
---snip---
Admin State      : Up                Oper State      : Up
---snip---
-----
Service Access & Destination Points
-----
Identifier                Type          AdmMTU  OprMTU  Adm  Opr
-----
sap:1/1/c1/3:1            q-tag        1518   1518   Up  Up
sdp:15:1 S(192.0.2.5)     Spok         0      1552   Up  Up
-----
* indicates that the corresponding row element may have been truncated.

```

The state of the SDP and of the Epipe service can be verified on AN-5, as follows:

```

[/]
A:admin@AN-5# show service sdp
=====
Services: Service Destination Points
=====
SdpId  AdmMTU  OprMTU  Far End          Adm  Opr          Del  LSP  Sig
-----
51     0       1552   192.0.2.1       Up  Up           MPLS B    TLDP
-----
Number of SDPs : 1
-----
Legend: R = RSVP, L = LDP, B = BGP, M = MPLS-TP, n/a = Not Applicable
        I = SR-ISIS, 0 = SR-OSPF, T = SR-TE, F = FPE
=====

```

```

[/]
A:admin@AN-5# show service id 1 base
=====
Service Basic Information
=====
Service Id       : 1                Vpn Id          : 0
Service Type     : Epipe
---snip---
Admin State      : Up                Oper State      : Up
---snip---

```

```

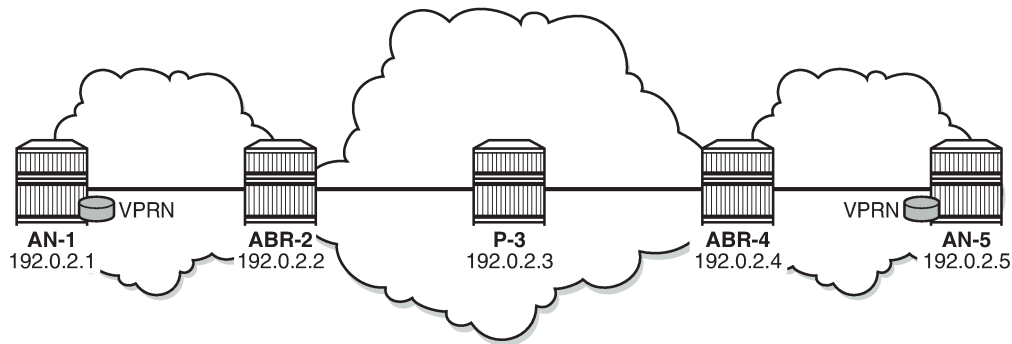
-----
Service Access & Destination Points
-----
Identifier                                     Type      AdmMTU  OprMTU  Adm  Opr
-----
sap:1/1/c1/3:1                               q-tag    1518    1518    Up   Up
sdp:51:1 S(192.0.2.1)                       Spok     0       1552    Up   Up
=====
* indicates that the corresponding row element may have been truncated.

```

Configure VPRN

An L3 VPN service is established on AN-1 and AN-5, as shown in [Figure 279: L3 VPN service](#).

Figure 279: L3 VPN service



25639

The VPRN service is configured on AN-1 and AN-5, as follows. For simplicity, no CEs are attached to the ANs and only one loopback is created for verification.

```

# on AN-1:
configure exclusive
  service {
    vprn "VPRN 2" {
      admin-state enable
      service-id 2
      customer "1"
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "64496:2"
          vrf-target {
            community "target:64496:2"
          }
          auto-bind-tunnel {
            resolution any
          }
        }
      }
    }
  }
  interface "loopback" {
    loopback true
    ipv4 {
      primary {
        address 192.0.1.1
        prefix-length 32
      }
    }
  }

```

```

    }
  }
}

# on AN-5:
configure exclusive
service {
  vprn "VPRN 2" {
    admin-state enable
    service-id 2
    customer "1"
    bgp-ipvpn {
      mpls {
        admin-state enable
        route-distinguisher "64496:2"
        vrf-target {
          community "target:64496:2"
        }
        auto-bind-tunnel {
          resolution any
        }
      }
    }
  }
  interface "loopback" {
    loopback true
    ipv4 {
      primary {
        address 192.0.1.5
        prefix-length 32
      }
    }
  }
}

```

The routing table for VPRN 2 contains the local and the remote loopback addresses. On AN-1, this can be verified as follows:

```

[/]
A:admin@AN-1# show router "2" route-table

=====
Route Table (Service: 2)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]                Type   Proto   Age           Pref
Metric
-----
192.0.1.1/32
loopback                                Local  Local   00h02m26s    0
0
192.0.1.5/32
192.0.2.5 (tunneled:BGP)                 Remote BGP VPN  00h01m08s    170
1000
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

On AN-5, this can be verified as follows:

```

[/]
A:admin@AN-5# show router "2" route-table

```

```

=====
Route Table (Service: 2)
=====
Dest Prefix[Flags]                                Type   Proto   Age           Pref
  Next Hop[Interface Name]                        Metric
-----
192.0.1.1/32                                       Remote BGP VPN 00h01m31s    170
      192.0.2.1 (tunneled:BGP)                    1000
192.0.1.5/32                                       Local  Local   00h02m19s     0
      loopback                                     0
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

Ping messages can be sent from the loopback address in VPRN 2 on AN-1 to the remote loopback address in VPRN 2 on AN-5, as follows:

```

[/]
A:admin@AN-1# ping 192.0.1.5 router-instance "VPRN 2"
PING 192.0.1.5 56 data bytes
64 bytes from 192.0.1.5: icmp_seq=1 ttl=64 time=2.69ms.
64 bytes from 192.0.1.5: icmp_seq=2 ttl=64 time=3.18ms.
. 64 bytes from 192.0.1.5: icmp_seq=3 ttl=64 time=2.87ms.
. 64 bytes from 192.0.1.5: icmp_seq=4 ttl=64 time=2.64ms.
. 64 bytes from 192.0.1.5: icmp_seq=5 ttl=64 time=2.36ms.

---- 192.0.1.5 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 2.36ms, avg = 2.75ms, max = 3.18ms, stddev = 0.271ms

```

In a similar way, ping messages are sent from the loopback address in VPRN 2 on AN-5 to the loopback address in VPRN 2 on AN-1, as follows:

```

[/]
A:admin@AN-5# ping 192.0.1.1 router-instance "VPRN 2"
PING 192.0.1.1 56 data bytes
64 bytes from 192.0.1.1: icmp_seq=1 ttl=64 time=2.05ms.
64 bytes from 192.0.1.1: icmp_seq=2 ttl=64 time=1.82ms.
64 bytes from 192.0.1.1: icmp_seq=3 ttl=64 time=1.92ms.
. 64 bytes from 192.0.1.1: icmp_seq=4 ttl=64 time=1.90ms.
. 64 bytes from 192.0.1.1: icmp_seq=5 ttl=64 time=2.40ms.

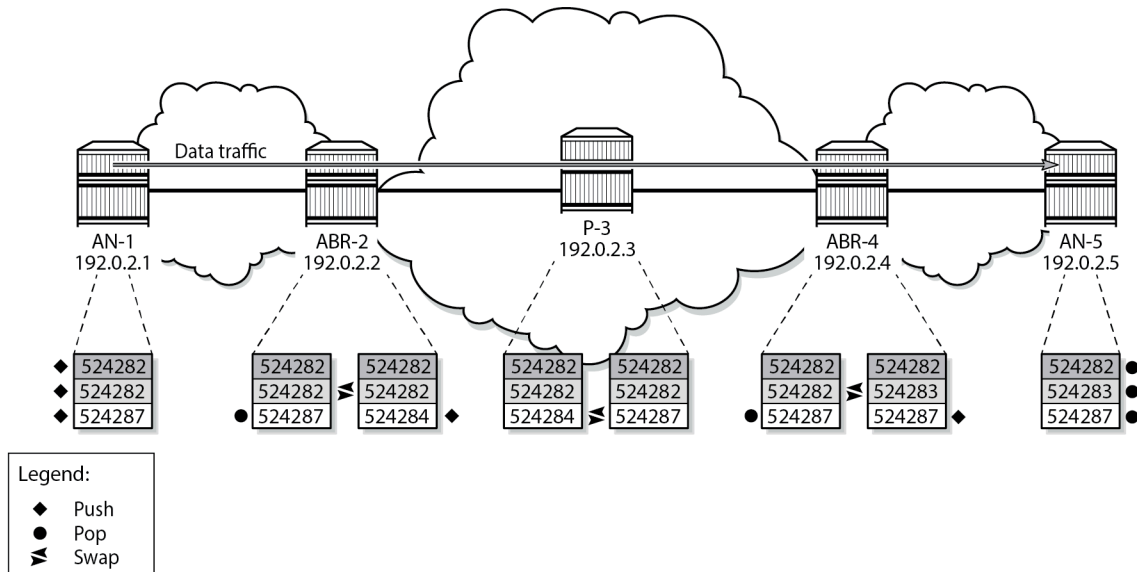
---- 192.0.1.1 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 1.82ms, avg = 2.02ms, max = 2.40ms, stddev = 0.206ms

```

Data plane overview

Figure 280: Label stacks for traffic from AN-1 to AN-5 shows the label stacks used for traffic from AN-1 to AN-5. As an example, an Epipe service is used.

Figure 280: Label stacks for traffic from AN-1 to AN-5



38583

- The service label added for Epipe 1 on AN-1 for egress traffic to AN-5 is 524282. Ingress traffic on AN-1 has service label 524282. This can be shown as follows:

```
[/]
A:admin@AN-1# show service id 1 labels

=====
Martini Service Labels
=====
Svc Id      Sdp Binding      Type  I.Lbl      E.Lbl
-----
1           15:1             Spok  524282     524282
=====
Number of Bound SDPs : 1
=====
```

This service label remains unchanged end-to-end.

On AN-1, the (middle) BGP label for traffic with destination AN-5 is 524282, as follows:

```
[/]
A:admin@AN-1# show router bgp routes 192.0.2.5 label-ipv4

=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP LABEL-IPV4 Routes
=====
Flag Network                                     LocalPref  MED
```

	Nexthop (Router) As-Path	Path-Id	IGP Cost Label
u*>i	192.0.2.5/32	100	None
	192.0.2.2	None	10
	No As-Path		524282

Routes : 1			
=====			

The next hop is ABR-2. AN-1 pushes the LDP label 524287 to reach ABR-2, as follows:

```
[/]
A:admin@AN-1# show router ldp bindings active prefixes prefix 192.0.2.2/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.2/32                         Push
--                                   524287
192.168.12.2                         1/1/c1/1
-----
No. of IPv4 Prefix Active Bindings: 1
=====
```

2. At ABR-2, the service label 524282 remains unchanged. The LDP label 524287 is popped, as follows:

```
[/]
A:admin@ABR-2# show router ldp bindings active prefixes prefix 192.0.2.2/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
(IPv6 LSR ID ::)
=====
---snip---
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.2/32                         Pop
-----
```

```
524287          --
--            --
-----
No. of IPv4 Prefix Active Bindings: 1
=====
```

On ABR-2, the BGP next hop is ABR-4 for prefix 192.0.2.5, as follows:

```
[/]
A:admin@ABR-2# show router bgp routes 192.0.2.5 label-ipv4
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP LABEL-IPV4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  192.0.2.5/32             100        None
      192.0.2.4             None        20
      No As-Path            524282
-----
Routes : 1
=====
```

On ABR-2, the BGP (middle) label 524282 is swapped with (in this case, the same label) 524282 for BGP next hop ABR-4, as follows:

```
[/]
A:admin@ABR-2# show router bgp inter-as-label
=====
BGP Inter-AS labels
Flags: B - entry has backup, P - entry is promoted
=====
NextHop          Received      Advertised    Label
                  Label         Label         Origin
-----
192.0.2.1        524283       524283       Internal
192.0.2.4      524282     524282     Internal
-----
Total Labels allocated:  2
=====
```

ABR-2 pushes a new LDP label (524284) to reach the BGP next hop (ABR-4), as follows:

```
[/]
A:admin@ABR-2# show router ldp bindings active prefixes prefix 192.0.2.4/32
=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
(IPv6 LSR ID ::)
=====
---snip---
```

```

=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.4/32                         Push
--
192.168.23.2                         524284
                                      1/1/c1/1

192.0.2.4/32                         Swap
524284                               524284
192.168.23.2                         1/1/c1/1

-----
No. of IPv4 Prefix Active Bindings: 2
=====

```

3. At LSR P-3, only an LDP label swap occurs. P-3 swaps LDP label 524284 with 524287, as follows:

```

[/]
A:admin@P-3# show router ldp bindings active prefixes prefix 192.0.2.4/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.3)
(IPv6 LSR ID ::)
=====
---snip---
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.4/32                         Push
--
192.168.34.2                         524287
                                      1/1/c1/1

192.0.2.4/32                         Swap
524284                               524287
192.168.34.2                         1/1/c1/1

-----
No. of IPv4 Prefix Active Bindings: 2
=====

```

4. At ABR-4, LDP label 524287 is popped and BGP label 524282 is swapped to label 524283, as follows:

```

[/]
A:admin@ABR-4# show router bgp inter-as-label

=====
BGP Inter-AS labels
Flags: B - entry has backup, P - entry is promoted
=====
NextHop                               Received      Advertised   Label
Label                                 Label         Label        Origin
-----
192.0.2.2                             524283       524283       Internal
192.0.2.5                             524283       524282       Internal
-----

```

```
Total Labels allocated: 2
=====
```

ABR-4 pushes a new LDP label 524287 to reach AN-5, as follows:

```
[/]
A:admin@ABR-4# show router ldp bindings active prefixes prefix 192.0.2.5/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.4)
              (IPv6 LSR ID ::)
=====
---snip---
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                Op
IngLbl                EgrLbl
EgrNextHop            EgrIf/LspId
-----
192.0.2.5/32          Push
--                    524287
192.168.45.2          1/1/c1/1

192.0.2.5/32          Swap
524284                524287
192.168.45.2          1/1/c1/1

-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

5. Finally, at AN-5, all labels in the stack are popped. The LDP label 524287 is popped as follows:

```
[/]
A:admin@AN-5# show router ldp bindings active prefixes prefix 192.0.2.5/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.5)
              (IPv6 LSR ID ::)
=====
---snip---
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                Op
IngLbl                EgrLbl
EgrNextHop            EgrIf/LspId
-----
192.0.2.5/32          Pop
524287                --
--                    --

-----
No. of IPv4 Prefix Active Bindings: 1
=====
```

The BGP (middle) label 524283 is popped.

```
[/]
A:admin@AN-5# show router bgp inter-as-label
```

```
=====
BGP Inter-AS labels
Flags: B - entry has backup, P - entry is promoted
=====
NextHop                Received      Advertised    Label
                        Label         Label         Origin
-----
0.0.0.0                0            524283        Edge
-----
Total Labels allocated: 1
=====
```

The ingress service label 524282 is popped, as follows:

```
[/]
A:admin@AN-5# show service id 1 labels
=====
Martini Service Labels
=====
Svc Id   Sdp Binding      Type  I.Lbl          E.Lbl
-----
1        51:1             Spok  524282         524282
-----
Number of Bound SDPs : 1
=====
```

OAM

The following Operations, Administration, and Maintenance (OAM) commands can be launched to validate reachability between regions using BGP labeled IPv4 routes.

```
[/]
A:admin@AN-1# oam lsp-ping bgp-label prefix 192.0.2.5/32
LSP-PING 192.0.2.5/32: 80 bytes MPLS payload
Seq=1, send from intf int-AN-1-ABR-2, reply from 192.0.2.5
  udp-data-len=32 ttl=255 rtt=2.63ms rc=3 (EgressRtr)

---- LSP 192.0.2.5/32 PING Statistics ----
1 packet sent, 1 packet received, 0.00% packet loss
round-trip min = 2.63ms, avg = 2.63ms, max = 2.63ms, stddev = 0.000ms
```

```
[/]
A:admin@AN-5# oam lsp-ping bgp-label prefix 192.0.2.1/32
LSP-PING 192.0.2.1/32: 80 bytes MPLS payload
Seq=1, send from intf int-AN-5-ABR-4, reply from 192.0.2.1
  udp-data-len=32 ttl=255 rtt=3.84ms rc=3 (EgressRtr)

---- LSP 192.0.2.1/32 PING Statistics ----
1 packet sent, 1 packet received, 0.00% packet loss
round-trip min = 3.84ms, avg = 3.84ms, max = 3.84ms, stddev = 0.000ms
```

In a similar way, LSP trace can validate the reachability of the remote AN, as follows:

```
[/]
A:admin@AN-1# oam lsp-trace bgp-label prefix 192.0.2.5/32 output-format detail
Lsp-trace to 192.0.2.5/32: 1 hops min, 30 hops max, 104 byte packets
```

```
0 192.0.2.1
  DS 1: ipaddr=192.168.12.2 ifaddr=192.168.12.2 iftype=ipv4Numbered MRU=1560
      label[1]=524282 protocol=2(BGP)
1 192.0.2.2 rtt=1.18ms rc=8(DSRtrMatchLabel) rsc=1
  DS 1: ipaddr=192.168.23.2 ifaddr=192.168.23.2 iftype=ipv4Numbered MRU=1560
      label[1]=524282 protocol=2(BGP).
2 192.0.2.4 rtt=2.03ms rc=8(DSRtrMatchLabel).
3 192.0.2.5 rtt=2.26ms rc=3(EgressRtr) rsc=1
```

```
[/]
A:admin@AN-5# oam lsp-trace bgp-label prefix 192.0.2.1/32 output-format detail
lsp-trace to 192.0.2.1/32: 1 hops min, 30 hops max, 104 byte packets
0 192.0.2.5
  DS 1: ipaddr=192.168.45.1 ifaddr=192.168.45.1 iftype=ipv4Numbered MRU=1560
      label[1]=524283 protocol=2(BGP)
1 192.0.2.4 rtt=1.06ms rc=8(DSRtrMatchLabel) rsc=1
  DS 1: ipaddr=192.168.34.1 ifaddr=192.168.34.1 iftype=ipv4Numbered MRU=1560
      label[1]=524283 protocol=2(BGP).
2 192.0.2.2 rtt=1.54ms rc=8(DSRtrMatchLabel).
3 192.0.2.1 rtt=1.88ms rc=3(EgressRtr) rsc=1
```

Conclusion

Seamless MPLS helps to solve the scalability problems of large networks. Seamless MPLS partitions the core, aggregation, and access networks into isolated IGP/LDP domains, which helps to maintain IGP databases small and controlled. Labeled BGP allows the establishment of hierarchical LSPs for end-to-end service set up.

Shared Risk Link Groups for RSVP-Based LSPs

This chapter provides information about Shared Risk Link Groups for RSVP-Based LSPs.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 7.0.R5, but the MD-CLI in the current edition corresponds to SR OS Release 21.2.R1. There are no prerequisites.

Overview

Introduction

Shared Risk Link Group (SRLG) is a feature which allows the user to establish a backup secondary label switched path (LSP) or a fast-reroute (FRR) LSP which is disjoint from the primary LSP. Links which are members of the same SRLG represent resources which share the same risk. For example, fiber links sharing the same conduit or multiple wavelengths sharing the same fiber.

A typical application of the SRLG feature is to provide an automatic placement of secondary backup LSPs or FRR bypass/detour LSPs that minimizes the probability of fate sharing with the primary LSP.

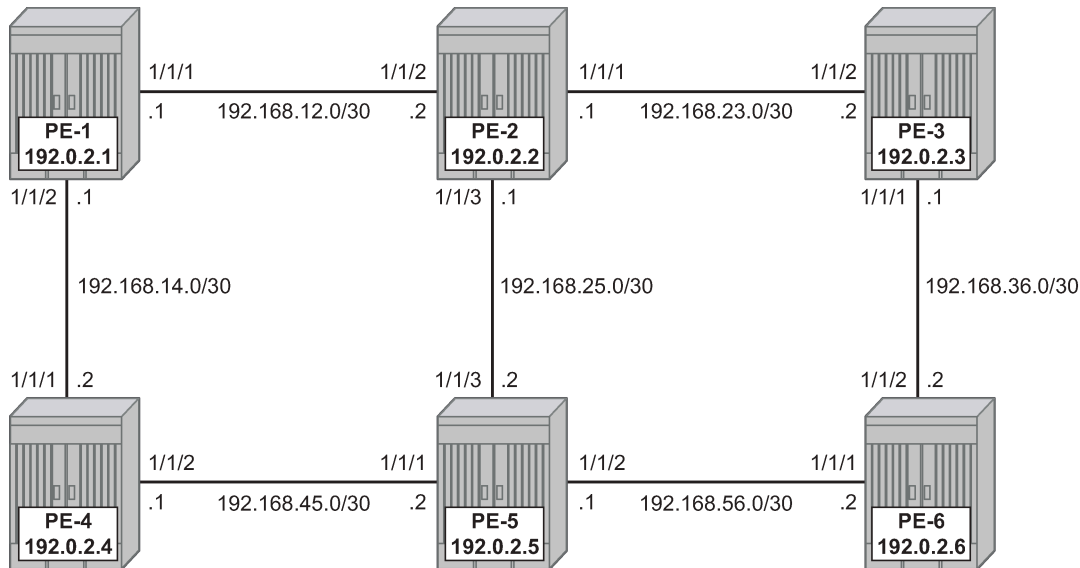
SRLG groups are used to determine which links belong to the same SRLG. The mechanism is similar to Multi-Protocol Label Switching (MPLS) admin groups. To advertise SRLG, the information is part of the IGP TE parameters in an opaque link state advertisement (LSA). In IS-IS (RFC 4205, *Intermediate System to Intermediate System (IS-IS) Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)*), the SRLG is advertised in a new Shared Risk Link Group TLV (type 138). In OSPF (RFC 4203, *OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)*), the SRLG is advertised in a new SRLG sub-TLV (type 16) of the existing Link TLV.

For FRR, a choice can be made on what to do when no FRR tunnel can be found with the SRLG constraints. No FRR tunnel may be signaled or a FRR tunnel might be signaled not taking the SRLG constraints into account.

SRLG

[Figure 281: Example topology](#) shows the example topology for this chapter.

Figure 281: Example topology



OSSG413

A single IGP area (IS-IS in this case) with traffic engineering (TE) enabled is required for the SRLG feature to work properly.

When OSPF is used as the IGP, the functionality is similar.

Configuration

Configuring the IP/MPLS network

IS-IS, MPLS, and RSVP are configured on all interfaces. TE is enabled in IS-IS. Optionally, admin groups "green" and "red" are configured on all nodes. The "green" links are the following: the link between PE-1 and PE-2, the link between PE-2 and PE-3, and the link between PE-3 and PE-6. The "red" links are: the link between PE-1 and PE-4, the link between PE-4 and PE-5, and the link between PE-5 and PE-6. The remaining link is the link between PE-2 and PE-5, which does not belong to an admin group. For more information about admin groups, see chapter [RSVP Point-to-Point LSPs](#).

In addition, ECMP is set to 2, instead of the default value 1 in order to highlight the application of SRLG in the final example: SRLG database.

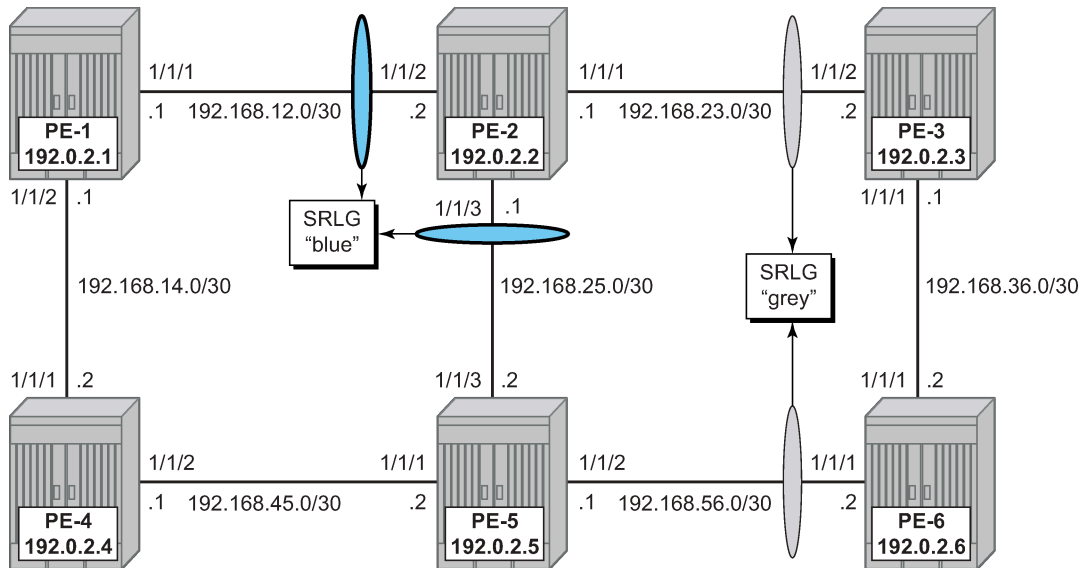
```
# on PE-1:
configure {
  router "Base" {
    ecmp 2
```

Define SRLG groups

Define the SRLG groups, and link them to the related MPLS interfaces.

Two SRLG groups are defined, named blue and gray, as shown in [Figure 282: SRLG topology](#).

Figure 282: SRLG topology



OSSG415

The configuration of the blue SRLG group is only mandatory on PE-1, PE-2, and PE-5, while the gray SRLG group is only mandatory on PE-2, PE-3, PE-5, and PE-6. However, it is good practice to configure both SRLG groups on all nodes, as follows:

```
# on all nodes:
configure {
  routing-options {
    if-attribute {
      srlg-group "blue" {
        value 1
      }
      srlg-group "gray" {
        value 2
      }
    }
  }
}
```

The IP/MPLS interfaces need to be linked to the related SRLG group, which is a unidirectional indicator, applying only to the egress direction; therefore, it needs to be configured on both sides of the IP/MPLS interface. For example on PE-1, the interface to PE-2 is part of srlg-group "blue". An interface can be part of multiple SRLG groups similar to the admin-group functionality.

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      admin-state enable
      interface "int-PE-1-PE-2" {
        admin-group ["green"]
        srlg-group "blue" { }
      }
      interface "int-PE-1-PE-4" {
        admin-group ["red"]
      }
    }
  }
}
```

```
}

```

The same must be done on PE-2, PE-3, PE-5, and PE-6. Afterward, verify the MPLS configuration for example on PE-2, where the SRLG groups are linked to the interfaces. Admin groups are configured in parallel to indicate that both can be configured and will work independently.

```
# on PE-2:
configure {
  router "Base" {
    mpls {
      admin-state enable
      interface "int-PE-2-PE-1" {
        admin-group ["green"]
        srlg-group "blue" { }
      }
      interface "int-PE-2-PE-3" {
        admin-group ["green"]
        srlg-group "gray" { }
      }
      interface "int-PE-2-PE-5" {
        srlg-group "blue" { }
      }
    }
  }
}

```

The SRLG configuration can be verified using the following show commands.

The following shows all SRLG groups on the node:

```
[/]
A:admin@PE-2# show router if-attribute srlg-group
=====
Interface Srlg Groups
=====
Group Name                Group Value    Penalty Weight
-----
blue                       1              0
gray                       2              0
-----
No. of Groups: 2
=====

```

In the following list of MPLS interfaces, admin groups and SRLG groups are indicated.

```
[/]
A:admin@PE-2# show router mpls interface
=====
MPLS Interfaces
=====
Interface                Port-id        Adm  Opr(V4/V6)  TE-
-----                -
system                   system        Up   Up/Down     None
  Admin Groups           None
  SRLG Groups           None
int-PE-2-PE-1           1/1/2        Up   Up/Down     None
  Admin Groups           green
  SRLG Groups           blue
int-PE-2-PE-3           1/1/1        Up   Up/Down     None
  Admin Groups           green
  SRLG Groups           gray
int-PE-2-PE-5           1/1/3        Up   Up/Down     None
  Admin Groups           None

```

```
SRLG Groups                blue
-----
Interfaces : 4
=====
```

To verify the SRLG groups in the IGP TE database, the following command can be used. The output can be extensive, but searching on the SRLG group name will lead to the correct interfaces.

The following output shows the link-state advertisements of PE-2 on PE-1 in this case. The SRLG information is linked to the IP interfaces in a dedicated TE-TLV.

```
[/]
A:admin@PE-2# show router isis database PE-2.00-00 detail

=====
Rtr Base ISIS Instance 0 Database (detail)
=====

Displaying Level 1 database
-----
LSP ID   : PE-2.00-00                Level   : L1
Sequence : 0x34                      Checksum : 0xdfef  Lifetime : 1051
Version  : 1                          Pkt Type : 18     Pkt Ver  : 1
Attributes: L1                        Max Area : 3      Alloc Len : 508
SYS ID   : 1920.0000.2002             SysID Len : 6     Used Len  : 508

TLVs :

---snip---

TE SRLGs :
  SRLGs : PE-1.00
  Lcl Addr : 192.168.12.2
  Rem Addr : 192.168.12.1
  Num SRLGs : 1
          1

---snip---

TE SRLGs :
  SRLGs : PE-3.00
  Lcl Addr : 192.168.23.1
  Rem Addr : 192.168.23.2
  Num SRLGs : 1
          2

---snip---

TE SRLGs :
  SRLGs : PE-5.00
  Lcl Addr : 192.168.25.1
  Rem Addr : 192.168.25.2
  Num SRLGs : 1
          1

---snip---
```

On-line verification

An on-line verification can be done by a **tools perform** command. This will trigger a Constrained Shortest Path First (CSPF) call to the Interior Gateway Protocol (IGP) TE database, and the result will be an Explicit Route Object (ERO) object which can potentially be used to set up a CSPF-based LSP.

The following shows the command syntax.

```
[/]
A:admin@PE-1# tools perform router mpls cspf to 192.0.2.3 ?

cspf to <IP address> [from <IP address>]
                    [bandwidth <number>]
                    [include-bitmap <string>]
                    [exclude-bitmap <string>]
                    [hop-limit <number>]
                    [exclude-address <ipv4 address>]
                    [metric-type-te]
                    [strict-srlg]
                    [srlg-group <number>]
                    [exclude-node <ipv4 address>]
                    [skip-interface <string>]
                    [ds-class-type <number>]
                    [cspf-reqtype <keyword>]
                    [least-fill-min-thd <number>]
                    [setup-priority <number>]
                    [hold-priority <number>]

bandwidth           - rate-in-mbps
cspf-reqtype        - all|random|least-fill : keywords
ds-class-type       - <number> - <0..7>
exclude-address     - a.b.c.d (outbound interface)
exclude-bitmap      - string '<1..40 characters>'
exclude-node        - [a.b.c.d] (outbound interface)
from                - ipv4 address '<d.d.d.d>' or ipv6 address
                    '(<x:x:x:x:x:x:x>|<x:x:x:x:x:d.d.d.d>)'
hold-priority       - <number> - <0..7>
hop-limit           - <number> - <2..255>
include-bitmap      - [0..4294967295] - accepted in decimal, hex(0x) or binary(0b)
least-fill-min-thd  - <number> - <1..100>
metric-type-te      - <keyword>
setup-priority      - <number> - <0..7>
skip-interface      - [max 32 chars]
srlg-group          - <number> - <0..4294967295>
strict-srlg         - <keyword>
```

Where the relevant parameters are:

- **to** — Defines the far-end address of the LSP. This is the system-address of the destination LER
- **srlg-group** — Specifies which SRLG groups should be avoided while building the path to the destination (ERO object)
- **strict-srlg** — Indicates whether the SRLG group is a strict requirement or not. When this parameter is given, only paths without traversing the SRLG will be displayed.

Example:

On PE-1, a CSPF calculation is made with PE-3 as destination, without any SRLG restrictions, as follows:

```
[/]
A:admin@PE-1# tools perform router mpls cspf to 192.0.2.3
Req CSPF for all ECMP paths
```

```

from: this node to: 192.0.2.3 w/(no Diffserv) class: 0 , setup Priority 7,
                               Hold Priority 0 TE Class: 7

CSPF Path
To      : 192.0.2.3
Path 1  : (cost 20)
  Src:   192.0.2.1  (= Rtr)
  Egr:   192.168.12.1  -> Ingr: 192.168.12.2      Rtr: 192.0.2.2      (met 10)
  Egr:   192.168.23.1  -> Ingr: 192.168.23.2      Rtr: 192.0.2.3      (met 10)
  Dst:   192.0.2.3  (= Rtr)

```

With a restriction on **srlg-group "blue"** (grp-id =1), the CSPF calculation is as follows:

```

[/]
A:admin@PE-1# tools perform router mpls cspf to 192.0.2.3 srlg-group 1
Req CSPF for all ECMP paths
  from: this node to: 192.0.2.3 w/(no Diffserv) class: 0 , setup Priority 7,
                               Hold Priority 0 TE Class: 7

CSPF Path
To      : 192.0.2.3
Path 1  : (cost 40)
  Src:   192.0.2.1  (= Rtr)
  Egr:   192.168.14.1  -> Ingr: 192.168.14.2      Rtr: 192.0.2.4      (met 10)
  Egr:   192.168.45.1  -> Ingr: 192.168.45.2      Rtr: 192.0.2.5      (met 10)
  Egr:   192.168.56.1  -> Ingr: 192.168.56.2      Rtr: 192.0.2.6      (met 10)
  1 SRLGs: 2
  Egr:   192.168.36.2  -> Ingr: 192.168.36.1      Rtr: 192.0.2.3      (met 10)
  Dst:   192.0.2.3  (= Rtr)

```

The path will be through PE-4, PE-5, and PE-6.

When a strict restriction is requested on **srlg-group "gray"**, no valid CSPF path toward the destination can be found.

```

[/]
A:admin@PE-1# tools perform router mpls cspf to 192.0.2.3 srlg-group 2 strict-srlg
Req CSPF for all ECMP paths
  from: this node to: 192.0.2.3 w/(no Diffserv) class: 0 , setup Priority 7,
                               Hold Priority 0 TE Class: 7

MINOR: CLI #2005: Error while processing command - No CSPF path to "192.0.2.3" with specified
constraints.

```

Removing the **strict** restriction results in a successful return of CSPF, indicating that the CSPF path is not SRLG disjoint.

```

*A:PE-1# tools perform router mpls cspf to 192.0.2.3 srlg-group 2
Req CSPF for all ECMP paths
  from: this node to: 192.0.2.3 w/(no Diffserv) class: 0 , setup Priority 7,
                               Hold Priority 0 TE Class: 7

CSPF Path
To      : 192.0.2.3 (NOT SRLG DISJOINT)
Path 1  : (cost 20)
  Src:   192.0.2.1  (= Rtr)
  Egr:   192.168.12.1  -> Ingr: 192.168.12.2      Rtr: 192.0.2.2      (met 10)
  1 SRLGs: 1
  Egr:   192.168.23.1  -> Ingr: 192.168.23.2      Rtr: 192.0.2.3      (met 10)
  1 SRLGs: 2
  Dst:   192.0.2.3  (= Rtr)

```

The best practice for debugging is to enable debug-tracing on the CSPF process, with following command:

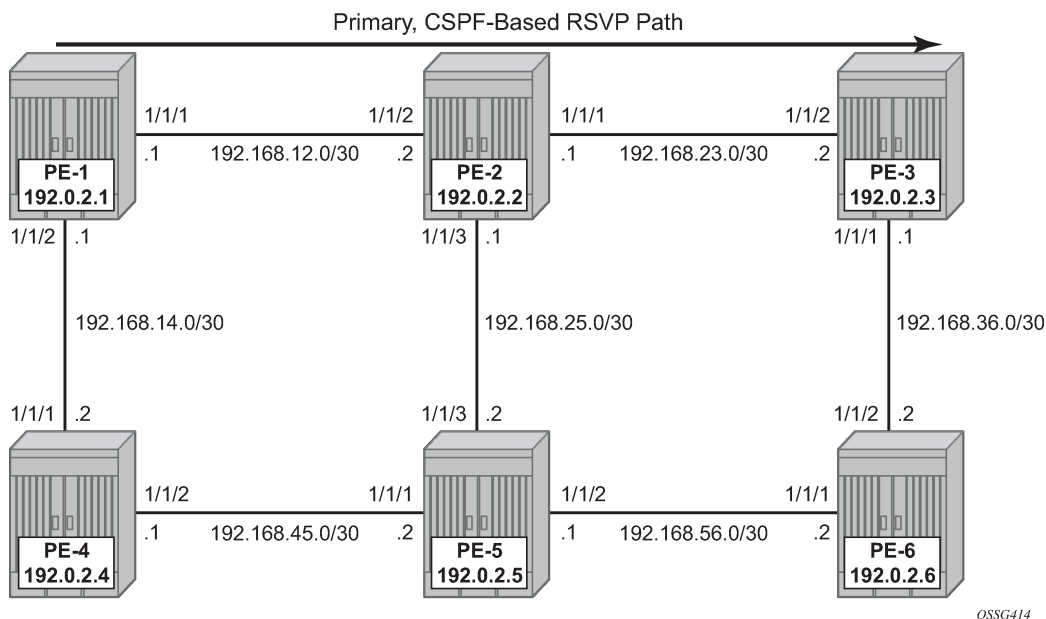
```
# on PE-1 (in classic CLI):
debug
router Base
  isis 0
  cspf
```

SRLG for FRR

The fast-reroute mechanism used here is facility link protection (**frr-method facility node-protect false**). The SRLG feature is independent of the FRR type and works for all combinations (facility versus one-to-one, link versus node protection).

Configure an LSP from PE-1 to PE-3, and enable CSPF.

Figure 283: Path primary RSVP-TE LSP



OSSG414

The configuration of the LSP "LSP-PE-1-PE-3_FRR_facility-link" is based on an empty path, with FRR facility link protection enabled.

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      path "dyn" {
        admin-state enable
      }
      lsp "LSP-PE-1-PE-3_FRR_facility-link" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.3
        path-computation-method local-cspf
        fast-reroute {
          frr-method facility
        }
      }
    }
  }
}
```

```

node-protect false
}
primary "dyn" {
}
}

```

To verify the primary path, the classic-CLI command **oam lsp-trace** can be used, checking the intermediate nodes.

```

[/]
A:admin@PE-1# //
INFO: CLI #2051: Switching to the classic CLI engine
A:PE-1# oam lsp-trace "LSP-PE-1-PE-3_FRR_facility-link" detail
lsp-trace to LSP-PE-1-PE-3_FRR_facility-link: 0 hops min, 0 hops max, 116 byte packets
1 192.0.2.2 rtt=3.00ms rc=8(DSRtrMatchLabel) rsc=1
   DS 1: ipaddr=192.168.23.2 ifaddr=192.168.23.2 iftype=ipv4Numbered MRU=1564
       label[1]=524287 protocol=4(RSVP-TE)
2 192.0.2.3 rtt=3.47ms rc=3(EgressRtr) rsc=1

```

To verify if the bypass tunnels are up and running, an indication (@) can be found in the detail output of **show router mpls ls <x> path detail** as seen in the following output.

```

*A:PE-1# show router mpls lsp "LSP-PE-1-PE-3_FRR_facility-link" path detail
=====
MPLS LSP LSP-PE-1-PE-3_FRR_facility-link Path (Detail)
=====
Legend :
@ - Detour Available          # - Detour In Use
b - Bandwidth Protected      n - Node Protected
s - Soft Preemption
S - Strict                    L - Loose
A - ABR                       + - Inherited
=====
-----
LSP LSP-PE-1-PE-3_FRR_facility-link
Path dyn
-----
LSP Name      : LSP-PE-1-PE-3_FRR_facility-link
From          : 192.0.2.1
To            : 192.0.2.3
Admin State   : Up              Oper State    : Up
Path Name     : dyn
Path LSP ID   : 25600           Path Type     : Primary
Path Admin    : Up              Path Oper     : Up
Out Interface : 1/1/1           Out Label     : 524287

---snip---

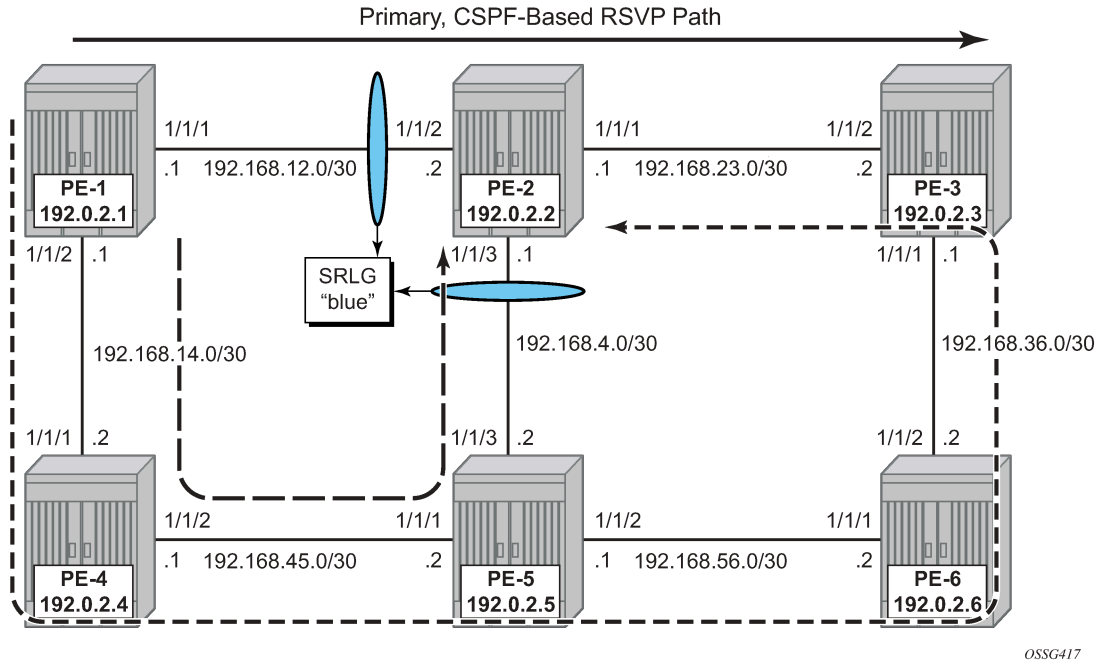
Explicit Hops :
  No Hops Specified
Actual Hops   :
  192.168.12.1(192.0.2.1) @      Record Label  : N/A
-> 192.168.12.2(192.0.2.2) @      Record Label  : 524287
-> 192.168.23.2(192.0.2.3) @      Record Label  : 524287
Computed Hops :
  192.168.12.1(S)
-> 192.168.12.2(S)
-> 192.168.23.2(S)
Resignal Eligible: False
Last Resignal   : n/a          CSPF Metric   : 20
=====

```


* indicates that the corresponding row element may have been truncated.

Two links are protected: one bypass tunnel originates in PE-1 protecting the link between PE-1 and PE-2. Another bypass tunnel originates in PE-2 protecting the link between PE-2 and PE-3. The focus is on the bypass tunnel originating in PE-1. When SRLG is enabled, the bypass tunnel originating in PE-1 will have different hops. The expected paths followed by the bypass tunnels originating in PE-1 with and without SRLG are shown in [Figure 284: FRR bypass tunnels originating in PE-1 with and without SRLG](#).

Figure 284: FRR bypass tunnels originating in PE-1 with and without SRLG



OSSG417

To verify the bypass data path on the point of local repair (PLR) PE-1, the following command can be used.

```
[/]
A:admin@PE-1# show router mpls bypass-tunnel detail

=====
MPLS Bypass Tunnels (Detail)
=====
-----
bypass-link192.168.12.2-61441
-----
To          : 192.168.25.1      State          : Up
Out I/F     : 1/1/2            Out Label     : 524287
Up Time    : 0d 00:05:20      Active Time   : n/a
Reserved BW : 0 Kbps          Protected LSP Count : 1
Type       : Dynamic          Bypass Path Cost : 30
Setup Priority : 7             Hold Priority   : 0
Class Type  : 0
Exclude Node : None           Inter-Area     : False
Computed Hops :
  192.168.14.1(S)            Egress Admin Groups :
  -> 192.168.14.2(S)         red
  -> 192.168.45.2(S)         red
                               Egress Admin Groups : None
```

```

-> 192.168.25.1(S)          Egress Admin Groups : None
Actual Hops      :
  192.168.14.1(192.0.2.1)  Record Label       : N/A
-> 192.168.14.2(192.0.2.4)  Record Label       : 524287
-> 192.168.45.2(192.0.2.5)  Record Label       : 524286
-> 192.168.25.1(192.0.2.2)  Record Label       : 524286
Last Resignal   :
Attempted At    : n/a      Resignal Reason    : n/a
Resignal Status: n/a      Reason             : n/a
=====

```

The SRLG restriction is not taken into account at this moment at PLR PE-1. The actual hops are PE-4, PE-5, and PE-2 visualized by the path with the long dashes in [Figure 284: FRR bypass tunnels originating in PE-1 with and without SRLG](#).

To take the SRLG restrictions into account, the following additional configuration is needed for MPLS on PE-1.

```

[ex:configure router "Base" mpls]
A:admin@PE-1# srlg-frr ?

srlg-frr <keyword>
<keyword> - (loose|strict)

Use of Shared Risk Link Group (SRLG) constraint for FRR path computation

```

```

# on PE-1:
configure {
  router "Base" {
    mpls {
      srlg-frr strict
    }
  }
}

```

The option strict should only be used if the logical topology allows this. In other words, one must be sure that an alternative path is possible which avoids SRLG-groups.



Note:

Enabling or disabling SRLG for FRR is a system-wide configuration and requires the MPLS routing instance to be manually disabled and then re-enabled to activate the change. This may cause service outage. Nokia recommends that the operator incorporates the SRLG into the initial network design and implementation to minimize the traffic loss.

```

# on all nodes:
configure exclusive
  router "Base" {
    rsvp {
      admin-state disable
      commit
      admin-state enable
      commit
    }
  }
}

```

The bypass tunnel originating in PLR PE-1 can be verified with a previously used command.

```

[/]
A:admin@PE-1# show router mpls bypass-tunnel detail

```

```

=====
MPLS Bypass Tunnels (Detail)

```

```

=====
-----
bypass-link192.168.12.2-61442
-----
-----
To          : 192.168.23.1      State          : Up
Out I/F     : 1/1/2           Out Label     : 524287
Up Time    : 0d 00:00:06     Active Time   : n/a
Reserved BW : 0 Kbps         Protected LSP Count : 1
Type       : Dynamic         Bypass Path Cost : 50
Setup Priority : 7           Hold Priority  : 0
Class Type  : 0
Exclude Node : None         Inter-Area    : False
Computed Hops :
    192.168.14.1(S)         Egress Admin Groups :
    -> 192.168.14.2(S)     red
    -> 192.168.45.2(S)    Egress Admin Groups :
    -> 192.168.56.2(S)    red
    -> 192.168.36.1(S)    Egress Admin Groups :
    -> 192.168.23.1(S)    green
    Actual Hops           Egress Admin Groups :
    192.168.14.1(192.0.2.1) Record Label       : N/A
    -> 192.168.14.2(192.0.2.4) Record Label       : 524287
    -> 192.168.45.2(192.0.2.5) Record Label       : 524286
    -> 192.168.56.2(192.0.2.6) Record Label       : 524286
    -> 192.168.36.1(192.0.2.3) Record Label       : 524284
    -> 192.168.23.1(192.0.2.2) Record Label       : 524285
Last Resignal :
Attempted At  : n/a         Resignal Reason  : n/a
Resignal Status: n/a      Reason           : n/a
=====
-----

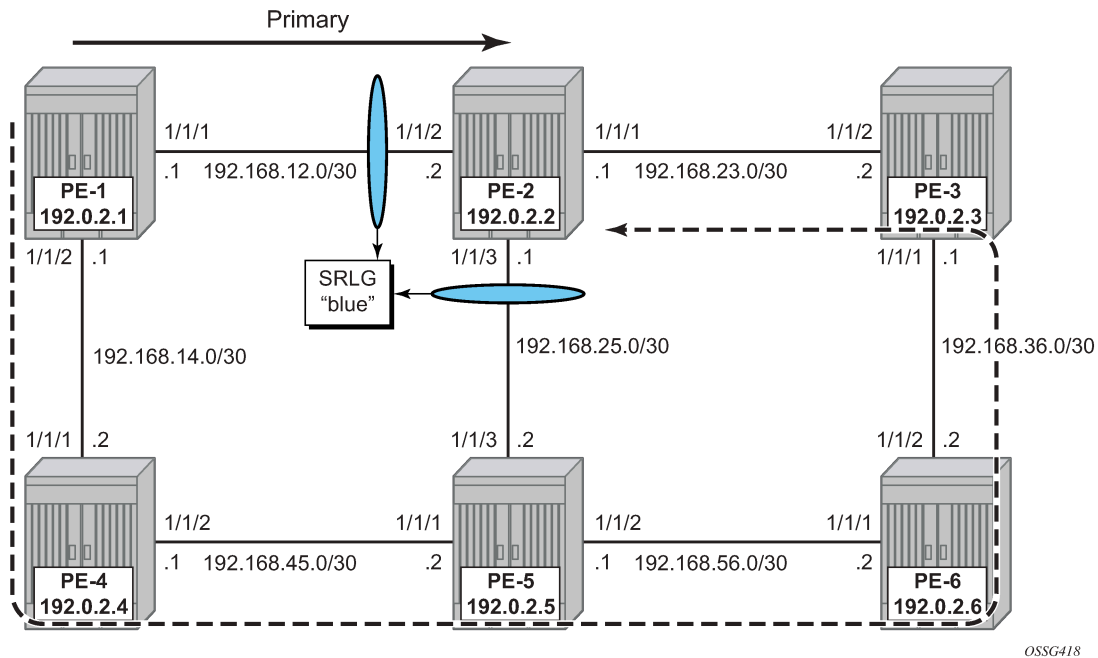
```

This path taking the SRLG constraints into account is represented by the line with the short dashes in [Figure 284: FRR bypass tunnels originating in PE-1 with and without SRLG](#).

SRLG for standby path

Where SRLG groups can be constraints for bypass tunnels, they can also be a constraint to set up a secondary path. [Figure 285: SRLG for secondary path](#) shows that the secondary path is expected to follow the dashed line instead of passing over the direct link between PE-5 and PE-2.

Figure 285: SRLG for secondary path



An LSP is configured with a primary and a secondary path, which have no hops defined. The configuration of the LSP will need a specific indication at the level of the secondary path to enable the restriction on the srlg-groups.

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      path "prim" {
        admin-state enable
      }
      path "secon" {
        admin-state enable
      }
    }
    lsp "LSP-PE-1-PE-2-srlg" {
      admin-state enable
      type p2p-rsvp
      to 192.0.2.2
      path-computation-method local-cspf
      primary "prim" {
      }
      secondary "secon" {
        srlg true
        standby true
      }
    }
  }
}
```

Where both paths are empty paths, the ERO object creation solely relies on CPSF without any specific hop.

To verify the data path, the detailed output of the **show router mpls lsp <.> path** command can be used, as well as the **lsp-trace** OAM command. This output shows both ERO objects of the primary and secondary path.

```
*A:PE-1# show router mpls lsp "LSP-PE-1-PE-2-srlg" path detail
=====
MPLS LSP LSP-PE-1-PE-2-srlg Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use
  b - Bandwidth Protected      n - Node Protected
  s - Soft Preemption
  S - Strict                    L - Loose
  A - ABR                      + - Inherited
=====
-----
LSP LSP-PE-1-PE-2-srlg
Path prim
-----
---snip---

Explicit Hops      :
  No Hops Specified
Actual Hops        :
  192.168.12.1(192.0.2.1)      Record Label      : N/A
-> 192.168.12.2(192.0.2.2)      Record Label      : 524287
Computed Hops      :
  192.168.12.1(S)
-> 192.168.12.2(S)
Resignal Eligible: False
Last Resignal     : n/a          CSPF Metric       : 10
-----
LSP LSP-PE-1-PE-2-srlg
Path secon
-----
---snip---

Explicit Hops      :
  No Hops Specified
Actual Hops        :
  192.168.14.1(192.0.2.1)      Record Label      : N/A
-> 192.168.14.2(192.0.2.4)      Record Label      : 524286
-> 192.168.45.2(192.0.2.5)      Record Label      : 524285
-> 192.168.56.2(192.0.2.6)      Record Label      : 524285
-> 192.168.36.1(192.0.2.3)      Record Label      : 524287
-> 192.168.23.1(192.0.2.2)      Record Label      : 524284
Computed Hops      :
  192.168.14.1(S)
-> 192.168.14.2(S)
-> 192.168.45.2(S)
-> 192.168.56.2(S)
-> 192.168.36.1(S)
-> 192.168.23.1(S)
Srlg               : Enabled          Srlg Disjoint     : True
Resignal Eligible: False
Last Resignal     : n/a          CSPF Metric       : 50
=====
```

The **lsp-trace** command can be used for secondary path as well. The intermediate LSRs and the MPLS labels used can be clearly seen.

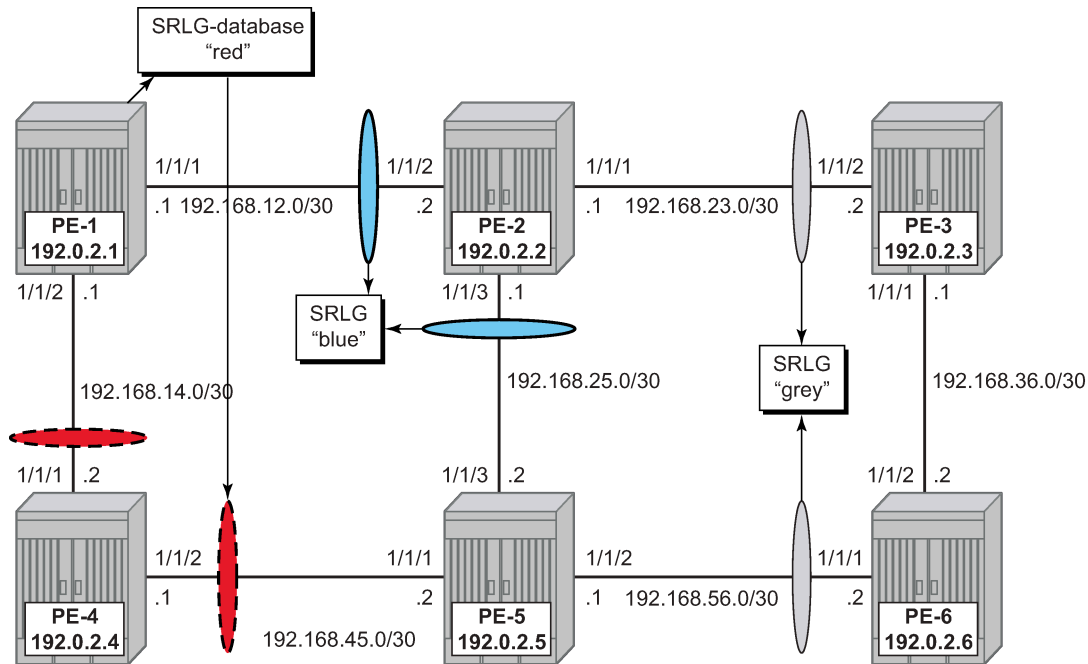
```
[/]  
A:admin@PE-1# //  
INFO: CLI #2051: Switching to the classic CLI engine  
A:PE-1# oam lsp-trace LSP-PE-1-PE-2-srlg"LSP-PE-1-PE-2-srlg" path "secon" detail  
lsp-trace to LSP-PE-1-PE-2-srlg: 0 hops min, 0 hops max, 116 byte packets  
1 192.0.2.4 rtt=2.61ms rc=8(DSRtrMatchLabel) rsc=1  
   DS 1: ipaddr=192.168.45.2 ifaddr=192.168.45.2 iftype=ipv4Numbered MRU=1564  
        label[1]=524285 protocol=4(RSVP-TE)  
2 192.0.2.5 rtt=3.97ms rc=8(DSRtrMatchLabel) rsc=1  
   DS 1: ipaddr=192.168.56.2 ifaddr=192.168.56.2 iftype=ipv4Numbered MRU=1564  
        label[1]=524285 protocol=4(RSVP-TE)  
3 192.0.2.6 rtt=4.78ms rc=8(DSRtrMatchLabel) rsc=1  
   DS 1: ipaddr=192.168.36.1 ifaddr=192.168.36.1 iftype=ipv4Numbered MRU=1564  
        label[1]=524287 protocol=4(RSVP-TE)  
4 192.0.2.3 rtt=4.43ms rc=8(DSRtrMatchLabel) rsc=1  
   DS 1: ipaddr=192.168.23.1 ifaddr=192.168.23.1 iftype=ipv4Numbered MRU=1564  
        label[1]=524284 protocol=4(RSVP-TE)  
5 192.0.2.2 rtt=4.38ms rc=3(EgressRtr) rsc=1  
A:PE-1# //  
INFO: CLI #2052: Switching to the MD-CLI engine
```

SRLG database

In case not all IP/MPLS routers in the area support SRLG, a static SRLG database can be created on the systems which will be used as an additional constraint when performing the CSPF calculation to define the path.

[Figure 286: SRLG database example](#) shows an example where an additional SRLG group "red" is defined on PE-1, with information related to the interface between PE-4 and PE-5.

Figure 286: SRLG database example



OSSG419

```
# on PE-1:
configure {
  routing-options {
    if-attribute {
      srlg-group "red" {
        value 3
      }
    }
  }
}
```

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      admin-state enable
      srlg-database {
        router-id 192.0.2.4 {
          admin-state enable
          interface 192.168.45.1 srlg-group "red" { }
        }
        router-id 192.0.2.5 {
          admin-state enable
          interface 192.168.45.2 srlg-group "red" { }
        }
      }
    }
  }
  interface "int-PE-1-PE-4" {
    srlg-group "red" { }
  }
}
```

This information is local to PE-1 and will only have effect on CSPF calculations on PE-1, not on the other nodes.

When a CSPF calculation is done for a path from PE-1 to PE-5, the result will be two equal-cost paths, because ECMP equals 2. When adding the **srlg-group "red"** as a restriction, only a single path will be found, passing PE-2.

Conclusion

Interpreting the SRLG information into the TE database makes it possible to protect an LSP even when multiple IP/MPLS interfaces fail as a result of an underlying transmission failure. Transmission failures can occur quite often because not all transmission links are one to one protected.

SRLG groups in MPLS provide a very dynamic and simple way to assure LSP FRR path protection on every PLR throughout the followed LSP. The SRLG groups are also taken into account when defining the ERO for secondary paths, at least if the configured secondary path is empty.

For interoperability reasons, the SRLG-database is available, because systems can link interfaces to an SRLG with interconnecting systems that do not support the SRLG feature; so they cannot advertise the SRLG information through the IGP.

The creation and maintenance of an SRLG database requires operational effort and systems that do not support SRLG will never take any SRLG information into account during CSPF calculation for the creation of FRR bypass or detour tunnels.

Static Point-to-Point LSPs

This chapter provides information about static point-to-point label switched paths (LSPs).

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter is applicable to SR OS and was originally written for SR OS Release 7.0.R5. The MD-CLI in the current edition corresponds to SR OS Release 21.2.R1. There are no prerequisites or conditions on the hardware for this configuration.

Overview

Due to the connectionless nature of the network layer protocol IP, packets travel through the network on a hop-by-hop basis with routing decisions made at each node. As a result, hyperaggregation of data on certain links may occur and it may impact the provider's ability to provide guaranteed service levels across the network end-to-end. To address these shortcomings, multiprotocol label switching (MPLS) was developed. MPLS provides the capability to establish connection-oriented paths, called label switched paths (LSPs), over a connectionless (IP) network.

The LSP offers a mechanism to engineer network traffic independently from the underlying network routing protocol (mostly IP) to improve the network resiliency and recovery options and to permit delivery of new services that are not readily supported by conventional IP routing techniques, such as Layer 2 IP Virtual Private Networks (VPNs). These benefits are essential for today's communication network explaining the wide deployment base of the MPLS technology.

RFC 3031, *Multiprotocol Label Switching Architecture*, specifies the MPLS architecture while this document describes the configuration and troubleshooting of static point-to-point LSPs on SR OS. Point-to-point LSPs can also be dynamically established using a label signaling protocol, such as label distribution protocol (LDP)—as described in chapter [LDP Point-to-Point LSPs](#)—or resource reservation protocol (RSVP)—as described in chapter [RSVP Point-to-Point LSPs](#).

Packet forwarding

As a packet of a connectionless network layer protocol travels from one router to the next, each router in the network makes an independent forwarding decision by performing the following basic tasks: first analyzing the packet header, then referencing the local routing table to find the longest match based on the destination address in the IP header, and finally sending out the packet on the selected interface.

In other terms, the first function partitions the entire set of possible packets into a set of forwarding equivalence classes (FECs). All packets associated with a particular FEC will be forwarded along the same

logical path to the same destination. The second function maps each FEC to a next hop destination router. Each router along the path performs these actions.

In MPLS, the assignment of a particular packet to a particular FEC is done just once, as the packet enters the network. In turn, the FEC is mapped to an LSP, which is established prior to any data flowing.

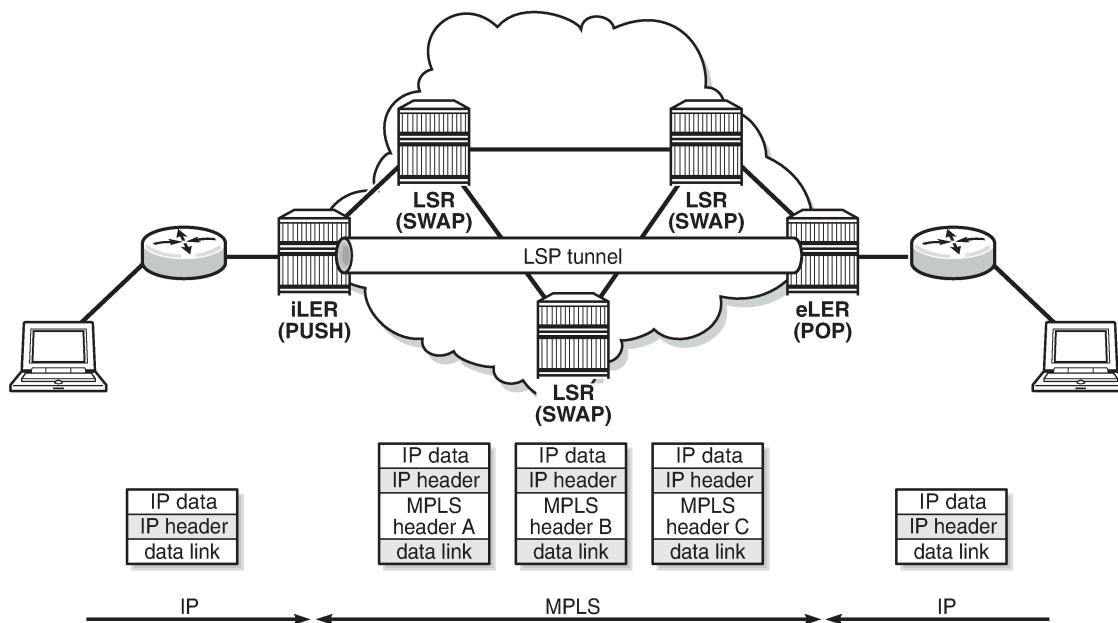
An MPLS label, representing the FEC to which the packet is assigned, is attached to the packet (push operation) and once labeled, the packet is forwarded to the next hop router along that LSP path.

At subsequent hops, there is no further analysis of the network layer header of the packet. Instead, the label is used as an index into a table which specifies the next hop and a new label. The old label is replaced with the new label (swap operation), and the packet is forwarded to its next hop.

At the MPLS network egress, the label is removed from the packet (pop operation). If this router is the final destination (based on the remaining packet), the packet is handed to the receiving application, such as a virtual private LAN service (VPLS). If this router is not the final destination of the packet, the packet will be sent into a new MPLS tunnel or forwarded by conventional IP forwarding toward the layer 3 destination.

Terminology

Figure 287: Generic MPLS network, MPLS label operations



25762

Figure 287: Generic MPLS network, MPLS label operations shows a general network topology clarifying the MPLS-related terms. A Label Edge Router (LER) is a device at the edge of an MPLS network, with at least one interface outside the MPLS domain. A router is usually defined as an LER based on its position relative to a particular LSP. The MPLS router at the head-end of an LSP is called the ingress label edge router (iLER). The MPLS router at the tail-end of an LSP is called the egress label edge router (eLER).

The iLER receives unlabeled packets from outside the MPLS domain, then applies MPLS labels to the packets, and forwards the labeled packets into the MPLS domain. The eLER receives labeled packets from the MPLS domain, then removes the labels, and forwards unlabeled packets outside the MPLS domain.

The last LSR before the eLER can be configured with an implicit-null label (numeric value 3). This LSR will pop the outer label and send MPLS packets without an outer label to the eLER. This is known as Penultimate Hop Popping (PHP). A Label Switching Router (LSR) is a device internal to an MPLS network, with all interfaces inside the MPLS domain. These devices switch labeled packets inside the MPLS domain. In the core of the network, LSRs ignore the network layer (IP) header of the packet and simply forward the packet using the MPLS label swapping mechanism.

A single LSP is unidirectional. In common practice, because the bidirectional nature of most traffic flows is implied, the term LSP often is used to define the pair of LSPs that enable the bidirectional flow. For ease of terminology and discussion however, the LSP in this chapter is referred to as a single entity.

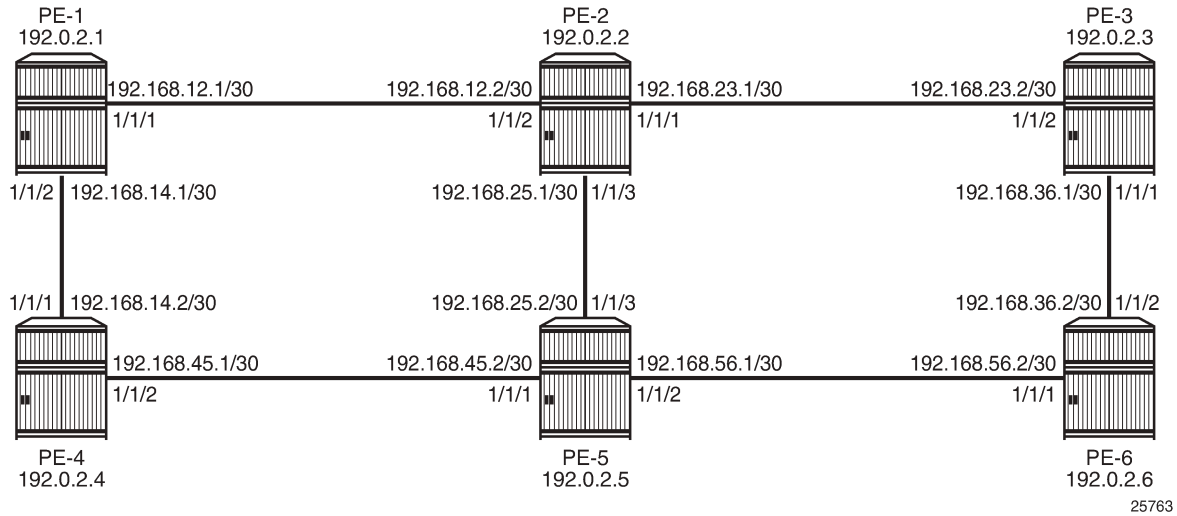
LSP establishment

Prior to packet forwarding, the LSP must be established. In order to do so, labels need to be distributed for the path. For static LSPs, the label distribution is done manually by the network administrator. Although a high control level of the labels in use is achieved, the LSP cannot enjoy the resilience and recovery functionality the dynamic label signaling protocols can offer.

Example topology

Figure 288: MPLS example topology shows the example topology consisting of six SR OS nodes located in a single autonomous system.

Figure 288: MPLS example topology



Configuration

For static LSPs, there is no need for an IGP.

For LSPs that are set up manually, the first step is to enable MPLS and RSVP on all network interfaces that will be used to carry LSPs. On PE-1, the following interfaces are added to the MPLS and RSVP protocol instances:

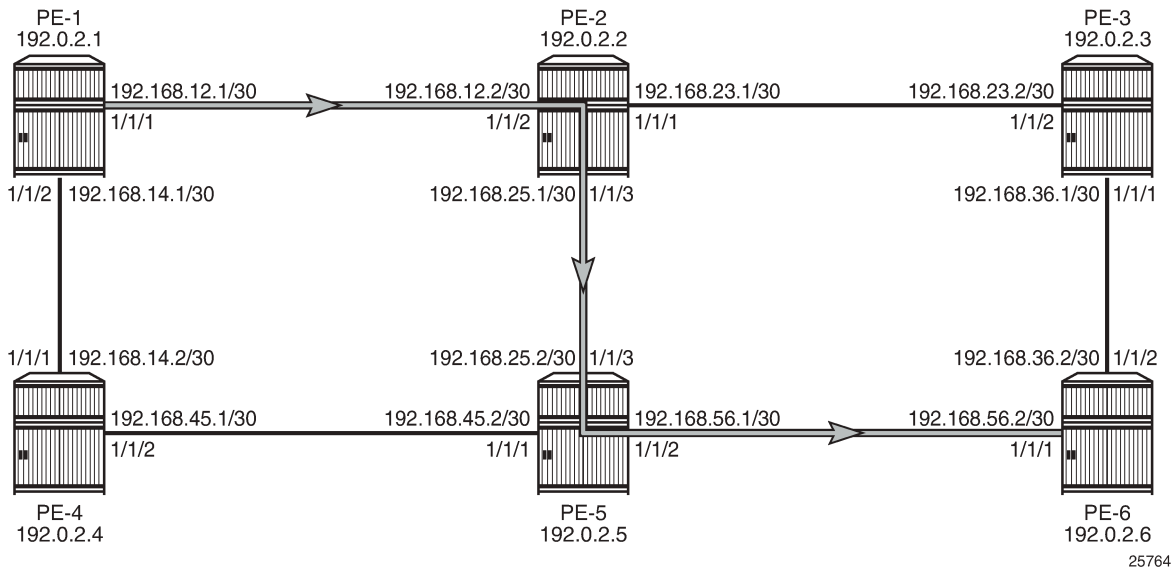
```
# on PE-1:configure {
  router "Base" {
    mpls {
      admin-state enable
      interface "int-PE-1-PE-2" {
      }
      interface "int-PE-1-PE-4" {
      }
    }
    rsvp {
      admin-state enable
      interface "int-PE-1-PE-2" {
      }
      interface "int-PE-1-PE-4" {
      }
    }
  }
}
```

For manually configured LSPs, any interface used by the static LSP must be added into the MPLS and RSVP protocol instances, even though RSVP is not actually used to signal labels. The following error messages are raised on PE-1 when attempting to add the interfaces into the MPLS protocol instance without configuring RSVP:

```
*[ex:configure router "Base"]
A:admin@PE-1# commit
MINOR: MGMT_CORE #4001: configure router "Base" mpls - RSVP needs to be present along with the
MPLS - configure router "Base" rsvp
MINOR: MGMT_CORE #4001: configure router "Base" mpls interface "system" - RSVP interface needs
to be present along with the MPLS interface - configure router "Base" rsvp interface "system"
MINOR: MGMT_CORE #4001: configure router "Base" mpls interface "int-PE-1-PE-2" - RSVP interface
needs to be present along with the MPLS interface - configure router "Base" rsvp interface
"int-PE-1-PE-2"
MINOR: MGMT_CORE #4001: configure router "Base" mpls interface "int-PE-1-PE-4" - RSVP interface
needs to be present along with the MPLS interface - configure router "Base" rsvp interface
"int-PE-1-PE-4"
```

As an example, a static LSP will be created starting from PE-1, running over PE-2 and PE-5, then terminating on PE-6 as shown in [Figure 289: Static LSP running over PE-1, PE-2, PE-5, PE-6](#).

Figure 289: Static LSP running over PE-1, PE-2, PE-5, PE-6



Verify the acceptable label range for use with static configurations for each node; as follows:

```
[/]
A:admin@PE-1# show router mpls-labels label-range

=====
Label Ranges
=====
Label Type      Start Label End Label  Aging    Available  Total
-----
Static          32          18431    -         18400     18400
Dynamic        18432       524287    0         505856    505856
  Seg-Route     0            0         -            0         0
=====
```

The label range for static LSPs extends from the value 32 to 18431. To ensure the labels have not yet been allocated to another configuration, use the command:

```
[/]
A:admin@PE-2# show router mpls-labels label 32 18431 in-use

=====
MPLS Labels from 32 to 18431 (In-use)
=====
Label           Label Type      Label Owner
-----
In-use labels (Owner: All) in specified range : 0
In-use labels in entire range                 : 0
=====
```

This command shows the number of incoming labels in use. At the iLER, the number of labels in use will remain 0 even after the static LSP has been configured where the iLER has a push operation for a label. The reason is that the labels shown are relevant to the labels that the router is generating, as for label

swap or pop operations. There is no information shown about labels that other routers are advertising. For the push operation, any label can be used, even if it is not within the label range of the router pushing the label. For the originating router PE-1, the label 100 will be used for the push operation on the interface toward PE-2.

Static LSPs are configured within the MPLS configuration context, but do not rely on dynamic label signaling.

The configuration of the MPLS static LSP head-end PE-1 contains:

- The system IP address of the destination router PE-6 (to).
- A push operation of the out-label 100.
- The interface address facing the current node of the next-hop along the static path, which is PE-2:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      static-lsp "LSP-PE-1-PE-6-static" {
        admin-state enable
        to 192.0.2.6
        push {
          out-label 100
          next-hop 192.168.12.2
        }
      }
    }
  }
}
```

The transit LSRs PE-2 and PE-5 perform swap operations and forward the packet to the manually defined next-hop. On the LSR under the context of the interface on which the incoming LSP arrives, the correct label is selected (label-map) and in this context a swap operation with a new label and the new next-hop is entered. The MPLS configuration on LSR PE-2 is as follows:

```
# on PE-2:
configure {
  router "Base" {
    mpls {
      interface "int-PE-2-PE-1" {
        label-map 100 {
          admin-state enable
          swap {
            out-label 150
            next-hop 192.168.25.2
          }
        }
      }
    }
  }
}
```

The MPLS configuration on LSR PE-5 is as follows:

```
# on PE-5:
configure {
  router "Base" {
    mpls {
      interface "int-PE-5-PE-2" {
        label-map 150 {
          admin-state enable
          swap {
            out-label 200
          }
        }
      }
    }
  }
}
```


On LSR PE-5:

```
[/]
A:admin@PE-5# show router mpls static-lsp transit-terminate transit

=====
MPLS Static LSPs (Transit)
=====
In Label   In Port   Out Label  Out Port   Next Hop           Adm  Opr
-----
150        1/1/3     200        1/1/2     192.168.56.2      Up   Up
-----
LSPs : 1
=====
```

On the terminating router (eLER), the keyword transit-terminate terminate is added, as follows:

```
[/]
A:admin@PE-6# show router mpls static-lsp transit-terminate terminate

=====
MPLS Static LSPs (Terminate)
=====
In Label   In Port   Out Label  Out Port   Next Hop           Adm  Opr
-----
200        1/1/1     n/a        n/a        n/a                Up   Up
-----
LSPs : 1
=====
```

To track the label action associated with the static LSP configuration, the **show router mpls interface label-map** command can be used on all LSRs and eLERs, but not on the iLER.

```
[/]
A:admin@PE-2# show router mpls interface label-map

=====
MPLS Interfaces (Label-Map)
=====
In Label   In I/F     Out Label  Out I/F     Next Hop           Type  Adm  Opr
-----
100        1/1/2     150        1/1/3     192.168.25.2      Static Up   Up
-----
Interfaces : 1
=====
```

```
[/]
A:admin@PE-6# show router mpls interface label-map

=====
MPLS Interfaces (Label-Map)
=====
In Label   In I/F     Out Label  Out I/F     Next Hop           Type  Adm  Opr
-----
200        1/1/1     n/a        n/a        n/a                Static Up   Up
-----
Interfaces : 1
=====
```


The **show router mpls status** command is used to verify each of the LSP types, the number of configured LSPs and whether they originate on, transit through or terminate on the router.

```
[/]
A:admin@PE-1# show router mpls status
=====
MPLS Status
=====
Admin Status           : Up
Oper(V4) State         : Up           Oper(V6) State           : Down
IPv4 Oper Down Reason  : n/a
IPv6 Oper Down Reason  : ipv6TeRtrDown
FRR Object             : Enabled   Resignal Timer          : Disabled
Hold Timer             : 1 seconds Next Resignal           : N/A
Srlg Frr               : Disabled  Srlg Frr Strict         : Disabled
Admin Group Frr        : Disabled
Dynamic Bypass         : Enabled   User Srlg Database      : Disabled
BypassResignalTimer    : Disabled BypassNextResignal     : N/A
LeastFill Min Thd     : 5 percent LeastFill Reopti Thd   : 10 percent
Local TTL Prop         : Enabled   Transit TTL Prop        : Enabled
AB Sample Multiplier   : 1         AB Adjust Multiplier    : 288
Exp Backoff Retry      : Disabled  CSPF On Loose Hop       : Disabled
Lsp Init RetryTimeout  : 30 seconds MBB Pref Current Hops  : Disabled
Logger Event Bundling  : Disabled
Retry on IGP Overload  : Disabled  Resignal on IGP Overload : Disabled
P2mp Resignal Timer    : Disabled  P2mp Next Resignal     : N/A
Sec FastRetryTimer     : Disabled  Static LSP FR Timer     : 30 seconds
P2P Max Bypass Association: 1000
Max Bypass PLR Association: 16
P2PActPathFastRetry    : Disabled  P2MP S2L Fast Retry    : Disabled
In Maintenance Mode    : No
MplsTp                 : Disabled
Next Available Lsp Index : 2
Entropy Label RSVP-TE  : Enabled   Entropy Label SR-TE    : Enabled
PCE Report RSVP-TE     : Disabled  PCE Report SR-TE       : Disabled
PCE Init LSP           : Disabled
SR-TE Resignal Timer   : Disabled  SR-TE Next Resignal    : N/A
SR-TE Resig on IGP Event : Disabled
LSP Self Ping Timeout  : 300 seconds LSP Self Ping Interval : 1 seconds
RSVP-TE LSP Self Ping  : Disabled  Self Ping Timeout Action : retry
=====
MPLS LSP Count
=====
-----

```

	Originate	Transit	Terminate
Static LSPs	1	0	0
Dynamic LSPs	0	0	0
P2P LSPs	0	N/A	N/A
Detour LSPs	0	0	0
P2MP S2Ls	0	0	0
MPLS-TP LSPs	0	0	0
Mesh-P2P LSPs	0	N/A	N/A
One Hop-P2P LSPs	0	N/A	N/A
SR-TE LSPs	0	N/A	N/A
Mesh-P2P SR-TE LSPs	0	N/A	N/A
One Hop-P2P SR-TE LSPs	0	N/A	N/A
PCE Init SR-TE LSPs	0	N/A	N/A

```
=====
```

Penultimate Hop Popping (PHP) can be used with static LSPs. This is achieved by configuring the last LSR PE-5 before the eLER PE-6 to swap the incoming label to implicit-null instead of a specific label value, as follows:

```
# on PE-5:
configure {
  router "Base" {
    mpls {
      interface "int-PE-5-PE-2" {
        label-map 150 {
          swap {
            out-label implicit-null-label
            next-hop 192.168.56.2
          }
        }
      }
    }
  }
}
```

The previous configuration will cause PE-5 to pop the top label from the incoming labeled frame received from PE-2 and send it to PE-6 without adding another outer label. The following command shows out label 3, but label 3 is never actually pushed onto a frame.

```
[/]
A:admin@PE-5# show router mpls static-lsp transit-terminate transit

=====
MPLS Static LSPs (Transit)
=====
In Label   In Port   Out Label  Out Port   Next Hop           Adm  Opr
-----
150        1/1/3    3          1/1/2     192.168.56.2      Up   Up
-----
LSPs : 1
=====
```

If the traffic arriving at PE-5 were IP with a single label, then it would arrive at PE-6 as unlabeled IP traffic.

If the static LSP spans a single hop, for example, from PE-1 to PE-2, the ingress LER PE-1 pushes the implicit-null instead of a label. The configuration on PE-1 is as follows:

```
# on PE-1:
configure {
  router "Base" {
    mpls {
      static-lsp "LSP-PE-1-PE-2-static" {
        admin-state enable
        to 192.0.2.2
        push {
          out-label implicit-null-label
          next-hop 192.168.12.2
        }
      }
    }
  }
}
```

In this case, no MPLS action (swap or pop) is required for this LSP on PE-2.

Conclusion

MPLS provides the capability to establish connection-oriented paths over a connectionless network. The static LSP offers a mechanism to engineer network traffic. In this chapter, the configuration of static LSPs is given together with the associated show output which can be used to verify and troubleshoot.

Topology-Independent Loop-Free Alternate for Link Protection

This chapter describes the Topology-Independent Loop-Free Alternate for Link Protection.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written based on SR OS Release 16.0.R5, but the MD-CLI in the current edition corresponds to SR OS Release 21.2.R1. Topology-Independent Loop-Free Alternate (TI-LFA) is supported from SR OS Release 15.0.R1 for IS-IS and 15.0.R4 for OSPF.

Overview

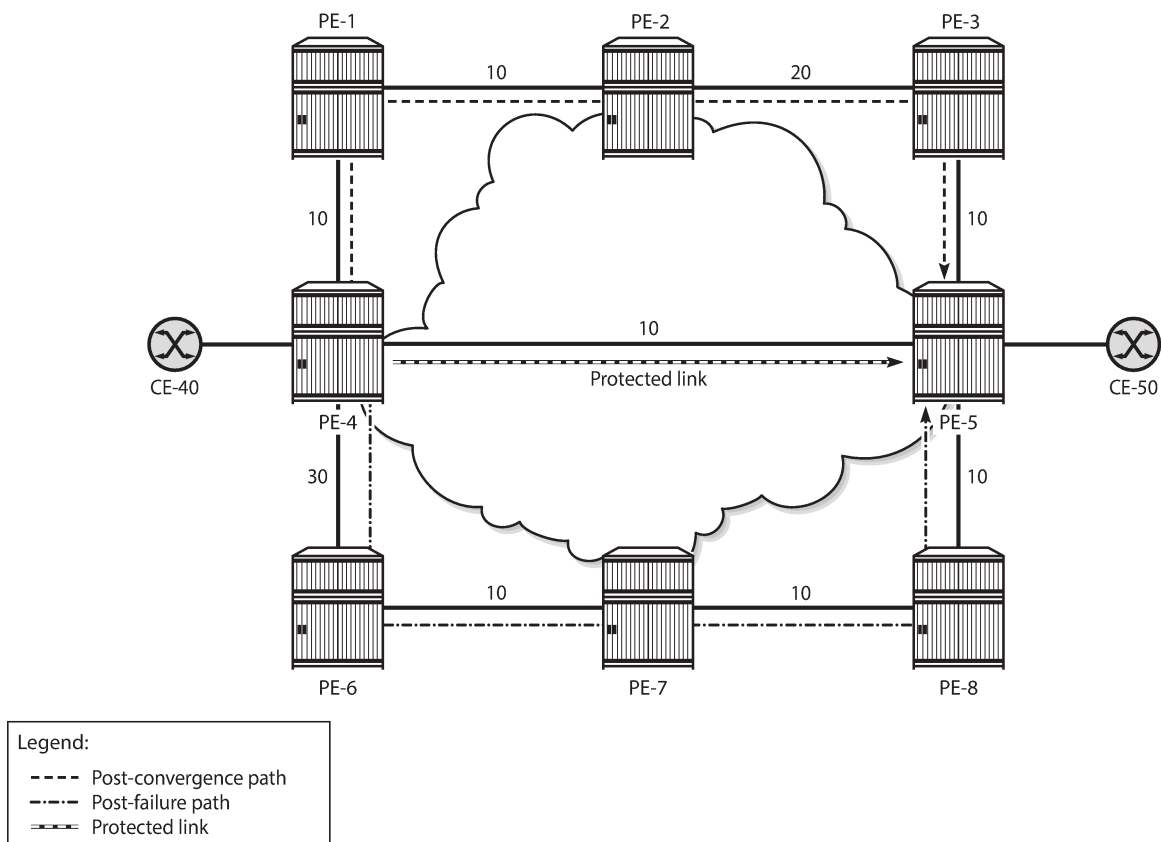
For IP Fast Reroute (FRR), the routers use a precomputed Loop-Free Alternate (LFA) next-hop installed in the FIB until the Shortest Path First (SPF) algorithm runs and the network converges again. The following LFA modes can be applied:

- Regular LFA installs an alternate next-hop in the FIB. Regular LFA provides protection for native IP traffic as well as for Segment Routing (SR) and LDP traffic.
- Remote LFA uses a repair tunnel to a PQ node, which is a node where traffic is not looped back toward the computing node. Remote LFA provides protection for SR and LDP traffic, not for native IP traffic.
- If a computing router has multiple backup next-hop routers, TI-LFA creates a repair tunnel on the post-convergence path so that the post-failure next-hop is avoided, if different from the post-convergence next-hop. In this case, traffic will not be dropped after SPF converges. TI-LFA extends the remote LFA algorithm by computing a backup tunnel where the P and Q nodes do not coincide. TI-LFA uses a repair tunnel to the closest Q node on the post-convergence path. This repair tunnel uses the shortest path to the P node and a source-routed path from the P node to the Q node. TI-LFA provides protection for SR and LDP traffic, not for native IP traffic.

Regular LFA is described in chapter MPLS LDP FRR using ISIS as IGP. Remote LFA and TI-LFA use segment routing to create repair tunnels in cases where there is no regular LFA backup.

[Figure 290: Post-failure LFA path does not match post-convergence path](#) shows the example topology where traffic flows from CE-40 toward CE-50, and a post-failure LFA path that does not match the post-convergence path.

Figure 290: Post-failure LFA path does not match post-convergence path



29352

During normal operation, traffic goes from CE-40 to PE-4 and straight on to PE-5 and CE-50. This is the shortest path between CE-40 and CE-50. Consider the failure of the link between PE-4 and PE-5. This is the protected link. If a failure occurs on the protected link between PE-4 and PE-5, there are two possible backup next-hops from computing node PE-4: PE-1 or PE-6.

When enabling regular LFA on PE-4, two consecutive failovers will occur: the first one, nearly instantaneously, from the preferred path (optimum distance) to the precomputed post-failure path via next-hop PE-6 and the second one, after SPF has run again, from the post-failure path to the post-convergence path via PE-1. When enabling TI-LFA, a single failover will occur, so the computed post-failure path must match the post-convergence path.

The post-convergence path will be from PE-4 to PE-1, PE-2, PE-3, and PE-5, with a path cost of $10 + 10 + 20 + 10 = 50$. With regular LFA, the post-failure path should not use PE-1 as next-hop, because PE-1 would loop back traffic to reach PE-5 via PE-4, through the protected link (which is not allowed).

As described in RFC 5286, the following inequality 1 for link protection must be true for a neighbor next-hop (NH) to provide an LFA. The cost is the optimum distance between the nodes:

$$\text{cost}(\text{NH}, \text{Destination}) < \text{cost}(\text{NH}, \text{Source}) + \text{cost}(\text{Source}, \text{Destination})$$

For next-hop PE-1, the following LFA inequality 1 is false on the calculating node PE-4, indicating that no regular LFA path is possible via PE-1:

$$\text{cost}(PE-1,PE-5) < \text{cost}(PE-1,PE-4) + \text{cost}(PE-4,PE-5)$$

$$(10 + 10) < 10 + 10 \rightarrow \text{False}$$

For next-hop PE-6, the following LFA inequality 1 is true on the calculating node PE-4, indicating that a regular LFA path is possible via PE-6:

$$\text{cost}(PE-6,PE-5) < \text{cost}(PE-6,PE-4) + \text{cost}(PE-4,PE-5)$$

$$(10 + 10 + 10) < 30 + 10 \rightarrow \text{True}$$

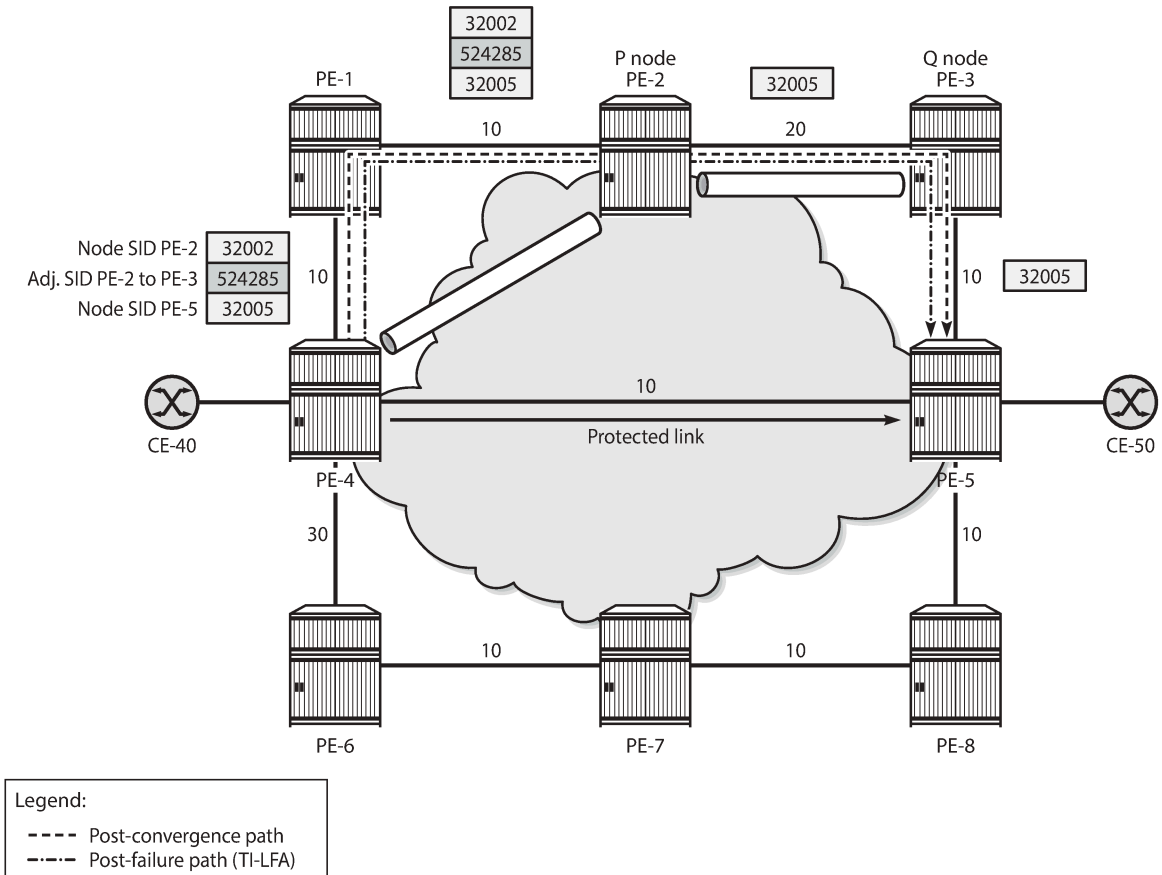
Because of the higher metric between PE-4 and PE-6 (30), PE-6 will not loop back traffic via PE-4: the path cost from PE-6 to PE-5 via PE-4 = 30 + 10 = 40, while the path cost from PE-6 to PE-5 via PE-7 and PE-8 = 10 + 10 + 10 = 30. So, PE-6 will forward the traffic to PE-7, PE-8, and PE-5.

For these reasons, the post-failure path uses PE-6 as regular LFA next-hop.

TI-LFA ensures that traffic is forwarded in a tunnel to the closest Q node, where it will not be looped back to PE-4. In this example, PE-3 is the Q node and it is one hop away from P node PE-2.

With TI-LFA enabled, additional labels are pushed to ensure that the post-failure next-hop matches the post-convergence next-hop. When the protected link between PE-4 and PE-5 fails, PE-4 pushes the node SID of PE-2 as top label plus the adjacency SID of the PE-2 to PE-3 link as an extra label. The bottom label is the node SID of the destination PE-5, which is present in any packet to PE-5 (located on the primary path); see [Figure 291: Post-failure TI-LFA path matches post-convergence path](#).

Figure 291: Post-failure TI-LFA path matches post-convergence path



29353

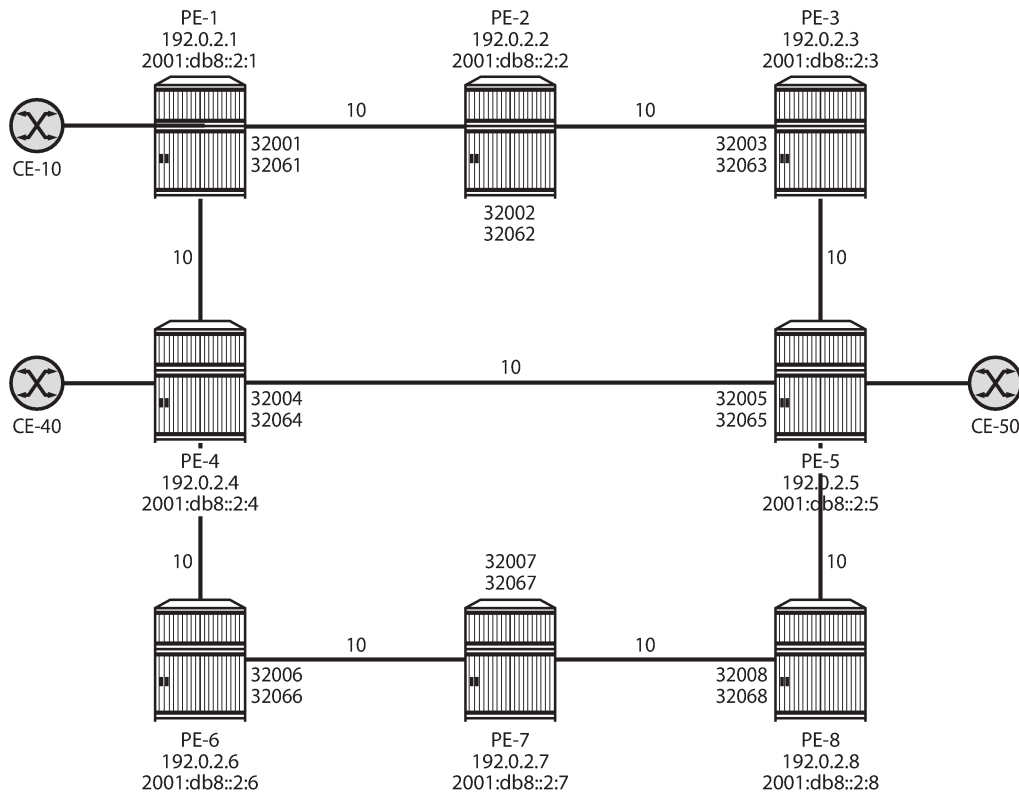
In this chapter, the following LFA modes are described and configured:

- Regular LFA
- Remote LFA
- TI-LFA

Configuration

Figure 292: Example topology shows the example topology, but that will be reduced in the first two scenarios. The default metric of all links is 10, but that may be configured with a different value afterward.

Figure 292: Example topology



29354

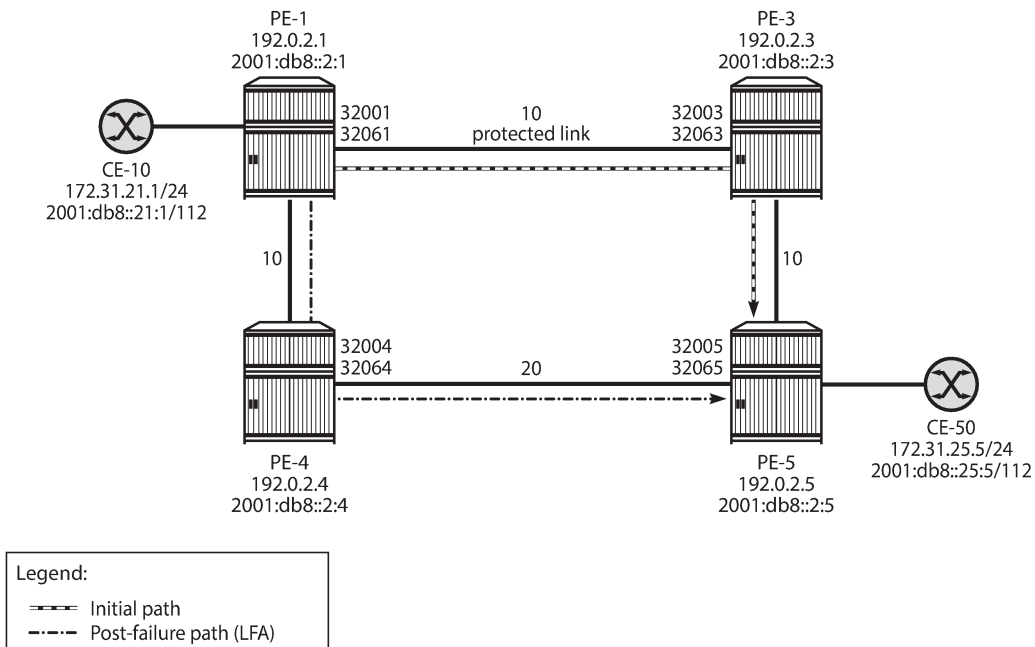
The initial configuration includes the following:

- Cards, MDAs, ports
- Dual-stack router interfaces (IPv4/IPv6)
- IS-IS as IGP on the router interfaces. The metric is 10, but that may be configured otherwise.
- Segment routing (SR-ISIS) with node SIDs 3200x for IPv4 and 3206x for IPv6 system addresses.

Regular LFA

Figure 293: Example topology with regular LFA configured on PE-4 shows the example topology reduced to four PEs. Without a failure of the protected link, traffic from CE-10 to CE-50 is sent via PE-3. The protected link is the link between PE-1 and PE-3 and the LFA path after failure goes via next-hop PE-4.

Figure 293: Example topology with regular LFA configured on PE-4



29355

The IGP metric on the interface between PE-4 and PE-5 is 20, as follows:

```
# on PE-4:
configure {
  router "Base" {
    isis 0 {
      interface "int-PE-4-PE-5" {
        level 1 {
          metric 20
        }
        level 2 {
          metric 20
        }
      }
    }
  }
}
```

```
#on PE-5:
configure {
  router "Base" {
    isis 0 {
      interface "int-PE-5-PE-4" {
        level 1 {
          metric 20
        }
        level 2 {
          metric 20
        }
      }
    }
  }
}
```

Regular LFA is configured on the nodes, as follows:

```
# on PE-1, PE-3, PE-4, PE-5:
configure {
  router "Base" {
```

```
isis 0 {
  loopfree-alternate {
  }
}
```

In the normal situation, without failures, the preferred traffic path from CE-10 to CE-50 is via PE-1, PE-3, and PE-5 with a cost (optimum distance) of 10 + 10 = 20. When the link between PE-1 and PE-3 fails, the post-failure LFA path is via PE-1, PE-4, and PE-5 with a cost of 10 + 20 = 30. The following LFA inequality 1 is true, so PE-4 is a valid LFA next-hop:

$$\text{cost}(\text{newNH}, \text{Destination}) < \text{cost}(\text{newNH}, \text{Source}) + \text{cost}(\text{Source}, \text{Destination})$$

$$\text{cost}(\text{PE-4}, \text{PE-5}) < \text{cost}(\text{PE-4}, \text{PE-1}) + \text{cost}(\text{PE-1}, \text{PE-5})$$

$$20 < 10 + (10 + 10) \rightarrow \text{True}$$

The route table on PE-1 for prefix 192.0.2.5 shows that the next-hop is 192.168.13.2 on PE-3 for the preferred path with metric 20; the LFA next-hop is 192.168.14.2 on PE-4 for the post-failure path with metric 30, as follows:

```
[/]
A:admin@PE-1# show router route-table 192.0.2.5 alternative
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type  Proto  Age           Pref
  Next Hop[Interface Name]                       Metric
  Alt-NextHop                                     Alt-
  Metric
-----
192.0.2.5/32                                     Remote ISIS   00h33m04s  15
  192.168.13.2                                   20
  192.168.14.2 (LFA)                             30
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      Backup = BGP backup route
      LFA = Loop-Free Alternate nexthop
      S = Sticky ECMP requested
=====
```

The following FP tunnel table on PE-1 shows the SR-ISIS label 32005, which is the node SID of PE-5 for prefix 192.0.2.5/32. The same label 32005 is used for the LFA post-failure path indicated with (B) for FRR backup.

```
[/]
A:admin@PE-1# show router fp-tunnel-table 1 192.0.2.5/32
=====
IPv4 Tunnel Table Display
Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                               Protocol  Tunnel-ID
  Lbl
  NextHop                                   Intf/Tunnel
  Lbl   (backup)
```

```

NextHop (backup)
-----
192.0.2.5/32                               SR-ISIS-0           524301
 32005
 192.168.13.2                             1/1/3:1000
 32005
 192.168.14.2(B)                         1/1/2:1000
-----
Total Entries : 1
=====

```

The following FP tunnel table on PE-1 shows the SR-ISIS label 32065, which is the node SID of PE-5 for prefix 2001:db8::2:5/128. The same label 32065 is used for the LFA post-failure path.

```

[/]
A:admin@PE-1# show router fp-tunnel-table 1 2001:db8::2:5/128

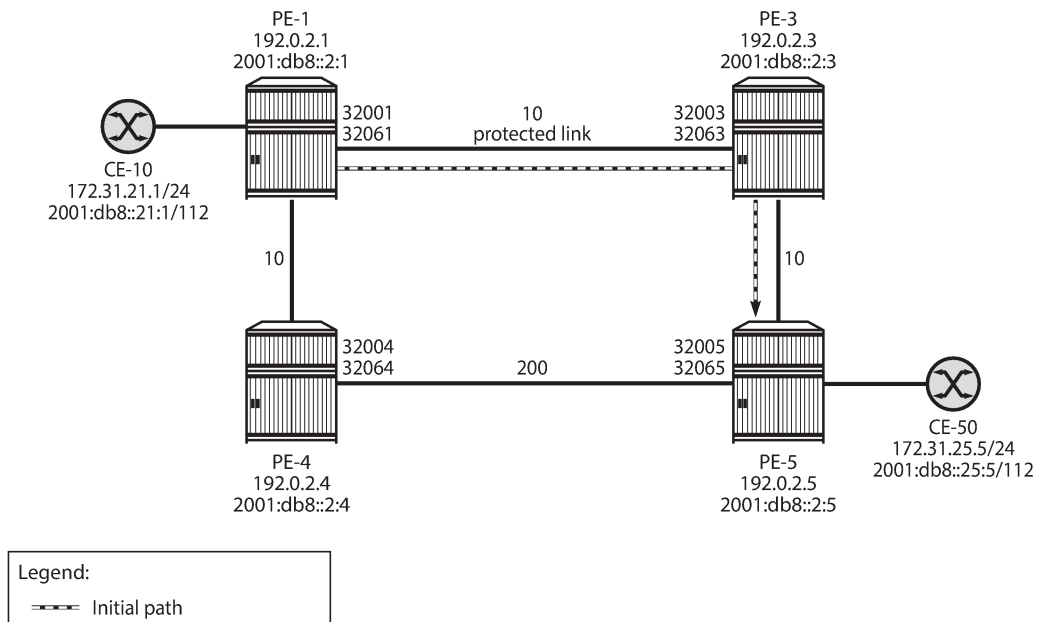
=====
IPv6 Tunnel Table Display

Legend:
Label stack is ordered from bottom most to top-most
B - FRR Backup
=====
Destination                               Protocol           Tunnel-ID
Lbl
NextHop
Lbl (backup)
NextHop (backup)
-----
2001:db8::2:5/128                         SR-ISIS-0         524302
 32065
 fe80::618:1ff:fe01:3-"int-PE-1-PE-3"    1/1/3:1000
 32065
 fe80::61c:1ff:fe01:1-"int-PE-1-PE-4"(B) 1/1/2:1000
-----
Total Entries : 1
=====

```

Figure 294: No post-failure LFA path when PE-4 loops back traffic shows that no backup LFA next-hop exists when the metric on the interface between PE-4 and PE-5 is increased to 200.

Figure 294: No post-failure LFA path when PE-4 loops back traffic



29356

The following configures the metric on the interface between PE-4 and PE-5 to a value of 200:

```
# on PE-4:
configure {
  router "Base" {
    isis 0 {
      interface "int-PE-4-PE-5" {
        level 1 {
          metric 200
        }
        level 2 {
          metric 200
        }
      }
    }
  }
}
```

```
# on PE-5:
configure {
  router "Base" {
    isis 0 {
      interface "int-PE-5-PE-4" {
        level 1 {
          metric 200
        }
        level 2 {
          metric 200
        }
      }
    }
  }
}
```

When the metric on the interface between PE-4 and PE-5 is increased to a value that exceeds the sum of the metrics on the path from PE-4 to PE-1 and the path from PE-1 to PE-5 (via PE-3), the computing node PE-1 cannot calculate a regular LFA path to protect the PE-5 prefixes. The following LFA inequality 1 is false:

$$\text{cost}(PE-4,PE-5) < \text{cost}(PE-4,PE-1) + \text{cost}(PE-1,PE-5)$$

200 < 10 + (10 + 10) --> False

If the preferred path cannot be used because of a failure, such as a link failure between PE-1 and PE-3, a micro-loop is created between PE-4 and PE-5 until convergence is completed. The following output shows that no LFA next-hop is available on PE-1:

```
[/]
A:admin@PE-1# show router route-table 192.0.2.5 alternative
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
  Next Hop[Interface Name]          Alt-NextHop      Metric  Alt-
                                          Metric
-----
192.0.2.5/32                      Remote ISIS  00h05m26s 15
  192.168.13.2                      20
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
       Backup = BGP backup route
       LFA = Loop-Free Alternate nexthop
       S = Sticky ECMP requested
=====
```

Remote LFA

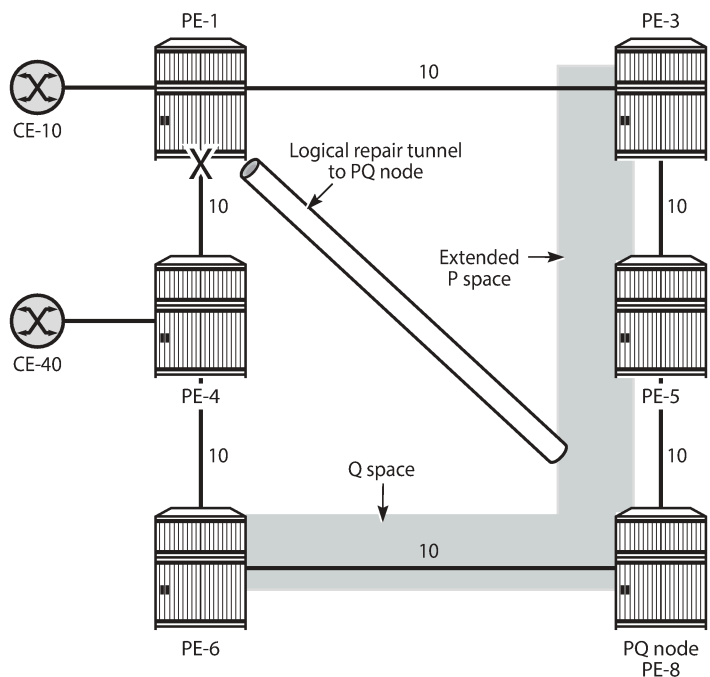
[Figure 295: Example topology for remote LFA](#) shows the example topology with six nodes in a ring. Traffic from CE-10 to CE-40 is preferably sent via PE-1 to PE-4.

The endpoint node of the repair tunnel for remote LFA (RLFA) is the PQ node, which is in the intersection of the extended P space of source PE-1 and the Q space of destination PE-4.

- The P space of PE-1 is the set of routers reachable on the shortest SPF path from the computing node PE-1, without using the protected link between PE-1 and PE-4; that is, SPF computed by PE-1 and rooted from PE-1. In this example, PE-3 and PE-5 are in the P space of PE-1.
- The extended P space of PE-1 is the set of routers, calculated by PE-1, in the P space of the next-hop router PE-3. An additional SPF computation by PE-1 and rooted from PE-3 results in P nodes PE-3, PE-5, and PE-8. The extended P space increases the repair coverage.
- The Q space of PE-4 is the set of routers that can reach the destination router PE-4 using the shortest path, without using the protected link; that is, reverse SPF computed by PE-1 and rooted from PE-4, resulting in Q nodes PE-6 and PE-8.
- PQ routers are in the intersection of the extended P space and the Q space; in this case, the only PQ node is PE-8.
- Repair tunnels are shortest path SR tunnels from the computing node PE-1 to the PQ router; in this case, from PE-1 to PE-8.

Figure 296: PQ node in remote LFA shows the extended P space of PE-1, comprising nodes PE-3, PE-5, and PE-8, and the Q space of PE-4, comprising nodes PE-6 and PE-8. In the event of a link failure, PE-1 will push the node SID of PE-8, along with the node SID of PE-4, and forward the packet toward the backup next-hop PE-8.

Figure 296: PQ node in remote LFA



29358

The following shows the SR LFA coverage on PE-1; the five other node SIDS are all protected: one with regular LFA and the remaining four with remote LFA (in the column RLFA). Besides the node SIDs, the

adjacency SIDs toward the direct neighbors PE-3 and PE-4 are protected using RLFA. The LFA coverage is the same for IPv4 and IPv6.

```
[/]
A:admin@PE-1# show router isis sr-lfa-coverage

=====
Rtr Base ISIS Instance 0 SR LFA Coverage
=====
MT-ID  SidType      Level Proto LFA      RLFA      TILFA      Coverage
-----
0      node-sid     L1    ipv4  1(20%)  4(80%)    0(0%)      5/5(100%)
0      node-sid     L1    ipv6  1(20%)  4(80%)    0(0%)      5/5(100%)
---snip---
0      adj-sid     L1L2  ipv4  0(0%)   2(100%)   0(0%)      2/2(100%)
0      adj-sid     L1L2  ipv6  0(0%)   2(100%)   0(0%)      2/2(100%)
=====
```

The repair tunnel from PE-1 to PQ node PE-8 uses node SID 32008 for IPv4 and 32068 for IPv6.

The fifth entry in the following FP tunnel table shows that destination 192.0.2.8/32 of PE-8 is protected with regular LFA. The only label is 32008, which is the node SID of PE-8. All other destinations in the table are protected with remote LFA, having two node SID labels for the RLFA path, such as 32004/32008 for prefix 192.0.2.4 with next-hop 192.168.13.2 on PE-3. This means that the top label 32008 is pushed by PE-1 to match the repair-tunnel going via PE-3 to PQ-node PE-8. From PE-8 onward, the bottom label 32004 is used toward PE-4. Likewise, the other destinations in the list have top label 32008, so a tunnel is established to PE-8. The output is similar for IPv6.

```
[/]
A:admin@PE-1# show router fp-tunnel-table 1

=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol      Tunnel-ID
Lbl
NextHop                                     Intf/Tunnel
Lbl      (backup)
NextHop  (backup)
-----
192.0.2.3/32                                SR-ISIS-0    524295
32003
192.168.13.2                                1/1/3:1000
32003/32008
192.168.14.2(B)                             1/1/2:1000
192.0.2.4/32                                SR-ISIS-0    524299
32004
192.168.14.2                                1/1/2:1000
32004/32008
192.168.13.2(B)                           1/1/3:1000
192.0.2.5/32                                SR-ISIS-0    524301
32005
192.168.13.2                                1/1/3:1000
32005/32008
192.168.14.2(B)                             1/1/2:1000
192.0.2.6/32                                SR-ISIS-0    524311
32006
```



```

192.168.14.2          1/1/2:1000
32006/32008
192.168.13.2(B)     1/1/3:1000
192.0.2.8/32        SR-ISIS-0  524312
32008
192.168.13.2          1/1/3:1000
32008
192.168.14.2(B)     1/1/2:1000
192.168.13.2/32     SR          524309
3
192.168.13.2          1/1/3:1000
32003/32008
192.168.14.2(B)     1/1/2:1000
192.168.14.2/32     SR          524297
3
192.168.14.2          1/1/2:1000
32004/32008
192.168.13.2(B)     1/1/3:1000
-----
Total Entries : 7
-----
=====

```

TI-LFA

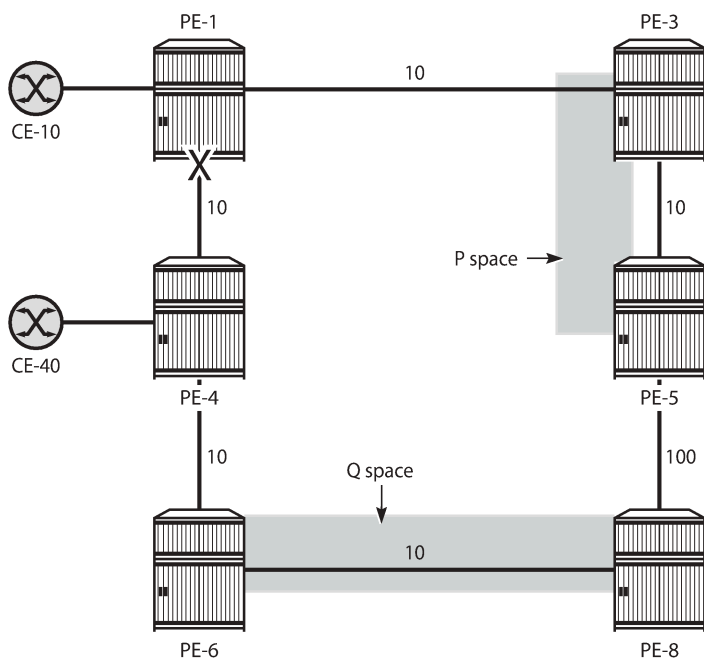
The following two use cases are described in this section:

- Directed LFA where the extended P space and the Q space do not overlap
- Extension of the RLFA algorithm to compute a repair path using directed LFA, but ensuring that the post-failure path matches the post-convergence path

Directed LFA

[Figure 297: Extended P space of PE-1 and Q space of PE-4 are one hop apart](#) shows the example topology with increased metric between PE-5 and PE-8, reducing the extended P space to PE-3 and PE-5, so there is no PQ node.

Figure 297: Extended P space of PE-1 and Q space of PE-4 are one hop apart



29359

There is no remote LFA repair tunnel. No Q routers are on the shortest path from the computing router, and the P routers are not in the reverse SPF of the endpoint of the protected link. However, TI-LFA can calculate a repair tunnel in case the gap is only one or two hops. TI-LFA is enabled using the following command:

```
# on PE-1, PE-3, PE-4, PE-5, PE-6, PE-8:
configure {
  router "Base" {
    isis 0 {
      loopfree-alternate {
        ti-lfa {
          max-sr-frr-labels 2
        }
      }
    }
  }
}
```

Table 17: Values of the `max-sr-frr-labels` parameter in TI-LFA lists the possible values of the `max-sr-frr-labels` parameter. This parameter is used to specify the maximum number of labels that the TI-LFA backup next-hop can use.

Table 17: Values of the `max-sr-frr-labels` parameter in TI-LFA

Max. SR-FRR labels	LFA behavior
0	Regular LFA: TI-LFA backup restricted to next-hop that does not require a repair tunnel, so PQ node is a neighbor of the computing node.
1	Remote LFA: extended P space and Q space intersect and the repair tunnel requires 1 FRR label:

Max. SR-FRR labels	LFA behavior
	<ul style="list-style-type: none"> Node SID to PQ router
2 (default)	TI-LFA with extended P space and Q space one hop apart: <ul style="list-style-type: none"> Node SID to P router Adjacency SID on P router to Q router
3	TI-LFA with extended P space and Q space two hops apart: <ul style="list-style-type: none"> Node SID to P router Two adjacency SIDs to Q router

In this case, the extended P space and the Q space are one hop apart and TI-LFA calculates a post-failure path that consists of a repair tunnel to P router PE-5 (node SID 32005 for IPv4) and an adjacency SID toward Q router PE-8. For routes from PE-1 to PE-4, the LFA route has two additional labels combined with the bottom label that is the node SID of PE-4 (32004), which is also used for the primary path. The top label is the node SID of P router PE-5 (32005); the next label is the adjacency SID on PE-5 toward PE-8 (524285).

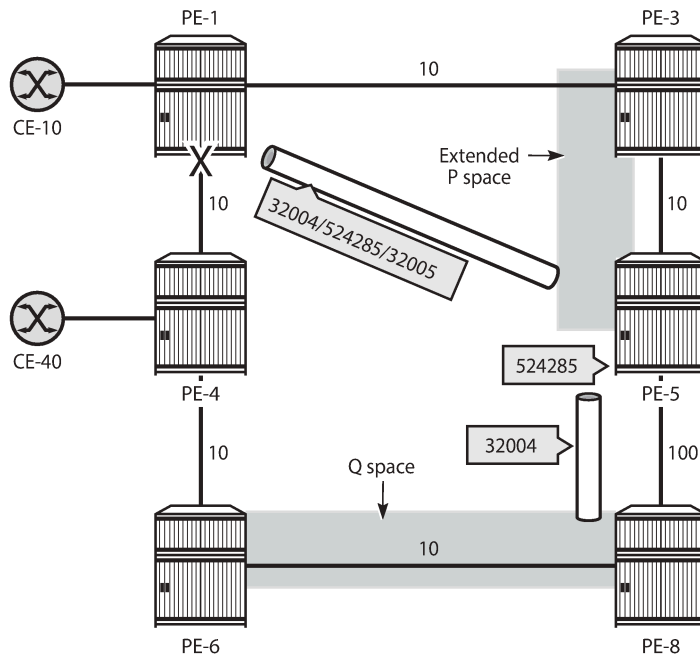
```
[/]
A:admin@PE-1# show router fp-tunnel-table 1 192.0.2.4/32

=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol      Tunnel-ID
Lbl
  NextHop                                    Intf/Tunnel
Lbl      (backup)
  NextHop  (backup)
-----
192.0.2.4/32                                SR-ISIS-0    524299
32004
  192.168.14.2                               1/1/2:1000
32004/524285/32005
  192.168.13.2(B)                             1/1/3:1000
-----
Total Entries : 1
=====
```

Figure 298: Directed LFA with P router and Q router one hop apart shows the directed LFA path from source PE-1 to P router PE-5 (node SID), the adjacency SID from P router PE-5 to Q router PE-8, and the node SID of destination PE-4. P router PE-5 uses the adjacency SID for forwarding, but only sends the packets with the node SID of PE-4 (32004).

Figure 298: Directed LFA with P router and Q router one hop apart

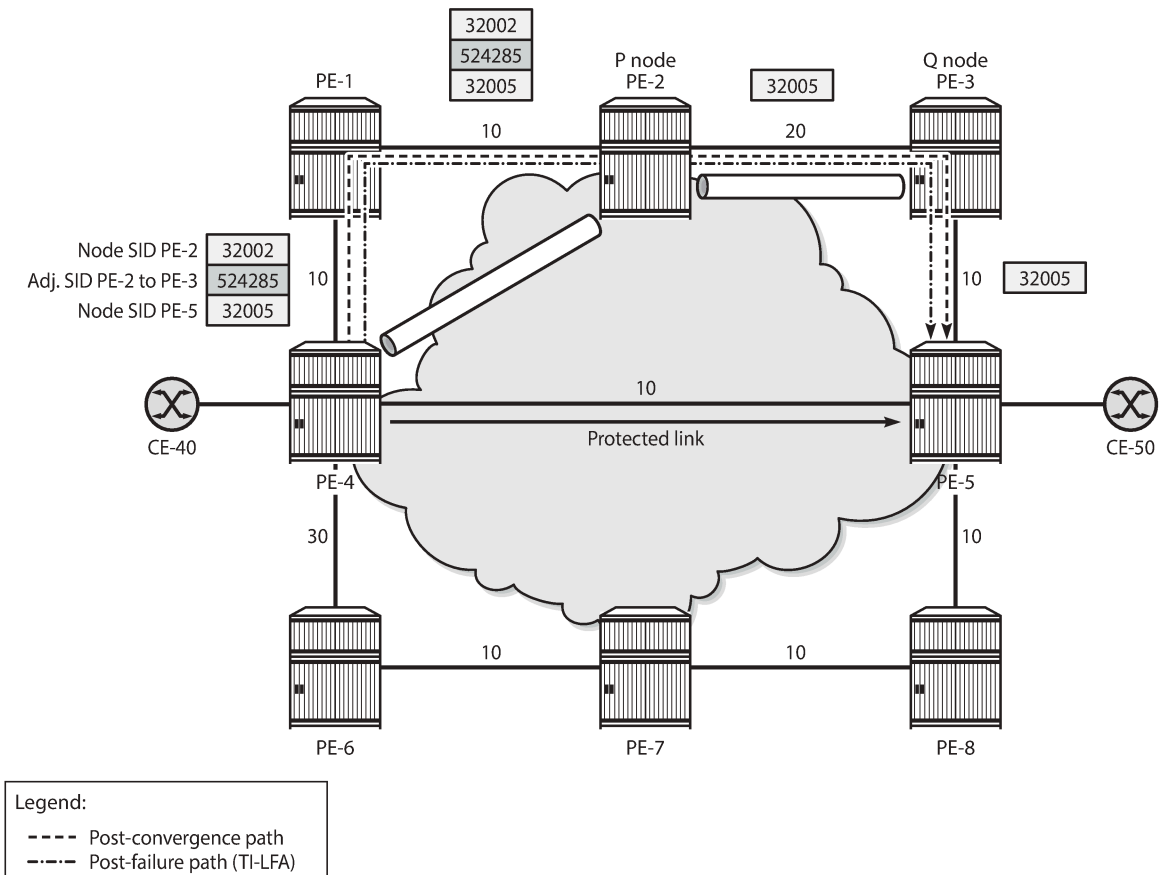


29360

TI-LFA for coinciding post-failure and post-convergence paths

Figure 299: Post-failure TI-LFA path coincides with post-convergence path is the same as Figure 291: Post-failure TI-LFA path matches post-convergence path and is repeated here for readability. The router interfaces have IGP metric 10 by default, except for the interfaces between PE-2 and PE-3 that have metric 20, and the interfaces between PE-4 and PE-6 that have metric 30. As in Figure 298: Directed LFA with P router and Q router one hop apart, Figure 299: Post-failure TI-LFA path coincides with post-convergence path shows the different tunnels used for the TI-LFA path. TI-LFA ensures that the post-failure path coincides with the post-convergence path by adding additional labels: the node SID 32002 (or 32062 for IPv6) to P router PE-2, the adjacency SID on PE-2 for the interface toward Q router PE-3, and the node SID 32005 (or 32065 for IPv6) toward the destination PE-5.

Figure 299: Post-failure TI-LFA path coincides with post-convergence path



29353

Regular LFA coverage

For a better comparison, the regular LFA coverage is calculated first. Without remote LFA and TI-LFA enabled, the LFA coverage is limited. The following command disables remote LFA and TI-LFA on all nodes, while regular LFA remains enabled:

```
# on all nodes:
configure {
  router "Base" {
    isis 0 {
      loopfree-alternate {
        delete remote-lfa
        delete ti-lfa
      }
    }
  }
}
```

The SR LFA coverage on PE-4 only protects node SIDs and adjacency SIDs that can be protected with regular LFA, as follows:

[/]

```
A:admin@PE-4# show router isis sr-lfa-coverage
```

```
=====
```

Rtr Base ISIS Instance 0 SR LFA Coverage							
MT-ID	SidType	Level	Proto	LFA	RLFA	TILFA	Coverage
0	node-sid	L1	ipv4	5(71%)	0(0%)	0(0%)	5/7(71%)
0	node-sid	L1	ipv6	5(71%)	0(0%)	0(0%)	5/7(71%)
---snip---							
0	adj-sid	L1L2	ipv4	2(66%)	0(0%)	0(0%)	2/3(66%)
0	adj-sid	L1L2	ipv6	2(66%)	0(0%)	0(0%)	2/3(66%)

```
=====
```

The following shows that no LFA paths exist on PE-4 for destinations 192.0.2.1 (PE-1), 192.0.2.2 (PE-2), and 192.168.14.1 (PE-1).

```
[/]
A:admin@PE-4# show router fp-tunnel-table 1
```

```
=====
```

IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup

```
=====
```

Destination	Protocol	Tunnel-ID
Lbl		Intf/Tunnel
NextHop		
Lbl (backup)		
NextHop (backup)		
192.0.2.1/32	SR-ISIS-0	524291
32001		
192.168.14.1		1/1/1:1000
192.0.2.2/32	SR-ISIS-0	524319
32002		
192.168.14.1		1/1/1:1000
192.0.2.3/32	SR-ISIS-0	524293
32003		
192.168.45.2		1/1/3:1000
32003		
192.168.46.2(B)		1/1/2:1000
192.0.2.5/32	SR-ISIS-0	524299
32005		
192.168.45.2		1/1/3:1000
32005		
192.168.46.2(B)		1/1/2:1000
192.0.2.6/32	SR-ISIS-0	524311
32006		
192.168.46.2		1/1/2:1000
32006		
192.168.45.2(B)		1/1/3:1000
192.0.2.7/32	SR-ISIS-0	524323
32007		
192.168.45.2		1/1/3:1000
32007		
192.168.46.2(B)		1/1/2:1000
192.0.2.8/32	SR-ISIS-0	524313
32008		
192.168.45.2		1/1/3:1000
32008		

192.168.46.2(B)		1/1/2:1000
192.168.14.1/32	SR	524317
3		
192.168.14.1		1/1/1:1000
192.168.45.2/32	SR	524321
3		
192.168.45.2		1/1/3:1000
32005		
192.168.46.2(B)		1/1/2:1000
192.168.46.2/32	SR	524309
3		
192.168.46.2		1/1/2:1000
32006		
192.168.45.2(B)		1/1/3:1000

Total Entries : 10		

=====		

For destination 192.0.2.5, the post-failure path has next-hop 192.168.46.2 on PE-6, so the post-failure path does not coincide with the post-convergence path with next-hop 192.168.14.1 on PE-1. The path cost of the post-convergence path from PE-4 to PE-5 (via PE-1, PE-2, and PE-3) equals $10 + 10 + 20 + 10 = 50$; the path cost of the post-failure path from PE-4 to PE-5 (via PE-6, PE-7, and PE-8) equals $30 + 10 + 10 + 10 = 60$.

TI-LFA enabled

TI-LFA can be configured with remote LFA enabled or disabled. The following command configures remote LFA and TI-LFA (with default max-sr-frr-labels 2).

```
# on all nodes:
configure {
  router "Base" {
    isis 0 {
      loopfree-alternate {
        remote-lfa {
        }
      }
      ti-lfa {
        max-sr-frr-labels 2
      }
    }
  }
}
```

With TI-LFA enabled, the SR LFA coverage increases to 100%, as follows. For almost all destinations, the LFA protection is now using TI-LFA, even when regular LFA was possible before. The advantage is that TI-LFA ensures the post-failure path coincides with the post-convergence path.

If there is regular LFA protection via a path that does not coincide with the post-convergence path, that regular LFA protection will only change to TI-LFA protection when max-sr-frr-labels allows the needed number of labels (tunnels) to force the TI-LFA protection to the post-convergence path. The same applies for remote LFA protection.

```
[/]
A:admin@PE-4# show router isis sr-lfa-coverage

=====
Rtr Base ISIS Instance 0 SR LFA Coverage
=====
MT-ID  SidType      Level Proto LFA      RLFA     TILFA     Coverage
-----
```

```

0      node-sid    L1    ipv4  0(0%)  0(0%)  7(100%)  7/7(100%)
0      node-sid    L1    ipv6  0(0%)  0(0%)  7(100%)  7/7(100%)
---snip---
0      adj-sid    L1L2  ipv4  0(0%)  0(0%)  3(100%)  3/3(100%)
0      adj-sid    L1L2  ipv6  0(0%)  0(0%)  3(100%)  3/3(100%)
=====

```

The following FP tunnel table shows that prefixes 192.0.2.1 (PE-1), 192.0.2.2 (PE-2), and 192.168.14.1 (PE-1) are now protected too. For destination 192.0.2.5 (PE-5), the next-hop now is 192.168.14.1, which is also the next-hop on the post-convergence path to PE-5 via PE-1, PE-2, and PE-3. The top label 32002 is the node SID of PE-2, the label 524285 is the adjacency SID on PE-2 for the interface toward PE-3, and the bottom label 32005 is the node SID to reach the destination PE-5.

```

[/]
A:admin@PE-4# show router fp-tunnel-table 1

=====
IPv4 Tunnel Table Display

Legend:
Label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol          Tunnel-ID
 Lbl
  NextHop
 Lbl      (backup)                          Intf/Tunnel
  NextHop  (backup)
-----
192.0.2.1/32                                SR-ISIS-0        524291
 32001
  192.168.14.1                              1/1/1:1000
  32001/524285/32003
  192.168.45.2(B)                            1/1/3:1000
192.0.2.2/32                                SR-ISIS-0        524319
 32002
  192.168.14.1                              1/1/1:1000
  32002/524285/32003
  192.168.45.2(B)                            1/1/3:1000
192.0.2.3/32                                SR-ISIS-0        524293
 32003
  192.168.45.2                              1/1/3:1000
  32003/524285/32002
  192.168.14.1(B)                            1/1/1:1000
192.0.2.5/32                                SR-ISIS-0        524299
 32005
  192.168.45.2                              1/1/3:1000
  32005/524285/32002
  192.168.14.1(B)                            1/1/1:1000
192.0.2.6/32                                SR-ISIS-0        524311
 32006
  192.168.46.2                              1/1/2:1000
  32006
  192.168.45.2(B)                            1/1/3:1000
192.0.2.7/32                                SR-ISIS-0        524323
 32007
  192.168.45.2                              1/1/3:1000
  32007
  192.168.46.2(B)                            1/1/2:1000
192.0.2.8/32                                SR-ISIS-0        524313
 32008
  192.168.45.2                              1/1/3:1000

```



```

32008
 192.168.46.2(B)
192.168.14.1/32          SR          1/1/2:1000
 3                          524317
 192.168.14.1             1/1/1:1000
32001/524285/32003
 192.168.45.2(B)         1/1/3:1000
192.168.45.2/32          SR          524321
 3                          1/1/3:1000
 192.168.45.2             1/1/1:1000
32005/524285/32002
 192.168.14.1(B)         1/1/1:1000
192.168.46.2/32          SR          524309
 3                          1/1/2:1000
 192.168.46.2             1/1/3:1000
32006
 192.168.45.2(B)         1/1/3:1000
-----
Total Entries : 10
-----
=====

```

The following **tools** command on PE-4 includes detailed information for the LFA protection for destination 192.0.2.5:

```

[/]
A:admin@PE-4# tools dump router isis sr-database prefix 192.0.2.5 detail
=====
Rtr Base ISIS Instance 0 SR Database

Legend:
label stack is ordered from bottom-most to top-most
=====
-----
SID 5
-----
Label           : 32005          Adv System Id   : 1920.0000.2005
Prefix          : 192.0.2.5
Route Level     : 1           MT Id           : 0
Rtm Preference  : 15          Ttm Preference  : 11
Metric          : 10          Last Action     : LfaNhops
Num Ip NextHop  : 1           Num SR-Tnl NextHop : 1
Mtu             : 8970
Mtu Prim        : 8982        Mtu Backup      : 8982
Exclude from LFA : 0         LFA Type       : TI LFA
Duplicate Pending : 0         Tunnel Active State : Reported/Ack
SR Error        : SR_ERR_OK

LFA NextHop IP  : 192.168.14.1
LFA IsTunl      : N
LFA GIfId/TunlType : 1         LFA IfId/LspId   : 2
LFA PgId        : 0           LFA Adv Node     : False
LFA Labels    : 32005/524285/32002

NHOP: IP           IsTunl GIfId/  IfId/ PgId  IsAdv Label  IsLfaX
                TunlType LspId
-----
192.168.45.2      N      2      3      13      1      32005      0
-----

No. of Entries: 1
-----
LDP = LDP FEC is the SID NH for SR-LDP stitching

```

TI-LFA enabled with max-sr-frr-labels lower than 2

When TI-LFA is configured with max-sr-frr-labels lower than 2, TI-LFA cannot substitute regular or remote LFA where more than 2 tunnel labels are needed for the substitution. Some destinations may remain protected then via regular or remote LFA, and only those destinations that can be protected with TI-LFA with less than 2 tunnel labels will have TI-LFA protection. The following configuration enables TI-LFA with max-sr-frr-labels equal to 1:

```
# on all nodes:
configure {
  router "Base" {
    isis 0 {
      loopfree-alternate {
        remote-lfa {
        }
      }
      ti-lfa {
        max-sr-frr-labels 1
      }
    }
  }
}
```

In the topology of [Figure 299: Post-failure TI-LFA path coincides with post-convergence path](#), for max-sr-frr-labels equal to 1, the SR LFA coverage drops below 100% again, as follows.

```
[/]
A:admin@PE-4# show router isis sr-lfa-coverage

=====
Rtr Base ISIS Instance 0 SR LFA Coverage
=====
MT-ID  SidType      Level Proto LFA      RLFA      TILFA      Coverage
-----
0      node-sid     L1   ipv4  2(28%)  0(0%)    3(42%)    5/7(71%)
0      node-sid     L1   ipv6  2(28%)  0(0%)    3(42%)    5/7(71%)
---snip---
0      adj-sid      L1L2 ipv4   1(33%)  0(0%)    1(33%)    2/3(66%)
0      adj-sid      L1L2 ipv6   1(33%)  0(0%)    1(33%)    2/3(66%)
=====
```

The preceding information can be derived from the FP tunnel table and the SR database as follows. For PE-4, the FP tunnel table shows that there are 10 destinations, 7 nodes and 3 next-hops. 5 out of 7 node destinations and 2 out of 3 next-hop destinations are protected with a backup (B). Node destination 192.0.2.1 (PE-1), and 192.0.2.2 (PE-2), and next-hop destination 192.168.14.1 are no longer protected.

```
[/]
A:admin@PE-4# show router fp-tunnel-table 1

=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                     Protocol      Tunnel-ID
Lbl
  NextHop
Lbl      (backup)                                Intf/Tunnel
```

NextHop (backup)		
192.0.2.1/32 32001 192.168.14.1	SR-ISIS-0	524291 1/1/1:1000
192.0.2.2/32 32002 192.168.14.1	SR-ISIS-0	524319 1/1/1:1000
192.0.2.3/32 32003 192.168.45.2	SR-ISIS-0	524293 1/1/3:1000
192.0.2.5/32 32005 192.168.45.2	SR-ISIS-0	524299 1/1/3:1000
192.0.2.6/32 32006 192.168.46.2(B)	SR-ISIS-0	524311 1/1/2:1000
192.0.2.7/32 32007 192.168.45.2	SR-ISIS-0	524323 1/1/3:1000
192.0.2.8/32 32008 192.168.45.2	SR-ISIS-0	524313 1/1/3:1000
192.168.14.1/32 3 192.168.14.1	SR	524317 1/1/1:1000
192.168.45.2/32 3 192.168.45.2	SR	524325 1/1/3:1000
192.168.46.2/32 3 192.168.46.2(B)	SR	524309 1/1/2:1000
192.168.46.2/32 3 192.168.46.2		1/1/2:1000
192.168.45.2(B)		1/1/3:1000

Total Entries : 10		

=====		

The SR database indicates what type of protection corresponds with the (topmost) label of the destinations in the FP tunnel tabel. Destination 192.0.2.1 (PE-1), 192.0.2.2 (PE-2), and 192.168.14.1 have no backup. Their label indicates that there is no LFA protection (LT = -). Destination 192.0.2.3 (PE-3), 192.0.2.5 (PE-5), and 192.168.45.2 have a backup with a (topmost) label that indicates regular LFA protection (LT = L). So, destination 192.0.2.5 (PE-5) is no longer TI-LFA protected, because that would require 2 tunnel labels, which max-sr-frr-labels=1 prevents. Destination 192.0.2.6 (PE-6), 192.0.2.7 (PE-7), 192.0.2.8 (PE-8), and 192.168.46.2 have a backup with a (topmost) label that indicates TI-LFA protection (LT = T). As these destinations have no TI-LFA tunnel label, their TI-LFA protection does not need tunnels to ensure that the TI-LFA protection is via the post-convergence path.

The following **tools** command on PE-4 includes detailed information for the type of LFA protection that corresponds with a label:

```
[/]
A:admin@PE-4# tools dump router isis sr-database ipv4-unicast
=====
Rtr Base ISIS Instance 0 SR Database
=====
SID  Label  Prefix          Last-act  Lev MT RtmPref TtmPref Metric  IpNh SrNh
Mtu   MtuPrim MtuBk   D xL LT Act AdvSystemId  SrErr
-----
1    32001  192.0.2.1      RemLfaNh 1  0 15    11    10    1    1
8974  8982    -        0 0 - +R 1920.0000.2001 SR_ERR_OK
2    32002  192.0.2.2      RemLfaNh 1  0 15    11    20    1    1
8974  8982    -        0 0 - +R 1920.0000.2002 SR_ERR_OK
3    32003  192.0.2.3      LfaNhops 1  0 15    11    20    1    1
8974  8982    8982    0 0 L +R 1920.0000.2003 SR_ERR_OK
4    32004  192.0.2.4      Local     -  -  -    -    -    -    -
-      -      -        0 - - +R 1920.0000.2004 SR_ERR_OK
5    32005  192.0.2.5      LfaNhops 1  0 15    11    10    1    1
8974  8982    8982    0 0 L +R 1920.0000.2005 SR_ERR_OK
6    32006  192.0.2.6      TnlChange 1  0 15    11    30    1    1
8974  8982    8982    0 0 T +R 1920.0000.2006 SR_ERR_OK
7    32007  192.0.2.7      TnlChange 1  0 15    11    30    1    1
8974  8982    8982    0 0 T +R 1920.0000.2007 SR_ERR_OK
8    32008  192.0.2.8      TnlChange 1  0 15    11    20    1    1
8974  8982    8982    0 0 T +R 1920.0000.2008 SR_ERR_OK
-----
No. of Entries: 8
-----
Lev = route level
IpNh = number of IP next-hops
SrNh = number of SR-tunnel next-hops
D = duplicate pending
xL = exclude from LFA
LT = LFA type (L:LFA, R:RLFA, T:TILFA, n:nodeProtection)
Act = tunnel active state (R:reported, F:failed, +:SR-ack)
=====
```

Independent from the preceding ISIS Segment Routing LFA coverage (per Segment Routing LFA type and per ISIS Level), there is also the ISIS IP-routing LFA coverage (per IP version and per ISIS Level), as follows:

```
[/]
A:admin@PE-4# show router isis lfa-coverage
=====
Rtr Base ISIS Instance 0 LFA Coverage
=====
Topology          Level  Node          IPv4          IPv6
-----
IPV4 Unicast      L1     5/7(71%)     9/13(69%)     9/13(69%)
IPV6 Unicast      L1     0/0(0%)      0/0(0%)      0/0(0%)
IPV4 Multicast    L1     0/0(0%)      0/0(0%)      0/0(0%)
```

IPv6 Multicast	L1	0/0(0%)	0/0(0%)	0/0(0%)
IPv4 Unicast	L2	5/7(71%)	9/13(69%)	9/13(69%)
IPv6 Unicast	L2	0/0(0%)	0/0(0%)	0/0(0%)
IPv4 Multicast	L2	0/0(0%)	0/0(0%)	0/0(0%)
IPv6 Multicast	L2	0/0(0%)	0/0(0%)	0/0(0%)

The preceding information can be derived from the table of alternative ISIS routes as follows. For PE-4, there are 17 routes: 8 routes to nodes and 9 routes to networks. The node and the networks that have 0.0.0.0 as next-hop must not be considered. This leaves $(8 - 1) = 7$ routes to nodes and $(9 - 3) = 6$ routes to networks. 5 out of 7 node destinations, and 4 out of 6 network destinations have an LFA next-hop (L). This leads to $(5 + 4) / (7 + 6) = 9/13$ IPv4 prefixes that have ISIS IP routing LFA coverage. A similar derivation applies for IPv6 prefixes.

```
[/]
A:admin@PE-4# show router isis routes alternative

=====
Rtr Base ISIS Instance 0 Route Table (alternative)
=====
Prefix[Flags]           Metric    Lvl/Typ    Ver.  SysID/Hostname
NextHop                MT        AdminTag/SID[F]
Alt-Nexthop            Alt-      Alt-Type
                        Metric
-----
192.0.2.1/32            10        1/Int.     57    PE-1
  192.168.14.1          0          0/1[NnP]
192.0.2.2/32            20        1/Int.     62    PE-1
  192.168.14.1          0          0/2[NnP]
192.0.2.3/32            20        1/Int.     82    PE-5
  192.168.45.2          0          0/3[NnP]
  192.168.46.2(L)      70        LP
192.0.2.4/32            0         1/Int.     3     PE-4
  0.0.0.0               0          0/4[NnP]
192.0.2.5/32            10        1/Int.     82    PE-5
  192.168.45.2          0          0/5[NnP]
  192.168.46.2(L)      60        LP
192.0.2.6/32            30        1/Int.     70    PE-6
  192.168.46.2          0          0/6[NnP]
  192.168.45.2(L)      40        LP
192.0.2.7/32            30        1/Int.     82    PE-5
  192.168.45.2          0          0/7[NnP]
  192.168.46.2(L)      40        NP
192.0.2.8/32            20        1/Int.     82    PE-5
  192.168.45.2          0          0/8[NnP]
  192.168.46.2(L)      50        NP
192.168.12.0/30         20        1/Int.     57    PE-1
  192.168.14.1          0          0
192.168.14.0/30         10        1/Int.     6     PE-4
  0.0.0.0               0          0
192.168.23.0/30         40        1/Int.     82    PE-1
  192.168.14.1          0          0
192.168.35.0/30         20        1/Int.     82    PE-5
  192.168.45.2          0          0
  192.168.46.2(L)      70        LP
192.168.45.0/30         10        1/Int.     82    PE-4
  0.0.0.0               0          0
192.168.46.0/30         30        1/Int.     66    PE-4
  0.0.0.0               0          0
192.168.58.0/30         20        1/Int.     82    PE-5
  192.168.45.2          0          0
  192.168.46.2(L)      70        LP
```

```
192.168.67.0/30          40      1/Int.    82    PE-5
  192.168.45.2          0        0
  192.168.46.2(L)      50      NP
192.168.78.0/30        30      1/Int.    82    PE-5
  192.168.45.2          0        0
  192.168.46.2(L)      60      NP
-----
No. of Routes: 17 (17 paths)
-----
Flags      : L = Loop-Free Alternate nexthop
Alt-Type   : LP = linkProtection, NP = nodeProtection
SID[F]     : R = Re-advertisement
            N = Node-SID
            nP = no penultimate hop POP
            E = Explicit-Null
            V = Prefix-SID carries a value
            L = value/index has local significance
=====
```

Conclusion

TI-LFA extends the calculation of a backup path for cases where the extended P space and the Q space do not overlap. TI-LFA also ensures that the post-failure path coincides with the post-convergence path, which avoids a switchover after SPF convergence.

Tunneling of ICMP Reply Packets over MPLS LSPs

This chapter provides information about tunneling of ICMP reply packets over MPLS LSPs.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter is applicable to SR OS routers and was initially written for SR OS Release 13.0.R7. The CLI in the current edition corresponds to SR OS Release 23.3.R1. Internet Control Message Protocol (ICMP) tunneling over Multi-Protocol Label Switching (MPLS) Label Switched Paths (LSPs) is supported in SR OS Release 12.0.R4 or later.

Overview

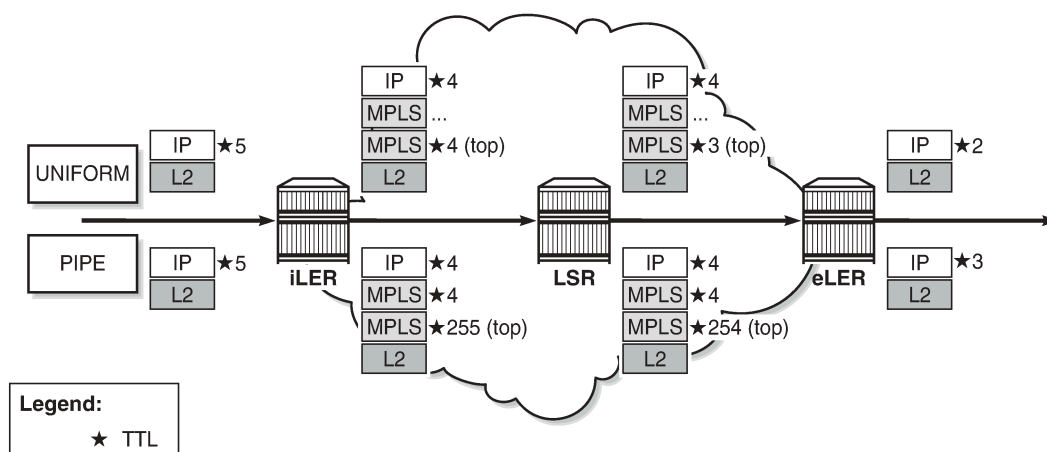
In IP forwarding, Time-To-Live (TTL) is a well-known mechanism to mitigate the damage in case of a loop. The TTL value in the IP header is decremented by one at each hop and the packet is discarded when the TTL equals 0. TTL is also used in traceroute, where the first batch of echo requests are sent with TTL equal to 1, the second batch of echo requests is sent with TTL equal to 2, and so on. Any intermediate node where the TTL expires (is decremented to 0) sends an ICMP reply of type "Time exceeded" (type 11) to the sender. From the replies, the sequence of hops can be determined.

If ICMP messages are sent in an MPLS tunnel, in pipe mode, the hops in the tunnels are invisible and the TTL is only decremented by the Label Edge Routers (LERs), not by the intermediate Label Switching Routers (LSRs). However, there are two modes for TTL handling, according to RFC 3443, *Time To Live Processing in MPLS Networks*:

1. Uniform mode: the MPLS network is visible from the outside. MPLS nodes use the TTL in the same way as any other IP node.
2. Pipe mode: the MPLS network is invisible from the outside. MPLS use of TTL is independent from IP TTL use. The network appears like a pipe between ingress Label Edge Router (iLER) and egress Label Edge Router (eLER).

Both TTL uses are shown in [Figure 300: Use of TTL: uniform versus pipe](#):

Figure 300: Use of TTL: uniform versus pipe



25696

Independent of the mode, the iLER decrements the TTL in the IP header by one. The iLER adds service and transport MPLS headers.



Note:

In an L2 Virtual Private Network (VPN), the TTL in the IP header is kept intact.

- In uniform mode, the iLER sets the TTL of every MPLS header to match the TTL in the IP header and every LSR decrements the MPLS TTLs. The IP header remains unchanged as long as the packet is in the MPLS tunnel. The eLER pops the MPLS labels and decrements the minimum TTL of the headers (which is the TTL in both MPLS headers) by one. This TTL is used in the IP header.
- In pipe mode, the iLER sets the TTL of the top MPLS header to 255 and every LSR decrements that TTL by one. The eLER pops the MPLS labels and decrements the minimum TTL of the headers (which is the IP TTL) by one. This TTL is used in the IP header. There can be uncounted hops in pipe mode, because the LSRs are not counted.

The LERs can be in uniform mode and the LSRs in pipe mode, and the other way around.

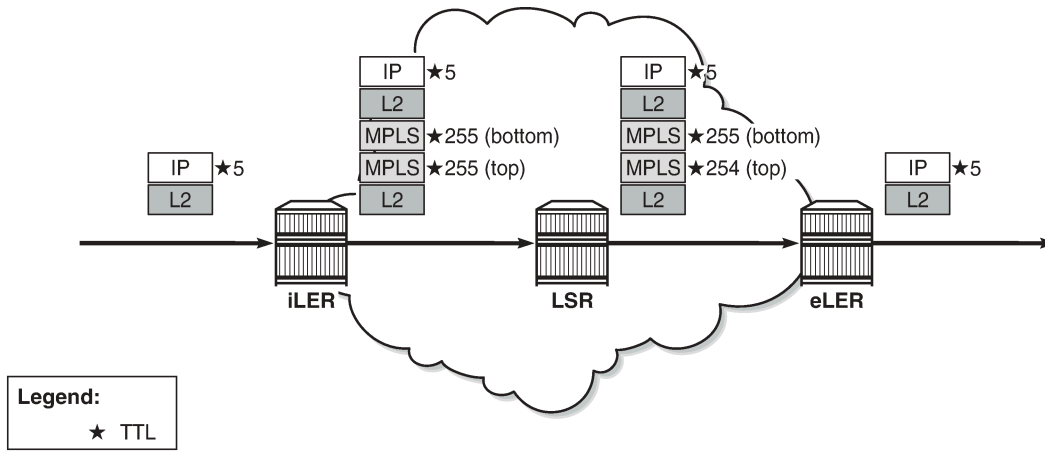
The default use of TTL in SR OS is as follows:

- Uniform mode for LSP shortcuts, ReSource reservation Protocol (RSVP) shortcuts, Label Distribution Protocol (LDP) shortcuts, and Border Gateway Protocol (BGP) shortcuts.
- Pipe mode for L2 and L3 VPN services, BGP labeled routes, IPv6 Provider Edge (6PE) router, and IPv6 on VPN to PE router (6VPE).

However, the use of TTL can be changed by configuration.

Figure 301: Use of TTL in an L2 VPN service in pipe mode shows the use of TTL for an L2 VPN service in pipe mode. The TTL in the IP header is preserved. There is no processing of the IP header for an L2 service. The TTL in the pushed MPLS headers is 255 and the TTL in the top MPLS header is decremented by one in the LSRs. The eLER pops the MPLS labels.

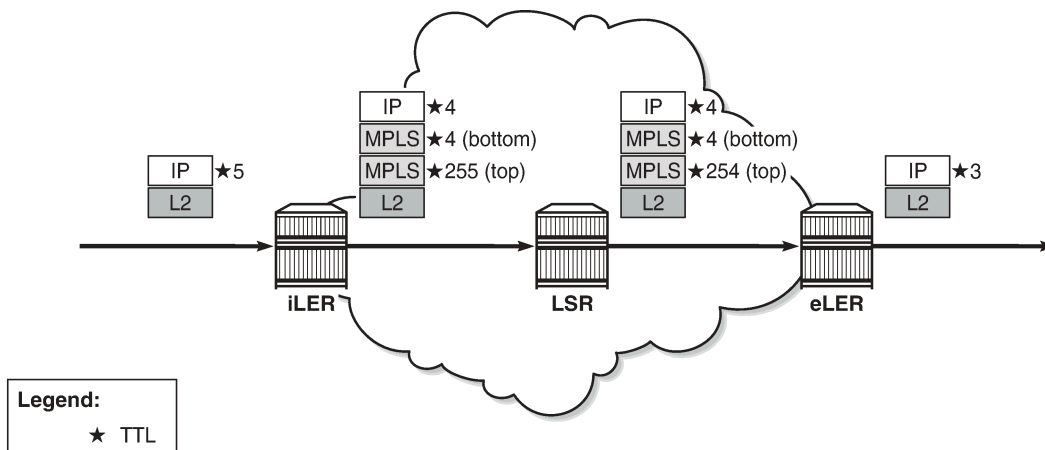
Figure 301: Use of TTL in an L2 VPN service in pipe mode



25697

Figure 302: Use of TTL in an L3 VPN service in pipe mode shows the use of TTL for an L3 VPN service in pipe mode. The TTL in the IP header is decremented by the iLER and the eLER, but not by the LSRs. In pipe mode, the bottom MPLS header inherits the IP TTL after it has been decremented by the iLER. The transport MPLS header gets TTL 255 and this TTL is decremented by one at each LSR. The eLER takes the minimum of the TTL of the MPLS headers and the IP TTL and decrements that by one. This will match the IP TTL in the forwarded packet. The MPLS labels are popped. There are uncounted hops, because the LSRs are invisible in pipe mode.

Figure 302: Use of TTL in an L3 VPN service in pipe mode



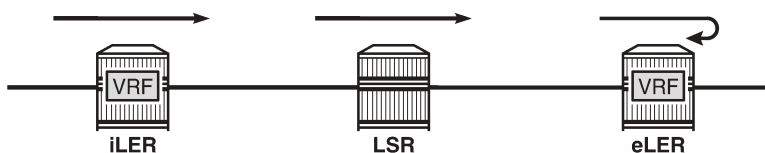
25698

Tunneling of ICMP reply packets over MPLS LSPs provides the ability for a network operator or customer to trace the MPLS network hops in the path, for Virtual Private Routed Network (VPRN), 6PE/6VPE, and BGP labeled routes.

ICMP tunneling over an MPLS LSP

Figure 303: Tunneling of ICMP reply packets over an MPLS LSP shows the actions performed in iLER, LSR, and eLER, when tunneling ICMP messages over an MPLS LSP:

Figure 303: Tunneling of ICMP reply packets over an MPLS LSP



- In the iLER, uniform mode is required within the VPRN service. The IP TTL is propagated in the MPLS TTL, both for in-transit and Control Processing Module (CPM) generated IP packets. In this example, it is assumed that a UDP traceroute message is forwarded with source IP address S1.
- In all LSRs, ICMP tunneling is enabled globally on the system, according to RFC 3032, *MPLS Label Stack Encoding*. When the MPLS TTL expires in an LSR, the LSR generates an ICMP reply with code "Time exceeded" and destination IP address S1. However, the CPM sends this ICMP reply packet in the forward direction of the MPLS LSP tunnel that the packet arrived on. The ICMP reply packet is sent to the eLER, not to the iLER.
- The eLER performs a lookup for the IP address S1 and sends the ICMP reply to S1 toward the iLER.
 - The lookup of the IP address S1 is in the Global Routing Table (GRT) for BGP shortcut, 6PE, and BGP labeled route prefixes.
 - The lookup of IP address S1 is in the Virtual Routing and Forwarding (VRF) table for VPRN and 6VPE prefixes.

TTL propagation

The TTL propagation can be configured in LERs and LSRs.

TTL propagation at the iLER

Different commands are used for TTL propagation in a VPRN versus BGP labeled routes. Pipe mode is enabled by default in either case.

TTL propagation in iLER for VPRN

The TTL propagation of VPN-IPv4 or VPN-IPv6 packets in a VPRN service can be enabled globally as follows:

```
A:admin@PE-3# configure exclusive
  router "Base" {
    ttl-propagate {
      vprn-local ?
    }
  }
[ex:/configure router "Base" ttl-propagate]
A:admin@PE-3#          vprn-local ?
```

```
vprn-local <keyword>
<keyword> - (none|all|vc-only)
Default   - vc-only

    TTL propagation from IP header into label stack for VPRN local packets

A:admin@PE-3# configure exclusive
  router "Base" {
    ttl-propagate {
      vprn-transit ?
[ex:/configure router "Base" ttl-propagate]
A:admin@PE-3#           vprn-transit ?

vprn-transit <keyword>
<keyword> - (none|all|vc-only)
Default   - vc-only

    TTL propagation from IP header into label stack for VPRN transit packets
```

There are three options for the propagation of TTL in the iLER of a VPRN:

- | | |
|-------------------|---|
| none | No IP TTL propagation to any MPLS header in the stack: the transport and the VC MPLS headers will have a value of 255. This is needed for correct operation of traceroute in an inter-AS option B VPRN. |
| all | Uniform mode: IP TTL is propagated to all MPLS headers in the stack. |
| vc-only (default) | Pipe mode: IP TTL is propagated to the VC header, but not to the transport headers in the stack. |

For more information about inter-AS option B, see chapter *Rosen MVPN Inter-AS Option B*.

In inter-AS option B, a traceroute for a VPN IP prefix issued from a Customer Edge (CE) router results in both ingress Autonomous System Boundary Router (ASBR) and egress ASBR not responding. The traceroute also misses a couple of hops if the target CE node is two or more hops away from the egress PE. The reason is that the VC label TTL inherits the decremented IP TTL at the ingress PE, but is decremented twice in the MPLS network whereas the IP TTL is only decremented at the ingress and the egress PE nodes. The option "none" for **ttl-propagate** makes the ASBRs transparent to the traceroute behavior and corrects the uncounted hop issue.

The global configuration can be overruled within each VPRN, as follows

```
A:admin@PE-3# configure exclusive
  service {
    vprn "1" {
      ttl-propagate {
        local ?
[ex:/configure service vprn "1" ttl-propagate]
A:admin@PE-3#           local ?

local <keyword>
<keyword> - (none|all|vc-only|use-base)
Default   - use-base

    Local TTL propagation control for the VPRN

A:admin@PE-3# configure exclusive
  service {
```

```
    vprn "1" {
        ttl-propagate {
            transit ?
[ex:/configure service vprn "1" ttl-propagate]
A:admin@PE-3#                transit ?

transit <keyword>
<keyword> - (none|all|vc-only|use-base)
Default   - use-base

    Transit TTL propagation control for the VPRN
```

TTL propagation in iLER for BGP labeled route

IPv4 and IPv6 packets are forwarded using BGP labeled routes in the GRT, as described in RFC 3107, *Carrying Label Information in BGP-4*. This also applies to 6PE. TTL propagation for RFC 3107 label routes can be configured as follows:

```
A:admin@PE-3# configure exclusive
  router "Base" {
    ttl-propagate {
      label-route-local ?
[ex:/configure router "Base" ttl-propagate]
A:admin@PE-3#                label-route-local ?

label-route-local <keyword>
<keyword> - (none|all)
Default   - none

    TTL propagation from IP header into label stack for local packets

A:admin@PE-3# configure exclusive
  router "Base" {
    ttl-propagate {
      label-route-transit ?
[ex:/configure router "Base" ttl-propagate]
A:admin@PE-3#                label-route-transit ?

label-route-transit <keyword>
<keyword> - (none|all)
Default   - none

    TTL propagation from IP header into label stack for transit packets
```

There are two options for TTL propagation in the iLER for BGP labeled routes:

- none (default) Pipe mode: No TTL is propagated from the IP header to the MPLS headers in the transport MPLS stack. However, the IP TTL is propagated to the bottom header: the virtual circuit (VC) header
- all Uniform mode: TTL is propagated to all headers in the transport MPLS stack.

If the BGP peer advertises the implicit-null label value for the BGP labeled route (in the case of a third-party implementation), the TTL propagation follows the configuration of the RSVP/LDP LSP shortcut that the BGP labeled route resolves to. This is not controlled by the preceding commands.

TTL propagation at the LSR

In a VPRN service, there is no TTL propagation to be configured in the LSRs.

TTL propagation in LSR for BGP labeled route

The IP TTL and VC TTL are not decremented by the LSRs. The TTL that is decremented is the minimum of the RSVP/LDP transport TTL and the BGP TTL.

1. The LSR determines the TTL using the following function:

TTL = MIN {incoming transport label stack TTL, incoming swapped/stitched label TTL}

2. The LSR decrements the TTL by one and writes it to the outgoing swapped/stitched BGP label.

This is always performed when an LSR is swapping or stitching a label at any stack depth.

The control plane indicates to the data plane whether a BGP labeled route is stitched or an LDP FEC is being stitched. The same node can perform stitching for one BGP labeled route and swapping for another one. See chapter [LDP FEC to BGP Label Route Stitching](#) for more information.

3. The LSR can propagate the decremented TTL to the outgoing transport label stack (if any) that is pushed on top of the BGP swapped/stitched label. This is configured as follows:

```
A:admin@PE-2# configure exclusive
  router "Base" {
    ttl-propagate lsr-label-route ?
  }
*[ex:/configure router "Base"]
A:admin@PE-2#          ttl-propagate lsr-label-route ?

  lsr-label-route <keyword>
  <keyword> - (none|all)
  Default   - none
```

TTL propagation from IP header into label stack

There are two options for TTL propagation in the LSR:

- | | |
|-------------------|--|
| none
(default) | No TTL propagation of the decremented TTL to the MPLS transport label stack. |
| all | TTL propagation of the decremented TTL to all LDP/RSVP transport labels. |

It is safe to not propagate the TTL to the transport label stack for an ASBR/Area Border Router (ABR)/data path Route Reflector (RR)/BGP-LDP stitching node. Not propagating the TTL provides isolation of the network domains downstream of the LSRs. Operations, Administration, and Maintenance (OAM) packets, such as traceroute and ping, sent in the context of a BGP labeled route or VPRN will not expire in LSR nodes within these domains.

A node performing pseudowire (PW) switching terminates the transport label stack in pipe mode; the node ignores the TTL of the incoming transport label stack and propagates the TTL of the VC label. The TTL of the new pushed transport label stack is always 255.

Some considerations on TTL propagation in LSR for BGP labeled routes

- When an LSR stitches an LDP label to a BGP label, the decremented TTL of the stitched label can be propagated to the LDP/RSVP transport labels with the preceding configuration.
- When an LSR stitches a BGP label to an LDP label, the decremented TTL of the stitched label is automatically propagated to the RSVP label if the outgoing LDP LSP is tunneled over RSVP.
- When the LSR pops a BGP label and forwards the packet using an IGP route (IGP route is preferred over BGP labeled route), the LSR pushes an LDP label on the packet and the TTL behavior is the same as when an LSR stitches a BGP label to an LDP label.
- In a Carrier Supporting Carrier (CSC) VPRN, the ingress CSC CE swaps an iBGP label for an eBGP label and the ingress CSC PE swaps the incoming eBGP label for a VPN-IPv4 label. The reverse operation is performed by the egress CSC PE and the egress CSC CE. In all cases, the decremented TTL of the swapped label is propagated to the LDP/RSVP transport labels.
- SR OS does not support ASBR or data path RR functionality for labeled IPv6 routes in the global routing instance (6PE).

TTL propagation at the eLER

For packets received with a BGP labeled route and searched for in the GRT, the TTL of the forwarded IP packet is set to $\text{MIN}\{\text{MPLS_TTL}-1, \text{IP_TTL}-1\}$, where `MPLS_TTL` refers to the TTL in the outermost label in the popped stack. This is the same behavior as for LSP shortcuts.

For packets received in the context of VPRN, the TTL of the forwarded IP packet is set to $\text{MIN}\{\text{MPLS_TTL}-1, \text{VC_TTL}-1, \text{IP_TTL}-1\}$, where `MPLS_TTL` refers to the TTL in the outermost label in the popped stack and `VC_TTL` refers to the TTL in the VC label in the popped stack.

Some considerations on TTL propagation at the eLER

- When a packet is received in one VPRN instance and is redirected using policy-based routing to be forwarded in another VPRN instance, the TTL is governed by the configuration of the outgoing VPRN instance.
- When a packet is received in a `vprn` context but is searched for in the GRT (GRT leaking configured), the behavior of the TTL propagation is governed by:
 - the BGP labeled route configuration when the matching route is an RFC 3107 label route or a 6PE route
 - the LSP shortcut configuration when the matching route is an RSVP or LDP shortcut (default uniform mode)

For shortcuts, uniform mode is default. Pipe mode can be configured as follows:

```
# on eLER PE-6:
configure exclusive
  router "Base" {
    ldp {
      shortcut-local-ttl-propagate false
      shortcut-transit-ttl-propagate false
    }
    mpls {
      shortcut-local-ttl-propagate false
      shortcut-transit-ttl-propagate false
    }
  }
```

```
}
 rsvp { }
```

Enabling ICMP tunneling on LSRs

For all scenarios (VPRN and BGP labeled routes), ICMP tunneling needs to be enabled on all LSRs, as follows:

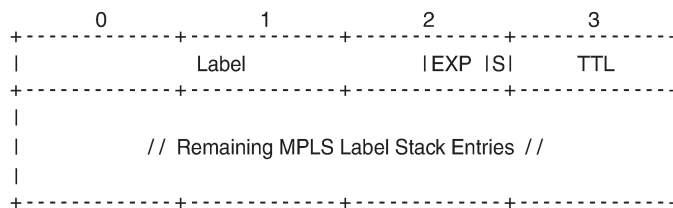
```
# on all LSRs:
configure exclusive
router "Base" {
    icmp-tunneling true
```

The LSR will generate the ICMP reply packet of type 11 - "Time exceeded", with source IP address set to a local address of the LSR node and appending the IP header and leading octets of the original datagram. The LSR does not perform a lookup for the destination IP address of the ICMP reply packet, which is the source IP address of the sender of the label TTL expiry packet. The CPM injects the ICMP reply packet in the forward direction toward the eLER. The TTL of pushed labels is 255.

There is no need to enable ICMP tunneling on the eLER. The eLER performs a user packet lookup in the data path in the VRF table or GRT and forwards the ICMP reply packet to the destination. If the eLER does not have a route to the destination, the packet is dropped.

RFC 4950, *ICMP Extensions for Multiprotocol Label Switching*, defines an extension object (MPLS label stack object) that allows LSRs to include label stack information to ICMP messages; see [Figure 304: MPLS label stack object](#):

Figure 304: MPLS label stack object

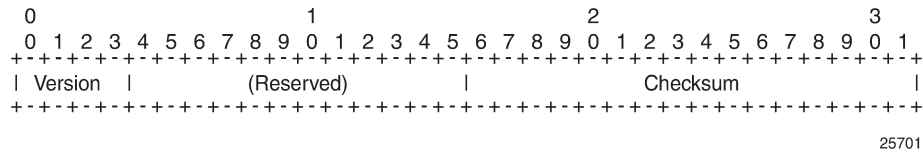


25700

The MPLS label stack object is applicable for ICMPv4 and ICMPv6. The MPLS label stack contains the MPLS shim header: label, experimental bits for Type of Service (ToS), S-bit indicating the bottom of the stack, and TTL. The object can be appended to the ICMP Time Exceeded and ICMP Destination Unreachable messages. The LSR that sends the ICMP reply message will not change the MPLS label stack.

RFC 4884, *Extended ICMP to Support Multi-Part Messages*, defines the ICMP extension header; see [Figure 305: ICMP extension header](#):

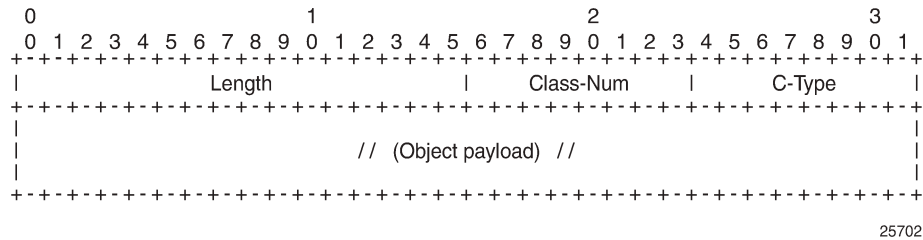
Figure 305: ICMP extension header



The version of the ICMP extension header is 2. The twelve reserved bits must be set to 0.

An extension object contains 32-bit words, representing an object header and payload, as defined in RFC 4884; see [Figure 306: ICMP extension object: object header and payload](#):

Figure 306: ICMP extension object: object header and payload



The length of the object is the length of the header (4 octets) plus the length of the object payload: 4 octets per LSR. The class number identifies the object class; in this case, object class 1 for MPLS label stack class. The C-type defines the object subtype; in this case, the subtype is 1 for an incoming MPLS label stack.

Backward compatibility is guaranteed between the ICMP message with extension header, the ICMP messages without extension header, and the ICMP message with a non-compliant extension header.

Effect of ICMP tunneling on OAM

ICMP tunneling over MPLS LSPs affects the behavior of some CPM originated OAM packets that are forwarded within a **vprn** context.

- ICMP ping and UDP traceroute are sent according to the TTL propagation configured in the **vprn** context.
- VPRN ping and VPRN traceroute are not affected.

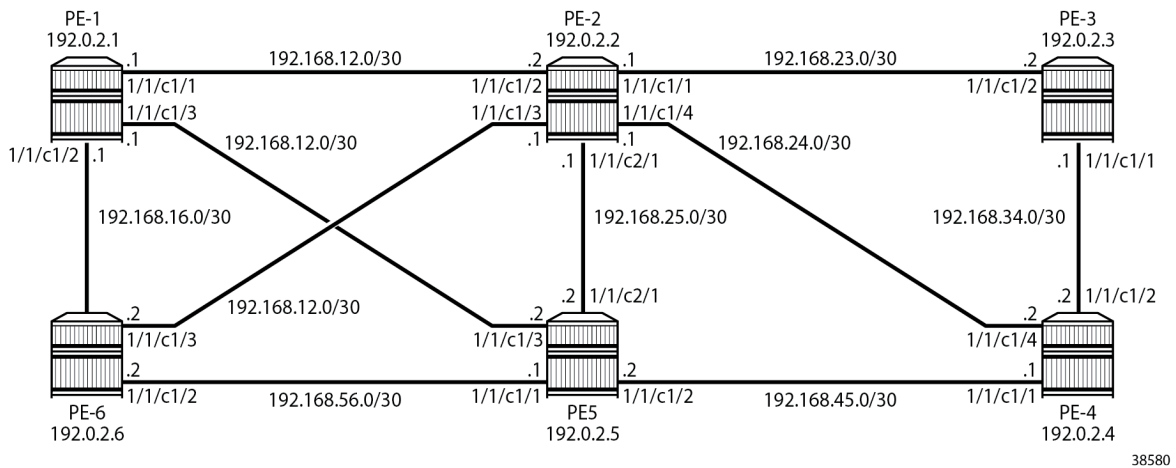
OAM packets forwarded over a BGP labeled route follow the TTL configuration of the iLER.

ICMP tunneling behavior at an LSR only applies to UDP traceroute packets. Other OAM packets expiring at the LSR, such as ICMP ping, VPRN ping, VPRN trace, LSP ping, and LSP trace, follow their specific procedures or are silently dropped.

Configuration

[Figure 307: Example configuration](#) shows the example configuration, which has six 7750 SRs:

Figure 307: Example configuration



38580

Initial configuration

The nodes have the following initial configuration:

- Cards, MDAs, ports
- Router interfaces.

```
# on PE-3:
configure exclusive
  router "Base" {
    interface "int-PE-3-PE-2" {
      port 1/1/c1/2
      ipv4 {
        primary {
          address 192.168.23.2
          prefix-length 30
        }
      }
    }
    interface "int-PE-3-PE-4" {
      port 1/1/c1/1
      ipv4 {
        primary {
          address 192.168.34.1
          prefix-length 30
        }
      }
    }
    interface "system" {
      ipv4 {
        primary {
          address 192.0.2.3
          prefix-length 32
        }
      }
    }
  }
}
```

- IGP: OSPF (alternatively, any IGP could have been used).

```
# on PE-3:
configure exclusive
router "Base" {
  ospf 0 {
    admin-state enable
    area 0.0.0.0 {
      interface "int-PE-3-PE-2" {
        interface-type point-to-point
      }
      interface "int-PE-3-PE-4" {
        interface-type point-to-point
      }
      interface "system" {
      }
    }
  }
}
```

- Link LDP.

```
# on PE-3:
configure exclusive
router "Base" {
  ldp {
    interface-parameters {
      interface "int-PE-3-PE-2" {
        ipv4 {
        }
      }
      interface "int-PE-3-PE-4" {
        ipv4 {
        }
      }
    }
  }
}
```

Configure VPRN

A VPRN service is configured on PE-3 and PE-6. The routes are exchanged via BGP. BGP is configured on all nodes with PE-2 as route reflector (RR).

```
# on PE-1, PE-3, PE-4, PE-5, and PE-6:
configure exclusive
router "Base" {
  autonomous-system 64496
  bgp {
    ebgp-default-reject-policy {
      import false
      export false
    }
    group "internal" {
      peer-as 64496
      family {
        vpn-ipv4 true
      }
    }
  }
  neighbor "192.0.2.2" {
    group "internal"
  }
}
```

The configuration on RR PE-2 is as follows:

```
# on PE-2:
configure exclusive
router "Base" {
  autonomous-system 64496
  bgp {
    cluster {
      cluster-id 1.1.1.1
    }
    ebgp-default-reject-policy {
      import false
      export false
    }
    group "internal" {
      peer-as 64496
      family {
        vpn-ipv4 true
      }
    }
    neighbor "192.0.2.1" {
      group "internal"
    }
    neighbor "192.0.2.3" {
      group "internal"
    }
    neighbor "192.0.2.4" {
      group "internal"
    }
    neighbor "192.0.2.5" {
      group "internal"
    }
    neighbor "192.0.2.6" {
      group "internal"
    }
  }
}
```

Import and export policies are configured on PE-3 and PE-6, as follows:

```
# on PE-3 and PE-6:
configure exclusive
policy-options {
  community "VPN1" {
    member "target:64496:1" { }
  }
}
policy-statement "VPN1-export" {
  entry 10 {
    from {
      protocol {
        name [direct]
      }
    }
    to {
      protocol {
        name [bgp-vpn]
      }
    }
    action {
      action-type accept
      community {
        add ["VPN1"]
      }
    }
  }
}
```

```
    }  
  }  
  policy-statement "VPN1-import" {  
    entry 10 {  
      from {  
        community {  
          name "VPN1"  
        }  
        protocol {  
          name [bgp-vpn]  
        }  
      }  
      action {  
        action-type accept  
      }  
    }  
  }  
}
```

VPN 1 is configured on PE-3 and PE-6, as follows:

```
# on PE-3:  
configure exclusive  
  service {  
    vprn "1" {  
      admin-state enable  
      customer "1"  
      service-id 1  
      bgp-ipvpn {  
        mpls {  
          admin-state enable  
          route-distinguisher "64496:13"  
          vrf-import {  
            policy ["VPN1-import"]  
          }  
          vrf-export {  
            policy ["VPN1-export"]  
          }  
          auto-bind-tunnel {  
            resolution filter  
            resolution-filter {  
              ldp true  
            }  
          }  
        }  
      }  
    }  
  }  
  interface "loopback1" {  
    loopback true  
    ipv4 {  
      primary {  
        address 192.0.1.3  
        prefix-length 32  
      }  
    }  
  }  
}
```

The configuration on PE-6 is similar, with route-distinguisher 64496:16 and loopback address 192.0.1.6/32.

Default TTL handling in VPRN

The default configuration for TTL propagation on the iLER corresponds to pipe mode, as follows:

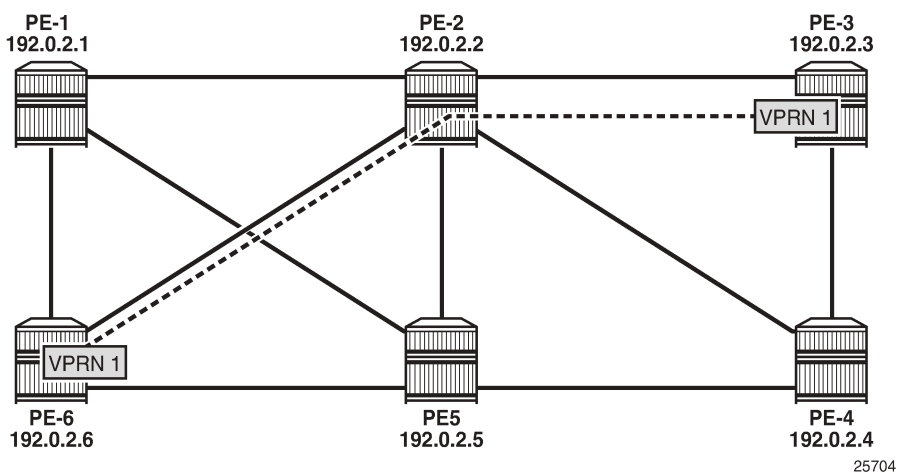
```
# on PE-3:
configure exclusive
  router "Base" {
    ttl-propagate {
      info detail
    }
  }
[ex:/configure router "Base" ttl-propagate]
A:admin@PE-3# info detail
## apply-groups
## apply-groups-exclude
label-route-local none
label-route-transit none
lsr-label-route none
vprn-local vc-only
vprn-transit vc-only
```

By default, no ICMP tunneling is enabled in the LSRs, which implies that no ICMP "Time exceeded" messages will be tunneled by the LSR to the eLER. A traceroute message sent in VPRN 1 from PE-3 to the loopback address in VPRN 1 on PE-6 shows that the loopback address is the next hop. There are no intermediate hops detected.

```
[/]
A:admin@PE-3# traceroute 192.0.1.6 router-instance "1"
traceroute to 192.0.1.6, 30 hops max, 40 byte packets
 1 192.0.1.6 (192.0.1.6) 1.23 ms 1.13 ms 1.16 ms
```

Figure 308: Tunnel from iLER PE-3 to eLER PE-6 via LSR PE-2 shows the tunnel from iLER PE-3 to eLER PE-6:

Figure 308: Tunnel from iLER PE-3 to eLER PE-6 via LSR PE-2



For comparison, a traceroute in the base router toward the system address of PE-6 shows PE-2 as intermediate hop, as follows:

```
[/]
A:admin@PE-3# traceroute 192.0.2.6
traceroute to 192.0.2.6, 30 hops max, 40 byte packets
```

```
1 192.168.23.1 (192.168.23.1) 0.856 ms 0.987 ms 0.854 ms
2 192.0.2.6 (192.0.2.6) 1.16 ms 1.54 ms 1.10 ms
```

Uniform mode in iLER and ICMP tunneling in LSR

In the iLER PE-3, uniform mode is enabled for local VPRNs, as follows:

```
# on iLER PE-3:
configure exclusive
service {
    vprn "1" {
        ttl-propagate {
            local all
        }
    }
}
```

This is a specific configuration for VPRN 1 that overrules the global configuration. By default, it is set to inherit the global configuration. By default, the global configuration is pipe mode.

This TTL propagation is only configured on PE-3, not on PE-6. This implies that traceroute messages from the VPRN in PE-3 will have TTL propagation to all MPLS labels (uniform mode), while traceroute messages from the VPRN in PE-6 will have pipe mode.

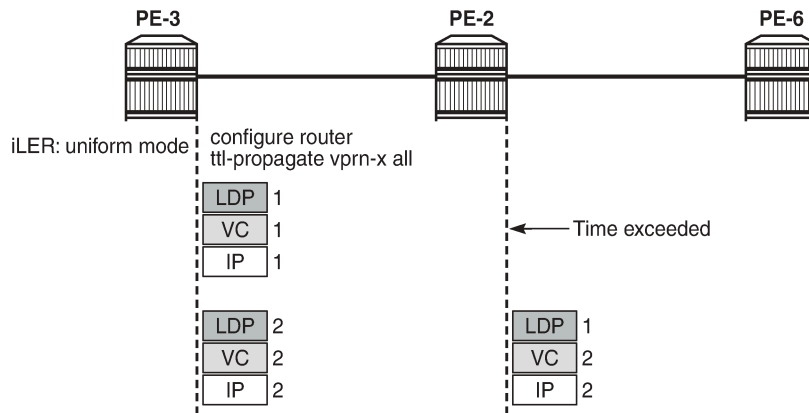
In the LSR PE-2, ICMP tunneling is enabled, as follows:

```
# on LSR PE-2:
configure exclusive
router "Base" {
    icmp-tunneling true
}
```

A UDP traceroute is sent from VPRN 1 on PE-3 to the loopback address in VPRN 1 on PE-6. A message with TTL 1 is sent first. The IP TTL and VC TTL are not decremented by the LSR PE-2. Only the LDP TTL is decremented, so this message times out on the LSR PE-2, which tunnels the ICMP Time Exceeded reply with destination address 192.0.1.3 toward eLER PE-6. The eLER looks up the prefix 192.0.1.3 in the VRF table and sends the ICMP Time Exceeded reply toward VPRN 1 in PE-3. Three UDP traceroute messages with TTL 1 are sent. Then, UDP traceroute messages with TTL 2 are sent. They reach the destination PE-6 before time-out.

[Figure 309: UDP traceroute in VPRN with iLER in uniform mode](#) shows the TTLs in the UDP traceroute messages:

Figure 309: UDP traceroute in VPRN with iLER in uniform mode



25705

In the following output, there is only an MPLS label stack object when TTL expires in the LSR, because ICMP tunneling is occurring. The MPLS label stack object is not used by the eLER. The LSR where ICMP tunneling occurs adds an MPLS label stack object to the ICMP reply message. The MPLS label stack object contains information about the MPLS labels (VC label and LDP transport label) in the stack: MPLS labels, experimental bits for ToS, and TTL. S indicates the bottom of the label stack. In the detailed output of the **traceroute** command, the MPLS label stack information is shown for the echo requests that timed out in the LSR:

```
[/]
A:admin@PE-3# traceroute 192.0.1.6 router-instance "1" detail
traceroute to 192.0.1.6, 30 hops max, 40 byte packets
 1  1  192.168.26.1 (192.168.26.1)  2.03 ms
      returned MPLS Label Stack Object
          entry 1: MPLS Label = 524282, Exp = 7, TTL =  1, S = 0
          entry 2: MPLS Label = 524280, Exp = 7, TTL =  1, S = 1
 1  2  192.168.26.1 (192.168.26.1)  1.70 ms
      returned MPLS Label Stack Object
          entry 1: MPLS Label = 524282, Exp = 7, TTL =  1, S = 0
          entry 2: MPLS Label = 524280, Exp = 7, TTL =  1, S = 1
 1  3  192.168.26.1 (192.168.26.1)  1.58 ms
      returned MPLS Label Stack Object
          entry 1: MPLS Label = 524282, Exp = 7, TTL =  1, S = 0
          entry 2: MPLS Label = 524280, Exp = 7, TTL =  1, S = 1
 2  1  192.0.1.6 (192.0.1.6)  1.70 ms
 2  2  192.0.1.6 (192.0.1.6)  1.72 ms
 2  3  192.0.1.6 (192.0.1.6)  1.76 ms
```

The top label or transport label 524282 is the LDP label pushed by PE-3:

```
[/]
A:admin@PE-3# show router ldp bindings active prefixes prefix 192.0.2.6/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.3)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
```

```
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
  (S) - Static          (M) - Multi-homed Secondary Support
  (B) - BGP Next Hop   (BU) - Alternate Next-hop for Fast Re-Route
  (I) - SR-ISIS Next Hop (O) - SR-OSPF Next Hop
  (C) - FEC resolved with class-based-forwarding

=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.6/32                         Push
--                                   524282
192.168.23.1                         1/1/c1/2

192.0.2.6/32                         Swap
524282                               524282
192.168.23.1                         1/1/c1/2

-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

The bottom label or service label 524280 is the BGP label, which remains the same end-to-end:

```
[/]
A:admin@PE-3# show router bgp routes vpn-ipv4
=====
BGP Router ID:192.0.2.3      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
BGP VPN-IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                    Path-Id    IGP Cost
      As-Path                               Path-Id    Label
-----
i     64496:13:192.0.1.3/32                100        None
      192.0.2.3                            None        0
      No As-Path                             None        524280
u*>i 64496:16:192.0.1.6/32                100        None
      192.0.2.6                            None        20
      No As-Path                             None        524280

-----
Routes : 2
=====
```

When the iLER is configured in pipe mode (vc-only) or if there is no TTL propagation to any MPLS label (none), the output of the **traceroute detail** command does not contain the MPLS label stack object information. In pipe mode, the IP TTL is propagated to the VC header, but not to the LDP header. When the TTL propagation is none, the IP TTL is not propagated to VC or LDP. The LSRs are invisible and there will be missing hops.

```
# on PE-3:
```



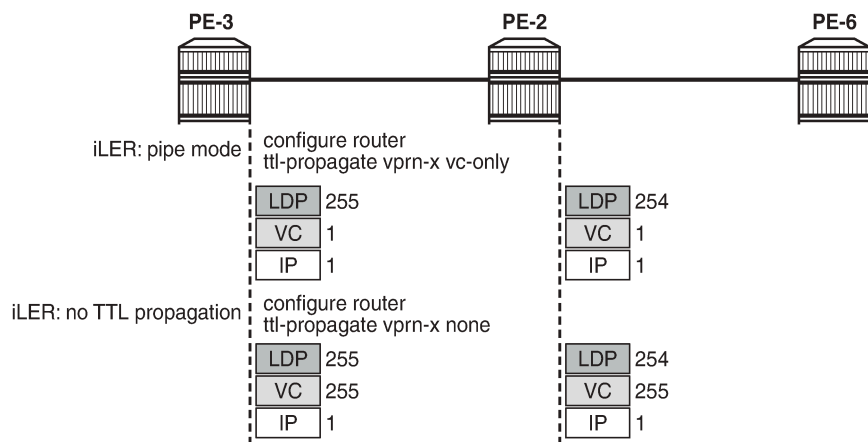
```

configure exclusive
  service {
    vprn "1" {
      ttl-propagate {
        local vc-only
      }
    }
  }
[/]
A:admin@PE-3# traceroute 192.0.1.6 router-instance "1" detail
traceroute to 192.0.1.6, 30 hops max, 40 byte packets
 1  1  192.0.1.6  (192.0.1.6)  2.02 ms
 1  2  192.0.1.6  (192.0.1.6)  2.05 ms
 1  3  192.0.1.6  (192.0.1.6)  2.03 ms

```

The reason is that the TTL of the LDP header is 255 at the iLER and it is decremented by one in every LSR. The UDP traceroute message will not time out on the LSR PE-2. The TTL of the VC header is not decremented in the LSR. When the traceroute message does not time out in PE-2, the hop PE-2 is invisible. In a similar way, the traceroute messages will not time out in the LSR when no TTLs are propagated to any MPLS header. [Figure 310: UDP traceroute in VPRN without TTL propagation to LDP](#) shows the TTLs in both cases:

Figure 310: UDP traceroute in VPRN without TTL propagation to LDP



25706

The TTL propagation is restored to uniform mode in the iLER as follows:

```

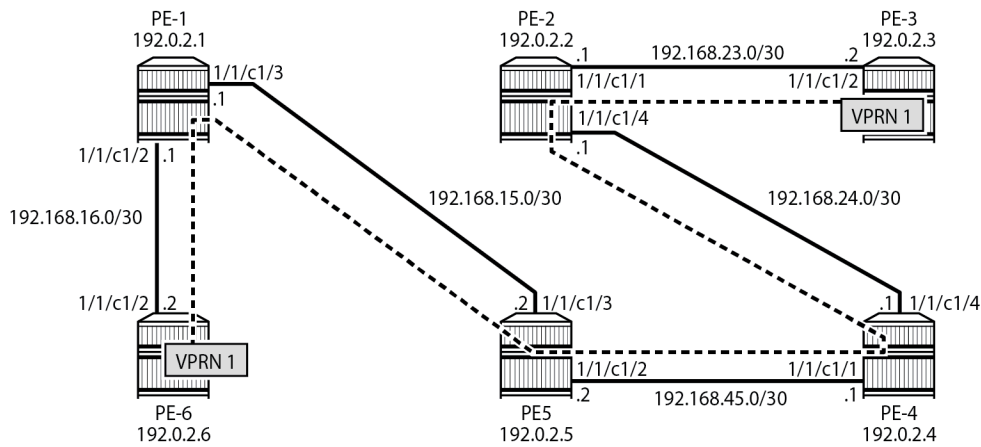
# on PE-3:
configure exclusive
  service {
    vprn "1" {
      ttl-propagate {
        local all
      }
    }
  }

```

Uniform mode in iLER and ICMP tunneling in multiple LSRs

After disabling some ports in the nodes, the tunnel from iLER PE-3 to PE-6 has four intermediate hops (LSRs) instead of one, as shown in [Figure 311: Tunnel from iLER PE-3 to eLER PE-6 with multiple LSRs](#):

Figure 311: Tunnel from iLER PE-3 to eLER PE-6 with multiple LSRs



38581

```
# on PE-2:
configure exclusive
port 1/1/c1/2 {
    admin-state disable
}
port 1/1/c1/3 {
    admin-state disable
}
port 1/1/c2/1 {
    admin-state disable
}

# on PE-3:
configure exclusive
port 1/1/c1/1 {
    admin-state disable

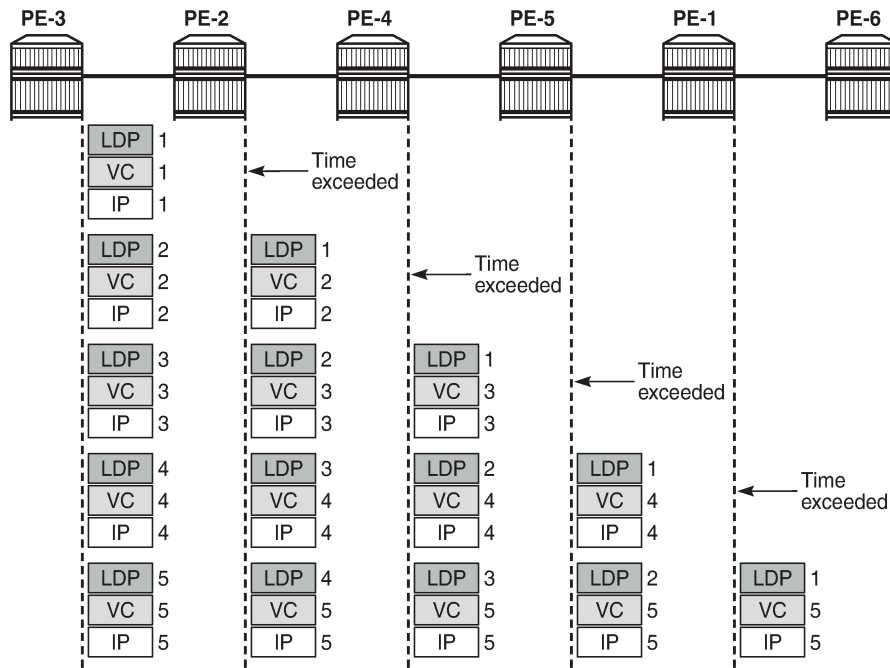
# on PE-5:
configure exclusive
port 1/1/c1/1 {
    admin-state disable
```

TTL propagation on the iLER PE-3 is in uniform mode. ICMP tunneling needs to be enabled on all LSRs, as follows:

```
# on all LSRs:
configure exclusive
router "Base" {
    icmp-tunneling true
```

UDP traceroute messages are sent from VPRN 1 on iLER PE-3 to VPRN 1 on PE-6, as shown in [Figure 312: UDP traceroute with iLER in uniform mode:](#)

Figure 312: UDP traceroute with iLER in uniform mode



25708

The detailed output of the **traceroute** command shows the MPLS label stack object information as added by the LSR where the ICMP Time Exceeded message was tunneled. For brevity, only the first of the three messages from each intermediate node is shown.

```
[/]
A:admin@PE-3# traceroute 192.0.1.6 router-instance "1" detail
traceroute to 192.0.1.6, 30 hops max, 40 byte packets
 1  1  192.168.24.1 (192.168.24.1) 2.04 ms
    returned MPLS Label Stack Object
      entry 1: MPLS Label = 524286, Exp = 7, TTL = 1, S = 0
      entry 2: MPLS Label = 524280, Exp = 7, TTL = 1, S = 1
---snip---
 2  1  192.168.45.1 (192.168.45.1) 2.29 ms
    returned MPLS Label Stack Object
      entry 1: MPLS Label = 524281, Exp = 7, TTL = 1, S = 0
      entry 2: MPLS Label = 524280, Exp = 7, TTL = 2, S = 1
---snip---
 3  1  192.168.15.2 (192.168.15.2) 2.02 ms
    returned MPLS Label Stack Object
      entry 1: MPLS Label = 524282, Exp = 7, TTL = 1, S = 0
      entry 2: MPLS Label = 524280, Exp = 7, TTL = 3, S = 1
---snip---
 4  1  192.168.16.1 (192.168.16.1) 1.96 ms
    returned MPLS Label Stack Object
      entry 1: MPLS Label = 524282, Exp = 7, TTL = 1, S = 0
      entry 2: MPLS Label = 524280, Exp = 7, TTL = 4, S = 1
---snip---
 5  1  192.0.1.6 (192.0.1.6) 1.96 ms
---snip---
```

The TTL for the bottom MPLS header (BGP) is not decremented in each hop; only the TTL for the transport MPLS header (LDP) is decremented. The bottom label or BGP label of 524280 is not changed end-to-end. The LDP transport label for the different nodes is as follows.

For iLER PE-3, the LDP transport label for traffic toward PE-6 is 524286:

```
[/]
A:admin@PE-3# show router ldp bindings active prefixes prefix 192.0.2.6/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.3)
(IPv6 LSR ID ::)
=====
---snip---
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.6/32                         Push
--                                   524286
192.168.23.1                         1/1/c1/2

192.0.2.6/32                         Swap
524285                                524286
192.168.23.1                         1/1/c1/2
-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

For LSR PE-2, the LDP transport label toward PE-6 is 524281:

```
[/]
A:admin@PE-2# show router ldp bindings active prefixes prefix 192.0.2.6/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.2)
(IPv6 LSR ID ::)
=====
---snip---
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.6/32                         Push
--                                   524281
192.168.24.2                         1/1/c1/4

192.0.2.6/32                         Swap
524286                                524281
192.168.24.2                         1/1/c1/4
-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

For LSR PE-4, the LDP transport label toward PE-6 is 524282:

```
[/]
A:admin@PE-4# show router ldp bindings active prefixes prefix 192.0.2.6/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.4)
(IPv6 LSR ID ::)
=====
---snip---
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.6/32                         Push
--                                   524282
192.168.45.2                         1/1/c1/1

192.0.2.6/32                         Swap
524281                               524282
192.168.45.2                         1/1/c1/1

-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

For LSR PE-5, the LDP transport label toward PE-6 is 524282:

```
[/]
A:admin@PE-5# show router ldp bindings active prefixes prefix 192.0.2.6/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.5)
(IPv6 LSR ID ::)
=====
---snip---
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.6/32                         Push
--                                   524282
192.168.15.1                         1/1/c1/3

192.0.2.6/32                         Swap
524282                               524282
192.168.15.1                         1/1/c1/3

-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

For LSR PE-1, the LDP transport label toward PE-6 is 524287, but this label will not be present in the traceroute detailed output, because this message cannot time out on an LSR where ICMP tunneling takes place:

```
[/]
A:admin@PE-1# show router ldp bindings active prefixes prefix 192.0.2.6/32

=====
LDP Bindings (IPv4 LSR ID 192.0.2.1)
      (IPv6 LSR ID ::)
=====
---snip---
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.6/32                          Push
--                                     524287
192.168.16.2                          1/1/c1/2

192.0.2.6/32                          Swap
524282                                  524287
192.168.16.2                          1/1/c1/2

-----
No. of IPv4 Prefix Active Bindings: 2
=====
```

Conclusion

Tunneling of ICMP reply messages over MPLS LSPs provides the ability to trace the hops in an MPLS tunnel. This mechanism applies to VPRN, 6PE/6VPE, and BGP labeled routes.

ICMP tunneling at an LSR applies to UDP traceroute packets that time out at the LSR. The ICMP Time Exceeded message is tunneled by the LSR toward the eLER and the eLER routes the packet to the sender of the traceroute message.

Unnumbered Interfaces in RSVP-TE and LDP

This chapter provides information about unnumbered interfaces in RSVP-TE and LDP.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter is applicable to SR OS routers and was initially written for SR OS Release 13.0.R7. The CLI in this edition corresponds to SR OS Release 22.10.R3. SR OS supports unnumbered interfaces in RSVP-TE and LDP in SR OS Release 11.0.R1 or later.

Overview

Unnumbered interfaces enable IP processing on a point-to-point (P2P) interface without an explicit IP address. Unnumbered interfaces are supported in ReSource reserVation Protocol with Traffic Engineering (RSVP-TE) and Label Distribution Protocol (LDP).

An unnumbered interface is uniquely identified in the network by the tuple (Router ID, If Index), where the interface index (If Index) is unique on the router. The two endpoints of an unnumbered link exchange the If Index that they assigned to the link.

The (Router ID, If Index) tuple is used by the following:

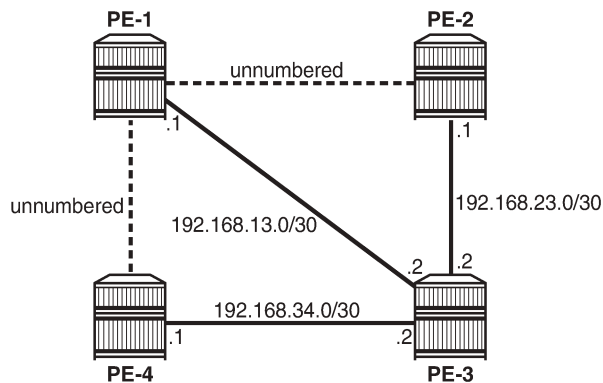
- Intermediate System to Intermediate System (IS-IS) or Open Shortest Path First (OSPF) to advertise link information
- RSVP to signal Label Switched Paths (LSPs) over this unnumbered interface
- LDP to establish hello adjacencies and resolve Forwarding Equivalence Classes (FECs)
- Operations, Administration, and Maintenance (OAM) to send or respond to a Multi-Protocol Label Switching (MPLS) echo request over an unnumbered interface

The unnumbered interface can "borrow" the IP address of another interface on the node.

The borrowed IP address is used exclusively as the source address for IP packets that are originated from the unnumbered interface. The borrowed IP address defaults to the system loopback interface address, but can be changed manually. The borrowed IP address corresponds to the Router ID in the tuple representing the unnumbered interface.

The configuration used in this chapter is shown in [Figure 313: Example topology for unnumbered interfaces in RSVP and LDP](#). There are two unnumbered links: one between PE-1 and PE-2, and another between PE-1 and PE-4. The remaining links are numbered.

Figure 313: Example topology for unnumbered interfaces in RSVP and LDP



25683

Configure unnumbered interfaces as follows:

```
configure {
  router <router-name> {
    interface <interface-name> {
      ipv4 {
        unnumbered {
          system|ip-address <ip-address>|ip-int-name <ip-int-name>
        }
      }
    }
  }
  exit all
}
```

To configure an unnumbered link with the system address as the borrowed IP address, no address needs to be configured:

```
# on PE-4: configure {
  router "Base" {
    interface "int-PE-4-PE-1" {
      port 1/1/c1/1
      ipv4 {
        unnumbered {
          system
        }
      }
    }
  }
  exit all
}
```

An unnumbered interface has to be a P2P link.

Unnumbered interfaces in IS-IS

Unnumbered interfaces are identified in IS-IS by a combination of the system ID and an extended local circuit ID, as described in RFC 5307, *IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)*.

Enable debugging on the unnumbered interface on PE-4 (int-PE-4-PE-1) as follows:

```
# on PE-4:
debug
  router
    isis
      packet "int-PE-4-PE-1" detail
```



```
exit all
```

The following IS-IS hello Protocol Data Unit (PDU) is received from PE-1. The I/F Address is the borrowed IP address; in this case, the system address of PE-1, because the interface is unnumbered.

```
# on PE-4:
2 2023/03/21 12:42:33.919 UTC MINOR: DEBUG #2001 Base ISIS PKT
"ISIS PKT:
RX ISIS PDU ifId 2 len 52:
  DMAC      : 09:00:2b:00:00:05
  Proto Disc : 131
  Header Len : 20
  Version PID : 1
  ID Length  : 0
  Version    : 1
  Reserved   : 0
  Max Area Addr : 3
  PDU Type   : (11) Point-2-Point IS-IS Hello Pdu
  Circuit Type : L1
  Source Id   : 19 20 00 00 20 01
  Hold Time   : 27
  Packet length : 52
  Circuit Id   : 0
  Area Addresses:
    Area Address : (3) 49.0001
  Supp Protocols:
    Protocols    : IPv4
  I/F Addresses :
    I/F Address : 192.0.2.1
  3Way Adjacency :
    State        : UP
    Ext ckt ID   : 4
    NbrSysID     : 19 20 00 00 20 04
    Nbr ext ckt ID : 2
"
```

The three-way adjacency contains the neighbor extended local circuit ID (**Nbr ext ckt ID: 2**). This is the local interface index of the unnumbered interface (int-PE-4-PE-1), which can be verified as follows:

```
[/]
A:admin@PE-4# show router interface "int-PE-4-PE-1" detail | match "If Index"
If Index      : 2                Virt. If Index   : 2
Last Oper Chg : 03/21/2023 23:55:10 Global If Index  : 1
```

On PE-1, the interface toward PE-4 has a different index, as follows:

```
[/]
A:admin@PE-1# show router interface "int-PE-1-PE-4" detail | match "If Index"
If Index      : 4                Virt. If Index   : 4
Last Oper Chg : 03/21/2023 23:54:36 Global If Index  : 3
```

For numbered interfaces, such as int-PE-4-PE-3, the I/F Address is the interface address; in that case, 192.168.34.1, for messages received from PE-3 instead of the Router ID.

When a Shared Risk Link Group (SRLG) is configured in combination with IS-IS and unnumbered interfaces, the least significant bit in the flags field of the SRLG Type-Length Value (TLV) indicates that the interface is unnumbered (0) or numbered (1).

Unnumbered interfaces in OSPF

For unnumbered interfaces in OSPF, link local and remote identifiers are defined in RFC 4203, *OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)*. The OSPF link state advertisement (LSA) is defined in RFC 2328, *OSPF version 2*.

For numbered interfaces, the link data is the IP interface address; for unnumbered interfaces, the link data is the interface index value. The value starts from 1 in the format 0.0.0.1. SR OS recognizes an unnumbered interface when the first byte in the link data has a value of 0; SR OS then treats the link data as an interface index instead of an IP address.

Unnumbered interfaces in RSVP-TE

Unnumbered IP interfaces can be used as Traffic Engineering (TE) links for the signaling of RSVP P2P LSPs and point-to-multipoint (P2MP) LSPs.

Fast Reroute (FRR) facility backup over unnumbered interfaces is supported, whereas FRR one-to-one only uses numbered interfaces in the detour path.

The unnumbered IP address is advertised by IS-IS or OSPF, and Constrained Shortest Path First (CSPF) can include them in the computation of a path.

Unnumbered interfaces of the remote router can be specified in the Explicit Route Object (ERO), and in the Record Route Object (RRO), by a combination of Router ID (borrowed IP address) and interface ID, as defined in RFC 3477, *Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)*.

The choice of the data interface is indicated in the Path message by including the interface identifier of the data channel. In the Path message (**PATH Msg**), the IP address equals the local borrowed IP address; in the Resv message (**RESV Msg**), the IP address is the remote borrowed IP address. As well as the borrowed IP address, there is also a Logical Interface Handle (**LIH**). This interface identification is defined in RFC 3473, *Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions*.

To see the Path and Resv messages on PE-4, enter the following debug commands:

```
# on PE-4:
debug
  router
    rsvp
      packet
        path detail
        resv detail
      exit all
```

The Path message contains the **RSVPHop** object with the local interface identifier of the data channel, as follows:

```
# on PE-4:
19 2023/03/21 13:03:45.671 UTC MINOR: DEBUG #2001 Base RSVP
"RSVP: PATH Msg
Send PATH From:192.0.2.4, To:192.0.2.2
      TTL:255, Checksum:0xa648, Flags:0x0
Session   - EndPt:192.0.2.2, TunnId:1, ExtTunnId:192.0.2.4
SessAttr  - Name:LSP-PE-4-PE-2::dyn
           SetupPri:7, HoldPri:0, Flags:0x17
```

```

RSVPHop - Ctype:3, Addr:192.1.2.4, LIH:2
           RouterId :192.0.2.4, InterfaceId :2
TimeValue - RefreshPeriod:30
SendTempl - Sender:192.0.2.4, LspId:2576
SendTSpec - Ctype:QOS, CDR:0.000 bps, PBS:0.000 bps, PDR:infinity
           MPU:20, MTU:1564
LabelReq  - IfType:General, L3ProtID:2048
RRO      - Unnumbered: RouterId 192.0.2.4 InterfaceID 2, Flags:0x0
ERO      - Unnumbered RouterId 192.0.2.1, LinkId 4, Strict
           Unnumbered RouterId 192.0.2.2, LinkId 2, Strict
FRR0bj    - SetupPri:7, HoldPri:0, HopLimit:16, BW:0.000 bps, Flags:0x2
           ExcAny:0x0, IncAny:0x0, IncAll:0x0
"
    
```

The ERO and RRO objects are also shown. The unnumbered interfaces are defined by the combination of the Router ID (**RouterId**) and the interface ID (**InterfaceId**).

The Resv message also contains the **RSVPHop** object, but the address is now the remote address of PE-1 instead of the local address of PE-4, as follows:

```

# on PE-4:
23 2023/03/21 13:04:03.396 UTC MINOR: DEBUG #2001 Base RSVP
"RSVP: RESV Msg
Recv RESV From:192.1.2.1, To:192.0.2.4
           TTL:255, Checksum:0xbe23, Flags:0x0
Session   - EndPt:192.0.2.2, TunnId:1, ExtTunnId:192.0.2.4
RSVPHop - Ctype:3, Addr:192.1.2.1, LIH:2
           RouterId :192.0.2.1, InterfaceId :4
TimeValue - RefreshPeriod:30
Style     - SE
FlowSpec  - Ctype:QOS, CDR:0.000 bps, PBS:0.000 bps, PDR:infinity
           MPU:20, MTU:1560, RSpecRate:0, RSpecSlack:0
FilterSpec - Sender:192.0.2.4, LspId:2576, Label:524287
RRO      - Unnumbered: RouterId 192.0.2.1 InterfaceID 4, Flags:0x1
           Label:524287, Flags:0x1
           Unnumbered: RouterId 192.0.2.2 InterfaceID 2, Flags:0x0
           Label:524287, Flags:0x1
"
    
```

To see the Patherr (**PATHERR Msg**) and Resverr (**RESVERR Msg**) messages on PE-4, enter the following debug command:

```

# on PE-4:
debug
  router
    rsvp
      packet
        patherr detail
        resverr detail
      exit all
    
```

The Resverr message contains the following **ErrorSpec** object, as defined by RFC 3473. In this case, the error is caused by disabling TE on ingress Label Egress Router (iLER) PE-4. No route can be found to the destination because there is no lookup in the TE database. The LSP does not come up, even if CSPF is disabled on the LSP.

```

# on PE-4:
32 2023/03/21 13:05:11.398 UTC MINOR: DEBUG #2001 Base RSVP
"RSVP: RESVERR Msg
Send RESVERR From:192.1.2.4, To:192.0.2.1
           TTL:255, Checksum:0x68c1, Flags:0x0
"
    
```

```
Session      - EndPt:192.0.2.2, TunnId:1, ExtTunnId:192.0.2.4
RSVPHop     - Ctype:3, Addr:192.1.2.4, LIH:2
              RouterId :192.0.2.4, InterfaceId :2
ErrorSpec   - Ctype:3, ErrNode:192.1.2.4, Flags:0x0, ErrCode:3, ErrValue:0
              RouterId :192.0.2.4, InterfaceId :2
Style       - SE
FlowSpec    - Ctype:QOS, CDR:0.000 bps, PBS:0.000 bps, PDR:infinity
              MPU:20, MTU:1560, RSpecRate:0, RSpecSlack:0
FilterSpec  - Sender:192.0.2.4, LspId:2576
"
```

Considerations for unnumbered interfaces in RSVP-TE

Consider the following for unnumbered interfaces in RSVP-TE:

- With RSVP, TE must be enabled in IS-IS or OSPF. The Router ID of the router that advertised an unnumbered interface index is obtained from the TE database. Therefore, if TE is disabled in IS-IS or OSPF, a non-CSPF LSP with the next hop for this path over an unnumbered interface does not come up. The Router ID of the neighbor that has the next hop of the Path message cannot be searched for.
 - The operational state of the LSP path remains down with reason **noRouteToDestination**.
 - If a Path message is received at the LSR in which TE is disabled and the next hop for the LSP path is over an unnumbered interface, a PathErr message is sent back to the iLER with error code 24: Routing problem; Error value 5: "No route available toward destination".
- Only FRR facility protection is supported; FRR one-to-one protection only works for numbered interfaces.
- There is no FRR facility protection if the point of local repair (PLR) is the iLER and the bypass tunnel egress interface is unnumbered.
- Bi-directional Forwarding Detection (BFD) can be enabled on an unnumbered router interface. Therefore, RSVP FRR procedures can be triggered via a BFD session timeout.
- Unnumbered interfaces cannot be configured as hops in a path. This is true for RSVP-TE LSPs, as well as for static LSPs.
- RSVP hello and hello-related capabilities, such as graceful restart helper, are not supported.
- SRLG is supported, but the user SRLG DB (**user-srlg-db**) feature at the iLER is not supported. Unnumbered interfaces cannot be added to the SRLG DB. When the user SRLG DB feature is enabled on the iLER, all unnumbered interfaces are considered as having no SRLG membership.

Unnumbered interfaces in LDP

LDP can establish hello adjacencies and can resolve unicast and multicast FECs over unnumbered interfaces.

For link LDP, hello adjacencies are brought up using hello packets with source IP address set to the borrowed IP address and a destination IP address set to 224.0.0.2. The borrowed IP address is the system address, by default. Hello packets with the same source IP address are accepted when received over parallel unnumbered interfaces from the same peer LSR ID. The corresponding hello adjacencies are associated with a single LDP session.

The transport address for the TCP connection, which is encoded in the hello packet, is always set to the LSR ID of the node. The user can configure the **local-lsr-id** option on the interface and change the value

of the LSR ID to either the local interface or some other interface name: loopback or not, numbered or not. The transport address for the LDP session is updated with the new LSR ID.

For targeted LDP, the source and destination addresses of targeted hello packets are the LDP LSR IDs of the nodes. The user can configure the **local-lsr-id** option on the targeted session. The transport address for the LDP session and the source IP address of targeted hello messages are updated to the new LSR ID value.

LDP advertises/withdraws unnumbered interfaces using the address/address-withdraw messages. The borrowed IP address of the interface is used.

A FEC can be resolved to an unnumbered interface in the same way as it is resolved to a numbered interface. The outgoing interface and the next hop are searched for in the Routing Table Manager (RTM). The next hop consists of the Router ID and link identifier of the interface to the peer LSR. All LDP FEC types are supported. LDP FEC Equal Cost Multi-Path (ECMP) over a mix of unnumbered and numbered interfaces is supported.

RFC 5036, *LDP Specification*, describes the address list TLV that is used in the LDP address message, and the LDP address withdrawal message. For unnumbered interfaces, the borrowed IP address is used, which is typically the system address of the sender node.

On PE-1, enable debugging for LDP packets from peer 192.0.2.2 as follows:

```
# on PE-1:
debug
  router
    ldp
      peer 192.0.2.2
        packet
          init detail
          label detail
        exit all
```

The following LDP address packets are shown at PE-1:

```
6 2023/03/21 13:23:27.615 UTC MINOR: DEBUG #2001 Base LDP
"LDP: LDP
Send Address packet (msgId 5) to 192.0.2.2:0
Protocol version = 1
Address Family = 1 Number of addresses = 3
Address 1 = 192.0.2.1
Address 2 = 192.1.2.1
Address 3 = 192.168.13.1
"
```

```
5 2023/03/21 13:23:27.436 UTC MINOR: DEBUG #2001 Base LDP
"LDP: LDP
Recv Address packet (msgId 5) from 192.0.2.2:0
Protocol version = 1
Address Family = 1 Number of addresses = 3
Address 1 = 192.0.2.2
Address 2 = 192.1.2.2
Address 3 = 192.168.23.1
"
```

The received LDP address packet contains three IP addresses: the system IP address 192.0.2.2 for an unnumbered interface on the sending node, the loop back IP address 192.1.2.2 for an unnumbered loop back interface on the sending node, and the interface IP address 192.168.23.1 for a numbered interface on the sending node.

Considerations for unnumbered interfaces in LDP

All LDP features are supported on unnumbered interfaces, except for the following:

- BFD can be enabled on an unnumbered router interface. The BFD parameters must be configured within the unnumbered **interface** context. If not, the BFD sessions are not established.
- Unnumbered interfaces cannot be added into LDP global and peer prefix policies.

Unnumbered interfaces in OAM

The following applies to unnumbered interfaces in RSVP-TE or LDP.

The downstream mapping object is a TLV that can be included in an echo request, as described in RFC 4379, *Detecting Multi-Protocol Label Switched Data Plane Failures*.

Only one downstream mapping object may appear in an echo request. The presence of a downstream mapping object is a request that a downstream mapping object be included in the echo reply.

For unnumbered interfaces, the address type is 2 (**ipv4Unnumbered**), the downstream IP address is the borrowed IP address of the downstream LSR, and the downstream interface address is the index assigned by the upstream LSR to the interface.

The downstream detailed mapping object is a TLV that can be included in an echo request, as described in RFC 6424, *Mechanism for Performing Label Switched Path Ping (LSP Ping) over MPLS Tunnels*.

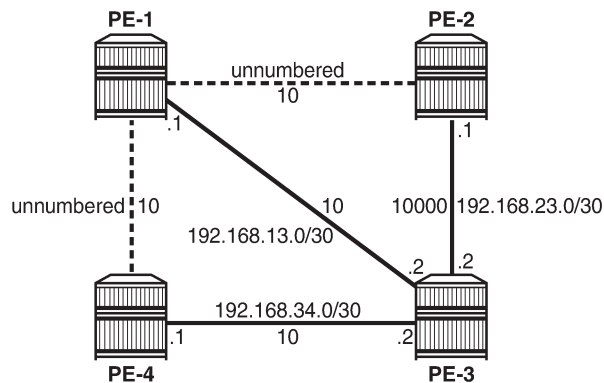
The following output shows the detailed LSP trace for an RSVP LSP from PE-4 to PE-2. Two unnumbered interfaces are used: the first between PE-4 and PE-1 and the second between PE-1 and PE-2. The interface type (**iftype**) is **ipv4Unnumbered**.

```
A:PE-4# oam lsp-trace "LSP-PE-4-PE-2" detail
lsp-trace to LSP-PE-4-PE-2: 0 hops min, 0 hops max, 116 byte packets
0 192.0.2.4
   DS 1: ipaddr=0.0.0.0 iftype=ipv4Unnumbered MRU=1564
       label[1]=524287 protocol=4(RSVP-TE)
1 192.0.2.1  rtt=4.09ms rc=8(DSRtrMatchLabel) rsc=1
   DS 1: ipaddr=0.0.0.0 ifaddr=2 iftype=ipv4Unnumbered MRU=1564
       label[1]=524287 protocol=4(RSVP-TE)
2 192.0.2.2  rtt=1.42ms rc=3(EgressRtr) rsc=1
```

Configuration

The following configuration example is for unnumbered interfaces in RSVP and LDP; see [Figure 314: Configuration example for unnumbered Interfaces in RSVP and LDP](#). The nodes are 7750 SRs.

Figure 314: Configuration example for unnumbered Interfaces in RSVP and LDP



25684

All interfaces have a TE metric of 10, while the link between PE-2 and PE-3 has a TE metric of 10000. So, the preferred path from PE-4 to PE-2 is over the unnumbered interfaces between PE-4 and PE-1 and between PE-1 and PE-2.

Unnumbered interfaces

Router interfaces are configured on all nodes, numbered and unnumbered. Initially, the unnumbered interfaces are configured with default settings. The following interfaces are configured on PE-1:

```
# on PE-1: configure {
  router "Base" {
    interface "int-PE-1-PE-2" {
      port 1/1/c1/1
      ipv4 {
        unnumbered {
          system
        }
      }
    }
    interface "int-PE-1-PE-3" {
      port 1/1/c1/3
      ipv4 {
        primary {
          address 192.168.13.1
          prefix-length 30
        }
      }
    }
    interface "int-PE-1-PE-4" {
      port 1/1/c1/2
      ipv4 {
        unnumbered {
          system
        }
      }
    }
  }
  interface "system" {
    ipv4 {
      primary {
        address 192.0.2.1
      }
    }
  }
}
```

```

        prefix-length 32
    }
}
exit all

```

There are two unnumbered interfaces: int-PE-1-PE-2 and int-PE-1-PE-4. There is no borrowed IP address configured for the unnumbered interfaces. So, the borrowed IP address is the system address of PE-1.

For the unnumbered interfaces, the borrowed IP address is indicated between square brackets in the following output:

```

[/]
A:admin@PE-1# show router interface

=====
Interface Table (Router: Base)
=====
Interface-Name      Adm      Opr(v4/v6)  Mode      Port/SapId
IP-Address          PfxState
-----
int-PE-1-PE-2      Up        Up/Down     Network  1/1/c1/1
  Unnumbered If[system]                n/a
int-PE-1-PE-3      Up        Up/Down     Network  1/1/c1/3
  192.168.13.1/30                       n/a
int-PE-1-PE-4      Up        Up/Down     Network  1/1/c1/2
  Unnumbered If[system]                n/a
system              Up        Up/Down     Network  system
  192.0.2.1/32                          n/a
-----
Interfaces : 4
=====

```

Each interface, numbered or unnumbered, gets an interface index. This interface index can be retrieved as follows:

```

[/]
A:admin@PE-1# show router interface "int-PE-1-PE-2" detail | match "If Index"
If Index      : 2                Virt. If Index  : 2
Last Oper Chg : 03/21/2023 23:54:36 Global If Index : 1

[/]
A:admin@PE-1# show router interface "int-PE-1-PE-3" detail | match "If Index"
If Index      : 3                Virt. If Index  : 3
Last Oper Chg : 03/21/2023 23:54:36 Global If Index : 2

[/]
A:admin@PE-1# show router interface "int-PE-1-PE-4" detail | match "If Index"
If Index      : 4                Virt. If Index  : 4
Last Oper Chg : 03/21/2023 23:54:36 Global If Index : 3

[/]
A:admin@PE-1# show router interface "system" detail | match "If Index"
If Index      : 1                Virt. If Index  : 1
Last Oper Chg : 03/21/2023 23:54:36 Global If Index : 256

```

The unnumbered interface toward PE-2 has If Index 2, and the unnumbered interface toward PE-4 has If Index 4.

BFD can be enabled on unnumbered interfaces, as follows:

```
# on PE-4: configure {
  router "Base" {
    interface "int-PE-4-PE-1" {
      ipv4 {
        bfd {
          admin-state enable
          transmit-interval 100
          receive 100
          multiplier 3
        }
      }
    }
  }
  exit all
```

The BFD parameters must be configured within the unnumbered **interface** context. If not, the BFD sessions are not established.

An Interior Gateway Protocol (IGP) must be configured. In this case, IS-IS is chosen. OSPF could have been used equally well. TE must be enabled for unnumbered interfaces used in RSVP, even when CSPF is disabled. The IS-IS configuration on PE-1 is as follows:

```
# on PE-1: configure {
  router "Base" {
    isis 0 {
      admin-state enable
      level-capability 1
      traffic-engineering true
      area-address [49.0001]
      interface "int-PE-1-PE-2" {
        interface-type point-to-point
      }
      interface "int-PE-1-PE-3" {
        interface-type point-to-point
      }
      interface "int-PE-1-PE-4" {
        interface-type point-to-point
      }
      interface "system" {
      }
    }
  }
  exit all
```

An unnumbered interface has to be a P2P link.

The TE database contains the Router ID and the If Index for unnumbered interfaces, as follows:

```
[/]
A:admin@PE-1# show router isis database PE-2.00-00 detail

=====
Rtr Base ISIS Instance 0 Database (detail)
=====

Displaying Level 1 database
-----
LSP ID      : PE-2.00-00          Level      : L1
Sequence    : 0x2                Checksum   : 0x1c5a   Lifetime   : 1049
Version     : 1                  Pkt Type  : 18       Pkt Ver    : 1
Attributes  : L1                 Max Area  : 3         Alloc Len  : 126
SYS ID      : 1920.0000.2002     SysID Len : 6         Used Len   : 126

TLVs :
  Area Addresses:
    Area Address : (3) 49.0001
```

```

Supp Protocols:
  Protocols      : IPv4
  IS-Hostname    : PE-2
  Router ID     :
    Router ID    : 192.0.2.2
  I/F Addresses :
    I/F Address  : 192.168.23.1
    I/F Address  : 192.0.2.2
  TE IS Nbrs   :
    Nbr         : PE-1.00
    Default Metric : 10
    Sub TLV Len  : 10
    LcLId       : 2
    RmtId       : 2
  TE IS Nbrs   :
    Nbr         : PE-3.00
    Default Metric : 10000
    Sub TLV Len  : 12
    IF Addr     : 192.168.23.1
    Nbr IP      : 192.168.23.2
---snip---
=====

```

PE-2 has an unnumbered interface toward neighbor PE-1 (**Nbr: PE-1.00**), with local interface index 2 (**LcLId: 2**) and remote interface index 2 (**RmtId: 2**). For the numbered interface toward neighbor PE-3, the local and remote interface IP addresses are shown (**IF Addr** and **Nbr IP**), not the interface index.

Unnumbered interfaces in RSVP

MPLS and RSVP must be enabled on the interfaces on the nodes. TE metrics are configured on the MPLS interfaces. For node PE-4, the configuration is as follows:

```

# on PE-4: configure {
  router "Base" {
    mpls {
      admin-state enable
      interface "int-PE-4-PE-1" {
        te-metric 10
      }
      interface "int-PE-4-PE-3" {
        te-metric 10
      }
    }
    rsvp {
      admin-state enable
      interface "int-PE-4-PE-1" {
      }
      interface "int-PE-4-PE-3" {
      }
    }
  }
exit all

```

An LSP is configured from PE-4 to PE-2 with CSPF enabled and using the TE metrics, not the IGP metrics. Unnumbered interfaces cannot be configured as hops in a path. A dynamic path **"dyn"**, without any hops, is configured to be used in an LSP from PE-4 to PE-2, as follows:

```

# on PE-4: configure {
  router "Base" {
    mpls {

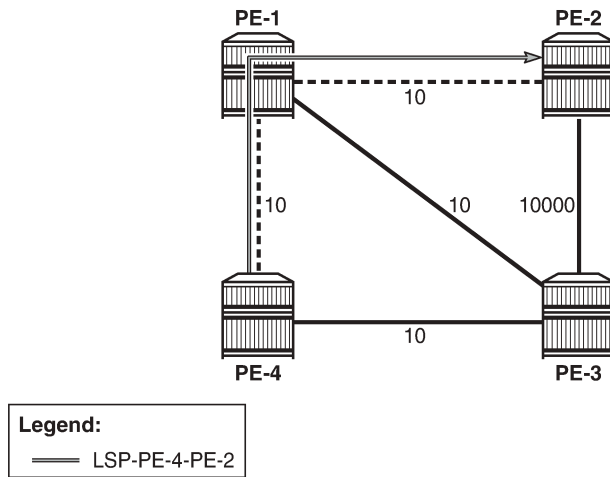
```

```

path "dyn" {
    admin-state enable
}
lsp "LSP-PE-4-PE-2" {
    admin-state enable
    type p2p-rsvp
    to 192.0.2.2
    path-computation-method local-cspf
    metric-type te
    primary "dyn" {
    }
}
exit all
    
```

The LSP from PE-4 to PE-2 has TE metric 20 when the next hop is PE-1, and TE metric 30 or 10010 when the next hop is PE-3. [Figure 315: LSP-PE-4-PE-2 on unnumbered interfaces](#) shows LSP-PE-4-PE-2, which uses only unnumbered interfaces.

Figure 315: LSP-PE-4-PE-2 on unnumbered interfaces



25685

The following tunnel table shows a next hop `int-PE-4-PE-1`, which implies that it is an unnumbered interface. The only unnumbered interface at PE-4 is `int-PE-4-PE-1`. The metric in this tunnel table is **16777215** because the IGP metric is not used.

```

[/]
A:admin@PE-4# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.2/32     rsvp      MPLS  1          7      int-PE-4-PE-1 16777215
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====
    
```

The actual and computed hops can be verified, as well as the CSPF metric (TE metric), as follows:

```
[/]
A:admin@PE-4# show router mpls lsp "LSP-PE-4-PE-2" path detail

=====
MPLS LSP LSP-PE-4-PE-2 Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use
  b - Bandwidth Protected      n - Node Protected
  s - Soft Preemption
  S - Strict                    L - Loose
  A - ABR                      + - Inherited
=====

-----
LSP LSP-PE-4-PE-2
Path dyn
-----
LSP Name      : LSP-PE-4-PE-2
From          : 192.0.2.4
To           : 192.0.2.2
Admin State   : Up                Oper State    : Up
Path Name    : dyn
Path LSP ID  : 12288              Path Type     : Primary
Path Admin   : Up                Path Oper     : Up
Out Interface : 1/1/c1/1          Out Label     : 524287
---snip---
Explicit Hops :
  No Hops Specified
Actual Hops   :
  192.0.2.4, If Index : 2          Record Label  : N/A
  -> 192.0.2.1, If Index : 4      Record Label  : 524287
  -> 192.0.2.2, If Index : 2      Record Label  : 524287
Computed Hops :
  192.0.2.4, If Index : 2(S)
  -> 192.0.2.1, If Index : 4(S)
  -> 192.0.2.2, If Index : 2(S)
Resignal Eligible: False
Last Resignal   : n/a              CSPF Metric   : 20
=====
```

The computed hops are strict hops, as indicated by **(S)**. Because the interfaces are unnumbered, the system address and the If Index are displayed. The CSPF metric is 20.

Configuring the borrowed IP address

The borrowed IP address does not need to be the system address, but the address must exist on the node. When the unnumbered interface is configured with a borrowed IP address that does not exist on the node, the interface goes down. This can be verified by assigning a non-existent address to the unnumbered interface int-PE-1-PE-2, as follows:

```
# on PE-1: configure {
  router "Base" {
    interface "int-PE-1-PE-2" {
      ipv4 {
        unnumbered {
          ip-address 192.1.2.1
        }
      }
    }
  }
}
```

The operational state of this interface goes down, which can be verified as follows:

```
[/]
A:admin@PE-1# show router interface

=====
Interface Table (Router: Base)
=====
Interface-Name          Adm      Opr(v4/v6)  Mode      Port/SapId
IP-Address              PfxState
-----
int-PE-1-PE-2         Up      Down/Down  Network 1/1/c1/1
Unnumbered If[192.1.2.1]
192.168.13.1/30        Up       Up/Down     Network  1/1/c1/3
n/a
int-PE-1-PE-4         Up       Up/Down     Network  1/1/c1/2
Unnumbered If[system]
system                 Up       Up/Down     Network  system
192.0.2.1/32          n/a
-----
Interfaces : 4
=====
```

The borrowed IP address is indicated between square brackets. The interface is down because the IP address is not known on PE-1. The down reason code **noIfAddress** can be retrieved as follows:

```
[/]
A:admin@PE-1# show router interface "int-PE-1-PE-2" detail | match "Down Reason Code"
Down Reason Code : noIfAddress
```

The IP address can be configured as a loopback address on PE-1 and assigned to all unnumbered interfaces, as follows:

```
# on PE-1: configure {
  router "Base" {
    interface "loopback1" {
      loopback
      ipv4 {
        primary {
          address 192.1.2.1
          prefix-length 32
        }
      }
    }
    interface "int-PE-1-PE-2" {
      port 1/1/c1/1
      ipv4 {
        unnumbered {
          ip-address 192.1.2.1
        }
      }
    }
    interface "int-PE-1-PE-4" {
      port 1/1/c1/2
      ipv4 {
        unnumbered {
          ip-address 192.1.2.1
        }
      }
    }
  }
}
exit all
```

When the borrowed IP address is known on node PE-1, the unnumbered interface is operationally up, which can be verified as follows:

```
[/]
A:admin@PE-1# show router interface

=====
Interface Table (Router: Base)
=====
Interface-Name      Adm    Opr(v4/v6)  Mode    Port/SapId
IP-Address          PfxState
-----
int-PE-1-PE-2      Up     Up/Down    Network 1/1/c1/1
Unnumbered If[192.1.2.1]      n/a
int-PE-1-PE-3      Up      Up/Down     Network  1/1/c1/3
192.168.13.1/30    n/a
int-PE-1-PE-4      Up     Up/Down    Network 1/1/c1/2
Unnumbered If[192.1.2.1]      n/a
loopback1         Up     Up/Down    Network loopback
192.1.2.1/32      n/a
system             Up      Up/Down     Network  system
192.0.2.1/32      n/a
-----
Interfaces : 5
=====
```

In a similar way, the borrowed IP address on PE-2 is configured as 192.1.2.2 and on PE-4 as 192.1.2.4.

TE required for unnumbered interfaces in RSVP

For unnumbered interfaces, the IGP looks up the Router ID in the TE database. Therefore, TE must be enabled even if CSPF is disabled.

TE is disabled in IS-IS and CSPF is disabled in the LSP on PE-4, as follows:

```
# on PE-4: configure {
  router "Base" {
    isis 0 {
      delete traffic-engineering
    }
    mpls {
      lsp "LSP-PE-4-PE-2" {
        admin-state disable
        delete metric-type
        delete path-computation-method
      }
    }
  }
  exit all
}

configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-4-PE-2" {
        admin-state enable
      }
    }
  }
  exit all
}
```

LSP-PE-4-PE-2 is operationally down with failure code **noRouteToDestination**, which can be verified as follows:

```
[/]
A:admin@PE-4# show router mpls lsp "LSP-PE-4-PE-2" path detail

=====
MPLS LSP LSP-PE-4-PE-2 Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use
  b - Bandwidth Protected      n - Node Protected
  s - Soft Preemption
  S - Strict                    L - Loose
  A - ABR                      + - Inherited
=====
-----
LSP LSP-PE-4-PE-2
Path dyn
-----
LSP Name      : LSP-PE-4-PE-2
From          : 192.0.2.4
To           : 192.0.2.2
Admin State   : Up              Oper State      : Down
Path Name    : dyn
Path LSP ID  : 12294            Path Type       : Primary
Path Admin   : Up              Path Oper       : Down
---snip---
MetricType   : igp              Oper MetricType : N/A
---snip---
Failure Code : noRouteToDestination
Failure Node : 192.0.2.4
---snip---
=====
```

The configuration is restored by enabling TE in IS-IS and CSPF in the **lsp** context, as follows:

```
# on PE-4: configure {
  router "Base" {
    isis 0 {
      traffic-engineering true
    }
    mpls {
      lsp "LSP-PE-4-PE-2" {
        admin-state disable
        path-computation-method local-cspf
        metric-type te
      }
    }
  }
exit all

configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-4-PE-2" {
        admin-state enable
      }
    }
  }
exit all
```

FRR facility

FRR facility is enabled on the LSP as follows:

```
# on PE-4: configure {
router "Base" {
  mpls {
    lsp "LSP-PE-4-PE-2" {
      fast-reroute {
        frr-method facility
      }
    }
  }
}
exit all
```

The following LSP path detail output shows where an FRR detour is available (@) and in which node a bypass tunnel originates. The letter **n** indicates that a node is protected, as in hop 192.0.2.4. When there is a detour available, but there is no **n**, link protection is available, as in hop 192.0.2.1:

```
[/]
A:admin@PE-4# show router mpls lsp "LSP-PE-4-PE-2" path detail

=====
MPLS LSP LSP-PE-4-PE-2 Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use
  b - Bandwidth Protected      n - Node Protected
  s - Soft Preemption
  S - Strict                    L - Loose
  A - ABR                       + - Inherited
=====
-----
LSP LSP-PE-4-PE-2
Path dyn
-----
LSP Name      : LSP-PE-4-PE-2
From          : 192.0.2.4
To            : 192.0.2.2
Admin State   : Up                Oper State    : Up
Path Name     : dyn
Path LSP ID   : 12296              Path Type     : Primary
Path Admin    : Up                Path Oper     : Up
Out Interface : 1/1/c1/1           Out Label     : 524286
---snip---
Explicit Hops :
  No Hops Specified
Actual Hops   :
  192.0.2.4, If Index : 2 @ n      Record Label  : N/A
  -> 192.0.2.1, If Index : 4 @      Record Label  : 524286
  -> 192.0.2.2, If Index : 2        Record Label  : 524286
Computed Hops :
  192.0.2.4, If Index : 2(S)
  -> 192.0.2.1, If Index : 4(S)
  -> 192.0.2.2, If Index : 2(S)
Resignal Eligible: False
Last Resignal   : n/a              CSPF Metric   : 20
---snip---
=====
* indicates that the corresponding row element may have been truncated.
```

Information about the bypass tunnel originating in PE-4 can be retrieved as follows:

```
[/]
```



```
A:admin@PE-4# show router mpls bypass-tunnel protected-lsp detail
```

```
=====
MPLS Bypass Tunnels (Detail)
=====
```

```
-----
bypass-node192.0.2.1-61441
-----
```

```
To           : 192.168.23.1      State           : Up
Out I/F      : 1/1/c1/2         Out Label      : 524286
Up Time     : 0d 00:02:27      Active Time    : n/a
Reserved BW  : 0 Kbps          Protected LSP Count : 1
Type        : Dynamic          Bypass Path Cost : 10010
Setup Priority : 7              Hold Priority   : 0
Class Type   : 0
Exclude Node : None            Inter-Area     : False
Computed Hops :
  192.168.34.2(S)              Egress Admin Groups : None
  -> 192.168.34.1(S)           Egress Admin Groups : None
  -> 192.168.23.1(S)           Egress Admin Groups : None
Actual Hops  :
  192.168.34.2(192.0.2.4)      Record Label    : N/A
  -> 192.168.34.1(192.0.2.3)   Record Label    : 524286
  -> 192.168.23.1(192.0.2.2)   Record Label    : 524284
Last Resignal :
Attempted At : n/a             Resignal Reason : n/a
Resignal Status: n/a          Reason          : n/a
```

```
Protected LSPs -
```

```
LSP Name      : LSP-PE-4-PE-2::dyn
From          : 192.0.2.4      To              : 192.0.2.2
Avoid Node/Hop : 192.0.2.1    Downstream Label : 524286
Bandwidth     : 0 Kbps
```

This bypass tunnel, via PE-3 to PE-2, offers node protection for node PE-1. There are no unnumbered interfaces in this path. In a similar way, information about the bypass tunnel to protect the link between PE-1 and PE-2 can be retrieved in PE-1, as follows:

```
[/]
```

```
A:admin@PE-1# show router mpls bypass-tunnel protected-lsp detail
```

```
=====
MPLS Bypass Tunnels (Detail)
=====
```

```
-----
bypass-link192.0.2.2-61441
-----
```

```
To           : 192.168.23.1      State           : Up
Out I/F      : 1/1/c1/3         Out Label      : 524287
Up Time     : 0d 00:02:24      Active Time    : n/a
Reserved BW  : 0 Kbps          Protected LSP Count : 1
Type        : Dynamic          Bypass Path Cost : 10010
Setup Priority : 7              Hold Priority   : 0
Class Type   : 0
Exclude Node : None            Inter-Area     : False
Computed Hops :
  192.168.13.1(S)              Egress Admin Groups : None
  -> 192.168.13.2(S)           Egress Admin Groups : None
  -> 192.168.23.1(S)           Egress Admin Groups : None
Actual Hops  :
  192.168.13.1(192.0.2.1)      Record Label    : N/A
```

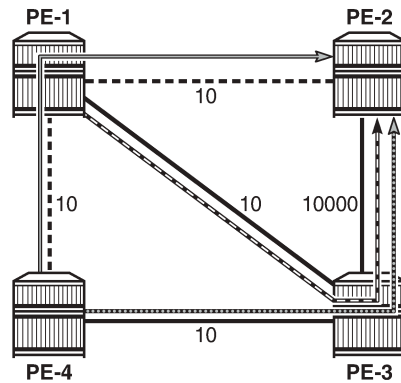
```

-> 192.168.13.2(192.0.2.3)      Record Label      : 524287
-> 192.168.23.1(192.0.2.2)    Record Label      : 524285
Last Resignal      :
Attempted At      : n/a          Resignal Reason   : n/a
Resignal Status   : n/a          Reason            : n/a

Protected LSPs -
LSP Name          : LSP-PE-4-PE-2::dyn
From              : 192.0.2.4    To                : 192.0.2.2
Avoid Node/Hop   : 192.0.2.2    Downstream Label  : 524286
Bandwidth         : 0 Kbps
    
```

Figure 316: LSP and FRR facility bypass tunnels shows the LSP and the two bypass tunnels: one in PE-4, offering node protection for node PE-1, and another in PE-1, bypassing the link between PE-1 and PE-2.

Figure 316: LSP and FRR facility bypass tunnels



Legend:

- LSP-PE-4-PE-2 Primary path
- - - - Bypass node PE-1
- == Bypass link between PE-1 and PE-2

25686

For each bypass tunnel, an additional RSVP session is set up. The following output shows that, in PE-4, two LSPs are signaled: the primary LSP and the bypass tunnel for node PE-1.

```

[/]
A:admin@PE-4# show router rsvp session

=====
RSVP Sessions
=====
RSVP Session Name
From          To          Tunnel ID  LSP ID    State
-----
LSP-PE-4-PE-2::dyn
192.0.2.4    192.0.2.2    1          12296     Up

bypass-node192.0.2.1-61441
192.0.2.4    192.168.23.1 61441     2          Up

-----
Sessions : 2
    
```

Similarly, PE-1 has an RSVP session for the primary LSP, but also for the bypass tunnel for the link toward PE-2, as follows:

```
[/]
A:admin@PE-1# show router rsvp session

=====
RSVP Sessions
=====
RSVP Session Name
  From           To           Tunnel ID   LSP ID     State
-----
LSP-PE-4-PE-2::dyn
192.0.2.4       192.0.2.2     1           12296      Up
bypass-link192.0.2.2-61441
192.0.2.1       192.168.23.1 61441       2           Up
-----
Sessions : 2
=====
```

PE-3 is only used by the bypass tunnels, as follows:

```
[/]
A:admin@PE-3# show router rsvp session

=====
RSVP Sessions
=====
RSVP Session Name
  From           To           Tunnel ID   LSP ID     State
-----
bypass-link192.0.2.2-61441
192.0.2.1       192.168.23.1 61441       2           Up
bypass-node192.0.2.1-61441
192.0.2.4       192.168.23.1 61441       2           Up
-----
Sessions : 2
=====
```

PE-2 terminates the LSP and the bypass tunnels, as follows:

```
[/]
A:admin@PE-2# show router rsvp session

=====
RSVP Sessions
=====
RSVP Session Name
  From           To           Tunnel ID   LSP ID     State
-----
LSP-PE-4-PE-2::dyn
192.0.2.4       192.0.2.2     1           12296      Up
bypass-link192.0.2.2-61441
192.0.2.1       192.168.23.1 61441       2           Up
-----
```

```
bypass-node192.0.2.1-61441
192.0.2.4          192.168.23.1          61441          2          Up
-----
Sessions : 3
=====
```

FRR one-to-one only supported on numbered interfaces

When FRR one-to-one is enabled on the LSP, the LSP does not use unnumbered interfaces. FRR is reconfigured on the LSP as follows:

```
# on PE-4: configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-4-PE-2" {
        fast-reroute {
          frr-method one-to-one
        }
      }
    }
  }
}
```

The LSP only comes up if it can use numbered interfaces end-to-end. In this case, the LSP takes the path via PE-3 with CSPF metric 10010, as follows:

```
[/]
A:admin@PE-4# show router mpls lsp "LSP-PE-4-PE-2" path detail

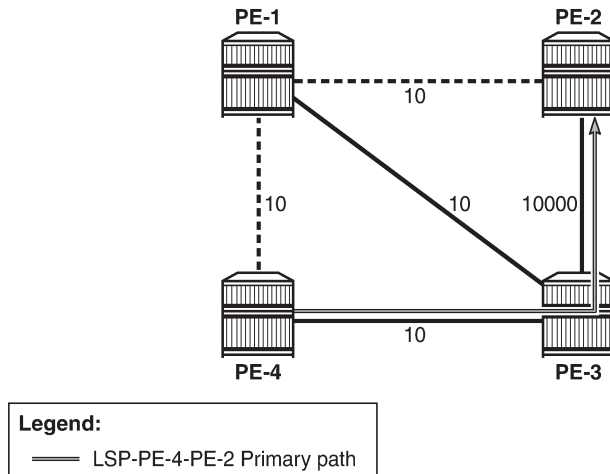
=====
MPLS LSP LSP-PE-4-PE-2 Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use
  b - Bandwidth Protected      n - Node Protected
  s - Soft Preemption          L - Loose
  S - Strict                    + - Inherited
  A - ABR

-----
LSP LSP-PE-4-PE-2
Path dyn
-----
LSP Name      : LSP-PE-4-PE-2
From          : 192.0.2.4
To           : 192.0.2.2
Admin State   : Up
Oper State    : Up
Path Name     : dyn
Path LSP ID   : 12300
Path Type     : Primary
Path Admin    : Up
Path Oper     : Up
Out Interface : 1/1/c1/2
Out Label     : 524287
---snip---
Explicit Hops :
  No Hops Specified
Actual Hops   :
  192.168.34.2(192.0.2.4) Record Label : N/A
  -> 192.168.34.1(192.0.2.3) Record Label : 524287
  -> 192.168.23.1(192.0.2.2) Record Label : 524287
Computed Hops :
  192.168.34.2(S)
  -> 192.168.34.1(S)
  -> 192.168.23.1(S)
Resignal Eligible: False
```

```
Last Resignal      : n/a                CSPF Metric       : 10010
=====
* indicates that the corresponding row element may have been truncated.
```

Figure 317: FRR one-to-one only supported on numbered interfaces shows the LSP in case of FRR one-to-one. Only numbered interfaces are used. Unfortunately, there is no bypass tunnel possible with only numbered interfaces; therefore, there is no protection.

Figure 317: FRR one-to-one only supported on numbered interfaces



25687

If there is no path available with only numbered interfaces, the LSP remains operationally down with failure code **noCspfRouteToDestination**". This can be verified by disabling port 1/1/c1/2 toward PE-3, as follows:

```
# on PE-4: configure {
  port 1/1/c1/2 {
    admin-state disable
  }
exit all
```

```
[/]
A:admin@PE-4# show router mpls lsp "LSP-PE-4-PE-2" path detail
```

```
=====
MPLS LSP LSP-PE-4-PE-2 Path (Detail)
=====
```

```
---snip---
```

```
LSP LSP-PE-4-PE-2
Path dyn
```

```
-----
LSP Name      : LSP-PE-4-PE-2
From          : 192.0.2.4
To            : 192.0.2.2
Admin State   : Up
Oper State    : Down
Path Name     : dyn
Path LSP ID   : 12302
Path Admin    : Up
Path Oper     : Down
Out Interface : n/a
Out Label     : n/a
---snip---
```

```
Failure Code  : noCspfRouteToDestination
Failure Node  : 192.0.2.4
```

```

Explicit Hops      :
  No Hops Specified
Actual Hops       :
  No Hops Specified
Computed Hops     :
  No Hops Specified
Resignal Eligible: False
Last Resignal    : n/a                CSPF Metric      : N/A
=====
* indicates that the corresponding row element may have been truncated.

```

The port is enabled again and the LSP configuration is restored to FRR facility, as follows:

```

# on PE-4: configure {
  port 1/1/c1/2 {
    admin-state enable
  }
}
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-4-PE-2" {
        admin-state disable
        fast-reroute {
          frr-method facility
        }
      }
    }
  }
}
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-4-PE-2" {
        admin-state enable
      }
    }
  }
}

```

FRR bypass not possible on iLER on unnumbered interfaces

FRR facility is not supported on the iLER PE-4 if the bypass is over an unnumbered interface. This restriction only applies to the iLER, not to the LSRs. The interface toward PE-3 is reconfigured as unnumbered, as follows:

```

# on PE-3: configure {
  router "Base" {
    interface "int-PE-3-PE-4" {
      ipv4 {
        delete primary
        unnumbered {
          system
        }
      }
    }
  }
  mpls {
    interface "int-PE-3-PE-4" {
      te-metric 10
    }
  }
}
exit all

```

```

# on PE-4: configure {
  router "Base" {

```

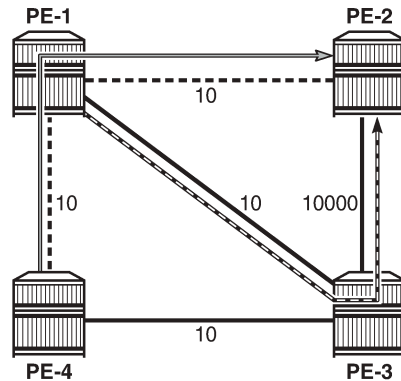
```

interface "int-PE-4-PE-3" {
  ipv4 {
    delete primary
    unnumbered {
      system
    }
  }
}
mpls {
  interface "int-PE-4-PE-3" {
    te-metric 10
  }
}
exit all

```

When an interface changes from numbered to unnumbered or the other way around, it is no longer known in the **mpls** context. Therefore, the interface must be added in the **mpls** context again. When the interface toward PE-3 is numbered, there is a bypass tunnel in PE-4 to protect node PE-1, but this bypass tunnel cannot be established on an unnumbered interface. The only remaining protection for the LSP is the bypass tunnel originating in PE-1 to protect the link between PE-1 and PE-2, as shown in [Figure 318: FRR on iLER: no bypass on unnumbered interfaces](#).

Figure 318: FRR on iLER: no bypass on unnumbered interfaces



Legend:
 — LSP-PE-4-PE-2 Primary path
 - - - Bypass link between PE-1 and PE-2

25688

The following output shows that there is only a detour available in PE-1:

```

[/]
A:admin@PE-4# show router mpls lsp "LSP-PE-4-PE-2" path detail

=====
---snip---
-----
LSP LSP-PE-4-PE-2
Path dyn
-----
LSP Name      : LSP-PE-4-PE-2
From          : 192.0.2.4
To           : 192.0.2.2
Admin State   : Up
Oper State    : Up

```

```

Path Name      : dyn
Path LSP ID    : 12308
Path Admin     : Up
Out Interface  : 1/1/cl/1
Path Type      : Primary
Path Oper      : Up
Out Label      : 524287
---snip---
Explicit Hops  :
  No Hops Specified
Actual Hops    :
  192.0.2.4, If Index : 2
-> 192.0.2.1, If Index : 4 @
-> 192.0.2.2, If Index : 2
Computed Hops  :
  192.0.2.4, If Index : 2(S)
-> 192.0.2.1, If Index : 4(S)
-> 192.0.2.2, If Index : 2(S)
Resignal Eligible: False
Last Resignal  : n/a
CSPF Metric    : 20
=====
* indicates that the corresponding row element may have been truncated.

```

In iLER PE-4, there is only the LSP tunnel, no bypass tunnel, as follows:

```

[/]
A:admin@PE-4# show router rsvp session

=====
RSVP Sessions
=====
RSVP Session Name
  From           To           Tunnel ID   LSP ID     State
-----
LSP-PE-4-PE-2::dyn
192.0.2.4       192.0.2.2   1           12308      Up
-----
Sessions : 1
=====

```

The original configuration is restored with numbered interfaces between PE-3 and PE-4, as follows:

```

# on PE-3: configure {
  router "Base" {
    interface "int-PE-3-PE-4" {
      ipv4
        delete unnumbered
        primary {
          address 192.168.34.1
          prefix-length 30
        }
      }
    }
  mpls {
    interface "int-PE-3-PE-4" {
      te-metric 10
    }
  }
}
exit all

```

```

# on PE-4: configure {
  router "Base" {
    interface "int-PE-4-PE-3" {
      ipv4

```



```
        delete unnumbered
        primary {
            address 192.168.34.2
            prefix-length 30
        }
    }
}
mpls {
    interface "int-PE-4-PE-3" {
        te-metric 10
    }
}
exit all
```

Admin groups for unnumbered interfaces in RSVP

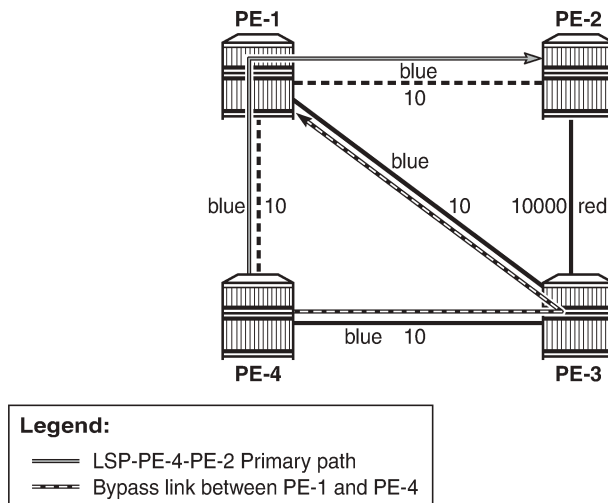
Administrative groups (link coloring) can be used to calculate a path with the restriction to only include, or exclude, links of a particular admin group (color). Paths can be disjointed from each other, without the need for an explicit hops list. For unnumbered interfaces, an explicit hops list is not an option, but admin groups are.

Two admin groups are configured on all nodes, as follows:

```
# on all nodes: configure {
    routing-options {
        if-attribute {
            admin-group "red" {
                value 0
            }
            admin-group "blue" {
                value 1
            }
        }
    }
}
exit all
```

Admin group "blue" is assigned to all MPLS interfaces, except for the link between PE-2 and PE-3; see [Figure 319: FRR facility and admin groups](#).

Figure 319: FRR facility and admin groups



25689

The admin groups are assigned to the interfaces in the **mpls** context, as follows:

```
# on PE-2: configure {
  router "Base" {
    mpls {
      resignal-timer 30
      interface "int-PE-2-PE-1" {
        admin-group ["blue"]
      }
      interface "int-PE-2-PE-3" {
        admin-group ["red"]
      }
    }
  }
  exit all
```

To ensure that FRR bypass tunnels only use links belonging to the same admin group, the following is configured on all nodes. It is required on all PLRs.

```
# on all nodes: configure {
  router "Base" {
    mpls {
      admin-group-frr true
    }
  }
  exit all
```

In the **lsp** context, the admin group "blue" is included. The option **propagate-admin-group** implies that the tunnels must use links belonging to the admin group "blue". This is configured for the LSP tunnel, and for the FRR bypass tunnels, as follows:

```
# on PE-4: configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-4-PE-2" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.2
        propagate-admin-group true
        path-computation-method local-cspf
      }
    }
  }
  exit all
```

```
metric-type te
  include-admin-group ["blue"]
  fast-reroute {
    propagate-admin-group true
    frr-method facility
  }
  primary "dyn" {
  }
}
exit all
```

This configuration implies that the link that does not belong to admin group "blue" is excluded, and cannot be used by the LSP nor by a bypass tunnel. Therefore, there is no bypass tunnel to protect node PE-1 and no bypass tunnel originating in PE-1 protecting the link to PE-2. There is a bypass tunnel originating in PE-4 to protect the link between PE-4 and PE-1, as shown in [Figure 319: FRR facility and admin groups](#). The following output shows that a detour is available for link protection in PE-4:

```
[/]
A:admin@PE-4# show router mpls lsp "LSP-PE-4-PE-2" path detail

=====
MPLS LSP LSP-PE-4-PE-2 Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use
  b - Bandwidth Protected      n - Node Protected
  s - Soft Preemption
  S - Strict                    L - Loose
  A - ABR                       + - Inherited
=====
-----
LSP LSP-PE-4-PE-2
Path dyn
-----
LSP Name      : LSP-PE-4-PE-2
From          : 192.0.2.4
To           : 192.0.2.2
Admin State   : Up
Oper State    : Up
Path Name     : dyn
Path LSP ID   : 12310
Path Admin    : Up
Path Type     : Primary
Path Oper     : Up
Out Interface : 1/1/c1/1
Out Label     : 524286
---snip---
Include Groups :
blue
Oper IncludeGroups:
blue
Exclude Groups :
None
Oper ExcludeGroups:
None
---snip---
Explicit Hops :
No Hops Specified
Actual Hops :
  192.0.2.4, If Index : 2 @
  -> 192.0.2.1, If Index : 4
  -> 192.0.2.2, If Index : 2
Record Label : N/A
Record Label : 524286
Record Label : 524285
Computed Hops :
  192.0.2.4, If Index : 2(S)
  -> 192.0.2.1, If Index : 4(S)
  -> 192.0.2.2, If Index : 2(S)
---snip---
=====
* indicates that the corresponding row element may have been truncated.
```

The following output shows two RSVP sessions in PE-4: one for the LSP and one for the bypass tunnel to protect the link between PE-4 and PE-1.

```
[/]
A:admin@PE-4# show router rsvp session

=====
RSVP Sessions
=====
RSVP Session Name
  From              To              Tunnel ID   LSP ID   State
-----
LSP-PE-4-PE-2::dyn
192.0.2.4          192.0.2.2      1           12310    Up
bypass-link192.0.2.1-61457
192.0.2.4          192.168.13.1  61457       22       Up
-----
Sessions : 2
=====
```

The configuration is restored as follows:

```
# on PE-4: configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-4-PE-2" {
        fast-reroute {
          delete propagate-admin-group
        }
        delete propagate-admin-group
        delete include-admin-group ["blue"]
      }
      delete admin-group-frr
    }
  }
  exit all
```

SRLGs for unnumbered interfaces in RSVP

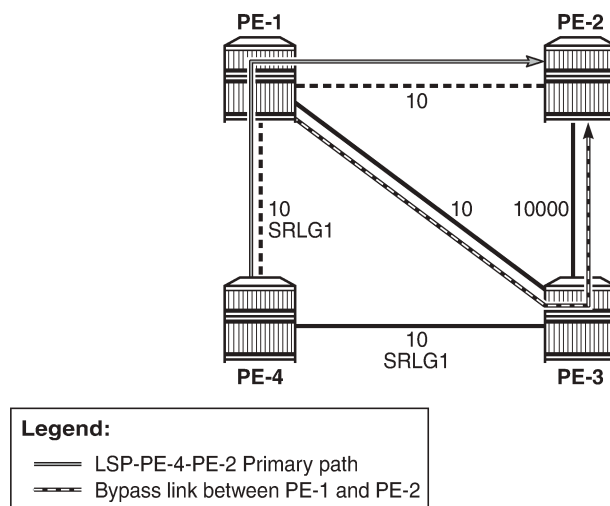
SRLGs allow operators to create automatic secondary LSPs or FRR tunnels that are disjointed from the protected primary tunnel. See chapter [Shared Risk Link Groups for RSVP-Based LSPs](#) for more information.

One SRLG group is configured on all nodes, as follows:

```
# on all nodes: configure {
  routing-options {
    if-attribute {
      srlg-group "SRLG1" {
        value 1
      }
    }
  }
  exit all
```

SRLG "SRLG1" is assigned to the interface between PE-4 and PE-1, and to the interface between PE-4 and PE-3, as shown in [Figure 320: SRLG-FRR strict: no bypass on PE-4](#).

Figure 320: SRLG-FRR strict: no bypass on PE-4



25690

The SRLG is assigned to the interfaces in the **mpls** context, as follows:

```
# on PE-4: configure {
  router "Base" {
    mpls {
      interface "int-PE-4-PE-1" {
        srlg-group "SRLG1" { }
      }
      interface "int-PE-4-PE-3" {
        srlg-group "SRLG1" { }
      }
    }
  }
  exit all
```

The configuration on PE-1 and PE-3 is similar.

When SRLG for FRR is enabled in strict mode, CSPF does not establish any detour LSP if there is no path that meets the SRLG constraint. This configuration implies that there is no bypass tunnel in PE-4. The following enables SRLG for FRR in strict mode on all nodes:

```
# on all nodes: configure {
  router "Base" {
    mpls {
      srlg-frr strict
    }
  }
  exit all
```



Note:

Enabling or disabling SRLG for FRR is a system-wide configuration that requires the MPLS routing instance to be manually disabled and then re-enabled, to activate the change. This can be service affecting. Nokia recommends that the operator include the SRLG in the initial network design and implementation to minimize the traffic loss.

The following output shows that there is only a detour available in PE-1:

```
[/]
A:admin@PE-4# show router mpls lsp "LSP-PE-4-PE-2" path detail
```

```

=====
MPLS LSP LSP-PE-4-PE-2 Path (Detail)
=====
Legend :
  @ - Detour Available          # - Detour In Use
  b - Bandwidth Protected      n - Node Protected
  s - Soft Preemption
  S - Strict                    L - Loose
  A - ABR                      + - Inherited
=====
-----
LSP LSP-PE-4-PE-2
Path dyn
-----
LSP Name      : LSP-PE-4-PE-2
From          : 192.0.2.4
To            : 192.0.2.2
Admin State   : Up                Oper State      : Up
Path Name     : dyn
Path LSP ID   : 12314             Path Type       : Primary
Path Admin    : Up                Path Oper       : Up
Out Interface : 1/1/c1/1          Out Label       : 524287
---snip---
Explicit Hops :
  No Hops Specified
Actual Hops   :
  192.0.2.4, If Index : 2          Record Label    : N/A
-> 192.0.2.1, If Index : 4 @      Record Label    : 524287
-> 192.0.2.2, If Index : 2          Record Label    : 524287
Computed Hops :
  192.0.2.4, If Index : 2(S)
-> 192.0.2.1, If Index : 4(S)
-> 192.0.2.2, If Index : 2(S)
---snip---
=====
* indicates that the corresponding row element may have been truncated.

```

The following output shows that PE-1 has two RSVP sessions: one for the LSP and one for the bypass tunnel to protect the link between PE-1 and PE-2.

```

[/]
A:admin@PE-1# show router rsvp session

=====
RSVP Sessions
=====
RSVP Session Name
  From          To          Tunnel ID  LSP ID    State
-----
LSP-PE-4-PE-2::dyn
192.0.2.4      192.0.2.2      1          12314     Up

bypass-link192.0.2.2-61496
192.0.2.1      192.168.23.1   61496     10        Up

-----
Sessions : 2
=====

```

This was the last example for unnumbered interfaces in RSVP. MPLS and RSVP are disabled in all nodes as follows:

```
# on all nodes: configure {
  router "Base" {
    rsvp {
      admin-state disable
    }
    mpls {
      admin-state disable
      delete srlg-frr
    }
  }
  exit all
```

Unnumbered interfaces in LDP

Link LDP is configured on PE-4, as follows:

```
# on PE-4: configure {
  router "Base" {
    ldp {
      interface-parameters {
        interface "int-PE-4-PE-1" {
          ipv4 {
            }
          }
        interface "int-PE-4-PE-3" {
          ipv4 {
            }
          }
        }
      }
    }
  exit all
```

The configuration of link LDP on the other nodes is similar. Link LDP sessions are established, which can be verified as follows:

```
[/]
A:admin@PE-4# show router ldp session family ipv4

=====
LDP IPv4 Sessions
=====
Peer LDP Id          Adj Type  State           Msg Sent  Msg Recv  Up Time
-----
192.0.2.1:0          Link      Established     25        26        0d 00:00:41
192.0.2.3:0          Link      Established     79        80        0d 00:02:49
-----
No. of IPv4 Sessions: 2
=====
```

The **Peer LDP Id** is the LSR ID, which is the system address, by default. The IP address configured on the unnumbered interface (such as 192.1.2.1) is not used. The following tunnel table shows a distinction between numbered and unnumbered interfaces in the next hop:

```
[/]
A:admin@PE-4# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
```

```

=====
Destination      Owner    Encap TunnelId Pref  Nexthop      Metric
  Color
-----
192.0.2.1/32    ldp     MPLS  65540   9    int-PE-4-PE-1 10
192.0.2.2/32    ldp     MPLS  65541   9    int-PE-4-PE-1 20
192.0.2.3/32    ldp     MPLS  65537   9    192.168.34.1  10
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====

```

For destination 192.0.2.1 or 192.0.2.2, the unnumbered interface toward PE-1 is taken. The next hop is represented by int-PE-4-PE-1. When a node has several unnumbered interfaces, the corresponding next hop values are different, as follows, for PE-1:

```

[/]
A:admin@PE-1# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner    Encap TunnelId Pref  Nexthop      Metric
  Color
-----
192.0.2.2/32    ldp     MPLS  65541   9    int-PE-1-PE-2 10
192.0.2.3/32    ldp     MPLS  65542   9    192.168.13.2  10
192.0.2.4/32    ldp     MPLS  65540   9    int-PE-1-PE-4 10
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====

```

The LDP active prefix bindings only contain system addresses, no other loopback prefixes, as follows:

```

[/]
A:admin@PE-4# show router ldp bindings active prefixes ipv4

=====
LDP Bindings (IPv4 LSR ID 192.0.2.4)
(IPv6 LSR ID ::)
=====
---snip---
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix          Op
IngLbl          EgrLbl
EgrNextHop      EgrIf/LspId
-----
192.0.2.1/32    Push
--              524283
Unnumbered      1/1/c1/1

192.0.2.1/32    Swap
524283          524283
Unnumbered      1/1/c1/1

```



```

192.0.2.2/32          Push
--                  524280
Unnumbered           1/1/c1/1

192.0.2.2/32          Swap
524282              524280
Unnumbered           1/1/c1/1

192.0.2.3/32          Push
--                  524287
192.168.34.1         1/1/c1/2

192.0.2.3/32          Swap
524286              524287
192.168.34.1         1/1/c1/2

192.0.2.4/32          Pop
524287              --
--                  --

-----
No. of IPv4 Prefix Active Bindings: 7
=====

```

For prefixes 192.0.2.1 and 192.0.2.2, the egress next hop is unnumbered. The egress interface for both is 1/1/c1/1. There is no If Index. Local addresses are advertised in LDP address messages, such as:

```

# on PE-4:
8 2023/03/21 13:24:43.317 UTC MINOR: DEBUG #2001 Base LDP
"LDP: LDP
Send Address packet (msgId 5) to 192.0.2.1:0
Protocol version = 1
Address Family = 1 Number of addresses = 3
Address 1 = 192.0.2.4
Address 2 = 192.1.2.4
Address 3 = 192.168.34.2
"

9 2023/03/21 13:24:43.378 UTC MINOR: DEBUG #2001 Base LDP
"LDP: LDP
Recv Address packet (msgId 5) from 192.0.2.1:0
Protocol version = 1
Address Family = 1 Number of addresses = 3
Address 1 = 192.0.2.1
Address 2 = 192.1.2.1
Address 3 = 192.168.13.1
"

```

The list of advertised local addresses includes the loopback addresses "loopback1": 192.1.2.1 and 192.1.2.4. These loopback addresses did not occur in the preceding lists of LDP sessions or LDP bindings, but they occur in the LDP session local addresses, as follows:

```

[/]
A:admin@PE-4# show router ldp session local-addresses ipv4

=====
LDP Session Local-Addresses
=====
-----
Session with Peer 192.0.2.1:0,
      Local 192.0.2.4:0
-----

```

```
IPv4 Sent Addresses:
                192.0.2.4      192.1.2.4      192.168.34.2
IPv4 Recv Addresses:
                192.0.2.1      192.1.2.1      192.168.13.1
-----
---snip---
```

If there were only unnumbered addresses and no additional loopback addresses, only the system address and other loopback addresses would be sent or received. The interface addresses in the list of local addresses are from numbered interfaces.

Configuring the local LSR ID

To use the loopback address "loopback1" in the LDP sessions, the local LSR ID is configured as follows:

```
# on PE-4: configure {
  router "Base" {
    ldp {
      interface-parameters {
        interface "int-PE-4-PE-1" {
          ipv4 {
            local-lsr-id {
              interface-name "loopback1"
            }
          }
          transport-address interface
        }
      }
    }
  }
  exit all
```

The transport address is the system address, by default, but here it is changed to the address of "loopback1", which is 192.1.2.4. The configuration is similar on PE-1. On PE-2, the system addresses are kept and no additional configuration is required.

LDP hello messages are sent from the transport address to 224.0.0.2 to establish hello adjacencies. The transport address is 192.1.2.4 for the unnumbered interface toward PE-1, and 192.0.2.4 (system address of PE-4) for the numbered interface toward PE-3. LDP hello adjacencies are verified as follows:

```
[/]
A:admin@PE-4# show router ldp discovery ipv4

=====
LDP IPv4 Hello Adjacencies
=====
Interface Name      Local Addr      State
AdjType             Peer Addr
-----
int-PE-4-PE-1      192.1.2.4:0    Estab
link                192.1.2.1:0
int-PE-4-PE-3      192.0.2.4:0    Estab
link                192.0.2.3:0

-----
No. of IPv4 Hello Adjacencies: 2
=====
```

The LDP hello adjacencies are established, but the LDP session on the unnumbered interface is non-existent, as follows:

```
[/]
A:admin@PE-4# show router ldp session family ipv4

=====
LDP IPv4 Sessions
=====
Peer LDP Id      Adj Type  State      Msg Sent  Msg Recv  Up Time
-----
192.0.2.3:0     Link     Established 119       120       0d 00:04:32
192.1.2.1:0    Link    Nonexistent 10        11        0d 00:00:36
-----
No. of IPv4 Sessions: 2
=====
```

The LDP session is non-existent because the prefix 192.1.2.1/32 is not in the routing table and the LDP session is to be established between 192.1.2.4 and 192.1.2.1. The following export policy is configured and added in the IS-IS context on the nodes:

```
# on PE-4: configure {
  policy-options {
    policy-statement "export_ISIS" {
      entry 10 {
        from {
          protocol {
            name [direct]
          }
        }
        action {
          action-type accept
        }
      }
      default-action {
        action-type reject
      }
    }
  }
  router "Base" {
    isis 0 {
      export-policy ["export_ISIS"]
    }
  }
}
exit all
```

The loopback addresses are now exported in IS-IS. When the loopback addresses are in the routing table, the LDP session is established, as follows:

```
[/]
A:admin@PE-4# show router ldp session family ipv4

=====
LDP IPv4 Sessions
=====
Peer LDP Id      Adj Type  State      Msg Sent  Msg Recv  Up Time
-----
192.0.2.3:0     Link     Established 130       128       0d 00:04:57
192.1.2.1:0    Link    Established 29        30        0d 00:01:01
-----
No. of IPv4 Sessions: 2
=====
```

The local LSR ID can also be configured for targeted LDP sessions, as follows:

```
# on PE-4: configure {
  router "Base" {
    ldp {
      targeted-session {
        peer 192.1.2.1 {
          local-lsr-id {
            interface-name "loopback1"
          }
        }
      }
    }
  }
}
```

The configuration on PE-1 is similar for peer 192.0.2.4. On PE-4, the LDP adjacency type is now both link and targeted for peer 192.1.2.1, as follows:

```
[/]
A:admin@PE-4# show router ldp session family ipv4

=====
LDP IPv4 Sessions
=====
Peer LDP Id      Adj Type  State           Msg Sent  Msg Recv  Up Time
-----
192.0.2.3:0     Link     Established    144       144       0d 00:05:36
192.1.2.1:0    Both    Established  45        52        0d 00:01:40
-----
No. of IPv4 Sessions: 2
=====
```

Even though the LDP sessions are established, there is no LDP prefix binding for the loopback address, as follows:

```
[/]
A:admin@PE-4# show router ldp bindings active prefixes ipv4

=====
LDP Bindings (IPv4 LSR ID 192.0.2.4)
(IPv6 LSR ID ::)
=====
---snip---
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix          Op
IngLbl          EgrLbl
EgrNextHop      EgrIf/LspId
-----
192.0.2.1/32    Push
--             524283
Unnumbered     1/1/c1/1

192.0.2.1/32    Swap
524283         524283
Unnumbered     1/1/c1/1

192.0.2.2/32    Push
--             524280
Unnumbered     1/1/c1/1

192.0.2.2/32    Swap
```

```

524282                               524280
Unnumbered                           1/1/c1/1

192.0.2.3/32                          Push
--                                    524287
192.168.34.1                          1/1/c1/2

192.0.2.3/32                          Swap
524286                                524287
192.168.34.1                          1/1/c1/2

192.0.2.4/32                          Pop
524287                                --
--                                    --

-----
No. of IPv4 Prefix Active Bindings: 7
=====

```

There is no label mapping for prefix 192.1.2.1/32.

SDPs are created toward all other nodes, as follows:

```

# on PE-4: configure {
  service {
    sdp 411 {
      admin-state enable
      delivery-type mpls
      ldp true
      far-end {
        ip-address 192.0.2.1
      }
    }
    sdp 412 {
      admin-state enable
      delivery-type mpls
      ldp true
      far-end {
        ip-address 192.1.2.1
      }
    }
    sdp 421 {
      admin-state enable
      delivery-type mpls
      ldp true
      far-end {
        ip-address 192.0.2.2
      }
    }
    sdp 431 {
      admin-state enable
      delivery-type mpls
      ldp true
      far-end {
        ip-address 192.0.2.3
      }
    }
  }
exit all

```

The following output shows that, between PE-4 and PE-1, there are two LDP SDPs: one using the system address and another using the loopback address 192.1.2.1. The configuration on the other nodes is

similar. All SDPs that have a system address as the far end are operationally up, whereas the SDP toward 192.1.2.1 is down:

```
[/]
A:admin@PE-4# show service sdp

=====
Services: Service Destination Points
=====
SdpId  AdmMTU  OprMTU  Far End          Adm  Opr      Del  LSP  Sig
-----
411    0       1556   192.0.2.1        Up  Up       MPLS L    TLDP
412    0       0      192.1.2.1        Up  Down     MPLS L    TLDP
421    0       1556   192.0.2.2        Up  Up       MPLS L    TLDP
431    0       1556   192.0.2.3        Up  Up       MPLS L    TLDP
-----
Number of SDPs : 4
-----
Legend: R = RSVP, L = LDP, B = BGP, M = MPLS-TP, n/a = Not Applicable
        I = SR-ISIS, 0 = SR-OSPF, T = SR-TE, F = FPE
=====
```

The SDP toward 192.1.2.1 is down because the transport tunnel is down, as follows:

```
[/]
A:admin@PE-4# show service sdp 412 detail

=====
Service Destination Point (Sdp Id : 412) Details
=====
-----
Sdp Id 412 -192.1.2.1
-----
Description          : (Not Specified)
SDP Id               : 412                SDP Source           : manual
Admin Path MTU      : 0                Oper Path MTU        : 0
Delivery             : MPLS
Far End              : 192.1.2.1          Tunnel Far End       :
Oper Tunnel Far End  : 192.1.2.1
LSP Types            : LDP

Admin State          : Up                Oper State            : Down
Signaling            : TLDP                Metric                : 0
---snip---
Flags                : TranspTunnDown
---snip---
=====
```

The solution is to manually add an LDP prefix binding, as described in the following section.

Configuring the FEC originate

The labels to be used for a manually created LDP prefix binding must be chosen from the range for static labels: from 32 to 18431. This range can be retrieved as follows:

```
[/]
A:admin@PE-1# show router mpls-labels label-range

=====
```

Label Ranges					
Label Type	Start Label	End Label	Aging	Available	Total
Static	32	18431	-	18400	18400
Dynamic	18432	524287	0	505852	505856
Seg-Route	0	0	-	0	0

To manually add an LDP prefix binding for the loopback prefixes, configure the following:

```
# on PE-4: configure {
  router "Base" {
    ldp {
      fec-originate 192.1.2.1/32 {
        next-hop 192.0.2.4
        interface "int-PE-4-PE-1"
        swap-label 101
      }
      fec-originate 192.1.2.4/32 {
        advertised-label 104
        pop true
      }
    }
  }
  exit all
```

```
# on PE-1: configure {
  router "Base" {
    ldp {
      fec-originate 192.1.2.1/32 {
        advertised-label 101
        pop true
      }
      fec-originate 192.1.2.4/32 {
        next-hop 192.0.2.1
        interface "int-PE-1-PE-4"
        swap-label 104
      }
    }
  }
  exit all
```

This configuration for unnumbered interfaces includes the interface name, such as int-PE-4-PE-1. This parameter is optional for numbered interfaces.

If the label is chosen from the dynamic range instead of the static range, an error is raised for the pop operation, as follows:

```
# on PE-1: configure {
  router "Base" {
    ldp {
      fec-originate 192.1.2.1/32 {
        pop true
        advertised-label 100001
      }
    }
  }
}
MINOR: MGMT_CORE #4001: configure router "Base" ldp fec-originate 192.1.2.1/32 advertised-label
- advertised-label 100001 is not within the valid range for static labels
- configure router "Base" mpls-labels static-label-range
```

For interoperability, no error is raised for the swap operation.

As a result, three active LDP bindings are added: one pop operation for the local loopback prefix, and a swap and a push operation for the remote loopback prefix, as follows:

```
[/]
```

```
A:admin@PE-4# show router ldp bindings active prefixes ipv4
=====
LDP Bindings (IPv4 LSR ID 192.0.2.4)
(IPv6 LSR ID ::)
=====
---snip---
=====
LDP IPv4 Prefix Bindings (Active)
=====
Prefix                               Op
IngLbl                               EgrLbl
EgrNextHop                           EgrIf/LspId
-----
192.0.2.1/32                         Push
--                                  524283
Unnumbered                           1/1/c1/1

192.0.2.1/32                         Swap
524283                               524283
Unnumbered                           1/1/c1/1

192.0.2.2/32                         Push
--                                  524280
Unnumbered                           1/1/c1/1

192.0.2.2/32                         Swap
524282                               524280
Unnumbered                           1/1/c1/1

192.0.2.3/32                         Push
--                                  524287
192.168.34.1                         1/1/c1/2

192.0.2.3/32                         Swap
524286                               524287
192.168.34.1                         1/1/c1/2

192.0.2.4/32                         Pop
524287                               --
--                                  --

192.1.2.1/32                       Push
--                                  101
Unnumbered                        1/1/c1/1

192.1.2.1/32                       Swap
524285                             101
Unnumbered                        1/1/c1/1

192.1.2.4/32(S)                   Pop
104                                 --
--                                  --

-----
No. of IPv4 Prefix Active Bindings: 10
=====
```

The following output shows that the SDPs are all operationally up, including the one toward the loopback address:

```
[/]
A:admin@PE-4# show service sdp
```



```

=====
Services: Service Destination Points
=====
SdpId  AdmMTU  OprMTU  Far End          Adm  Opr          Del  LSP  Sig
-----
411    0        1556    192.0.2.1        Up  Up           MPLS L    TLDP
412    0        1556    192.1.2.1        Up  Up           MPLS L    TLDP
421    0        1556    192.0.2.2        Up  Up           MPLS L    TLDP
431    0        1556    192.0.2.3        Up  Up           MPLS L    TLDP
-----
Number of SDPs : 4
-----
Legend: R = RSVP, L = LDP, B = BGP, M = MPLS-TP, n/a = Not Applicable
        I = SR-ISIS, 0 = SR-OSPF, T = SR-TE, F = FPE
=====

```

LDP FRR Loop-Free Alternate on unnumbered interfaces

LDP FRR Loop-Free Alternate (LFA) is supported on unnumbered interfaces and on numbered interfaces. For information about LDP FRR LFA, see chapter [MPLS LDP FRR using ISIS as IGP](#). LDP FRR LFA can be configured as follows:

```

# on all nodes: configure {
  router "Base" {
    isis 0 {
      loopfree-alternate {
      }
    }
    ldp {
      fast-reroute {
      }
    }
  }
  routing-options {
    ip-fast-reroute true
  }
}
exit all

```

Enabling FRR LFA is a local decision. In this configuration, it is configured on all nodes. The LFA coverage can be retrieved as follows:

```

[/]
A:admin@PE-4# show router isis lfa-coverage

=====
Rtr Base ISIS Instance 0 LFA Coverage
=====
Topology          Level  Node          IPv4          IPv6
-----
IPv4 Unicast    L1     3/3(100%)    7/7(100%)    0/0(0%)
IPv6 Unicast      L1     0/0(0%)      0/0(0%)      0/0(0%)
IPv4 Multicast    L1     0/0(0%)      0/0(0%)      0/0(0%)
IPv6 Multicast    L1     0/0(0%)      0/0(0%)      0/0(0%)
IPv4 Unicast      L2     0/0(0%)      7/7(100%)    0/0(0%)
IPv6 Unicast      L2     0/0(0%)      0/0(0%)      0/0(0%)
IPv4 Multicast    L2     0/0(0%)      0/0(0%)      0/0(0%)
IPv6 Multicast    L2     0/0(0%)      0/0(0%)      0/0(0%)
=====

```

There is protection for the three other nodes and for all remote prefixes in the routing table, which can be verified as follows:

```
[/]
A:admin@PE-4# show router route-table alternative

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age           Pref
  Next Hop[Interface Name]        Metric
  Alt-NextHop                      Alt-
                                   Metric
-----
192.0.2.1/32                       Remote ISIS   00h43m13s  15
  int-PE-4-PE-1                    10
  192.168.34.1 (LFA)                20
192.0.2.2/32                       Remote ISIS   00h43m13s  15
  int-PE-4-PE-1                    20
  192.168.34.1 (LFA)                30
192.0.2.3/32                       Remote ISIS   00h22m23s  15
  192.168.34.1                    10
  int-PE-4-PE-1 (LFA)              20
192.0.2.4/32                       Local  Local   01h08m36s  0
  system                            0
192.1.2.1/32                       Remote ISIS   00h06m40s  15
  int-PE-4-PE-1                    10
  192.168.34.1 (LFA)                20
192.1.2.2/32                       Remote ISIS   00h06m39s  15
  int-PE-4-PE-1                    20
  192.168.34.1 (LFA)                30
192.1.2.4/32                       Local  Local   00h43m14s  0
  loopback1                         0
192.168.13.0/30                   Remote ISIS   00h06m40s  160
  int-PE-4-PE-1                    10
  192.168.34.1 (LFA)                20
192.168.23.0/30                   Remote ISIS   00h22m23s  15
  192.168.34.1                    10010
  int-PE-4-PE-1 (LFA)              10020
192.168.34.0/30                   Local  Local   00h22m24s  0
  int-PE-4-PE-3                    0
-----
No. of Routes: 10
Flags: n = Number of times nexthop is repeated
       Backup = BGP backup route
       LFA = Loop-Free Alternate nexthop
       S = Sticky ECMP requested
=====
```

For unnumbered interfaces, the interface name is shown (int-PE-4-PE-1); for numbered interfaces, the next hop IP address is shown (192.168.34.1). The LFA type is link protection for the three nodes, as follows:

```
[/]
A:admin@PE-4# show router isis topology lfa detail

=====
Rtr Base ISIS Instance 0 Topology Table
=====
-----
IS-IS IP paths (MT-ID 0), Level 1
-----
Node      : PE-1.00
```

```

Nexthop   : PE-1
Interface : int-PE-4-PE-1
SNPA      : none                               Metric    : 10

LFA nh    : PE-3
LFA intf  : int-PE-4-PE-3                     LFA Metric : 20
LFA type  : linkProtection

Node      : PE-2.00
Nexthop   : PE-1
Interface : int-PE-4-PE-1
SNPA      : none                               Metric    : 20

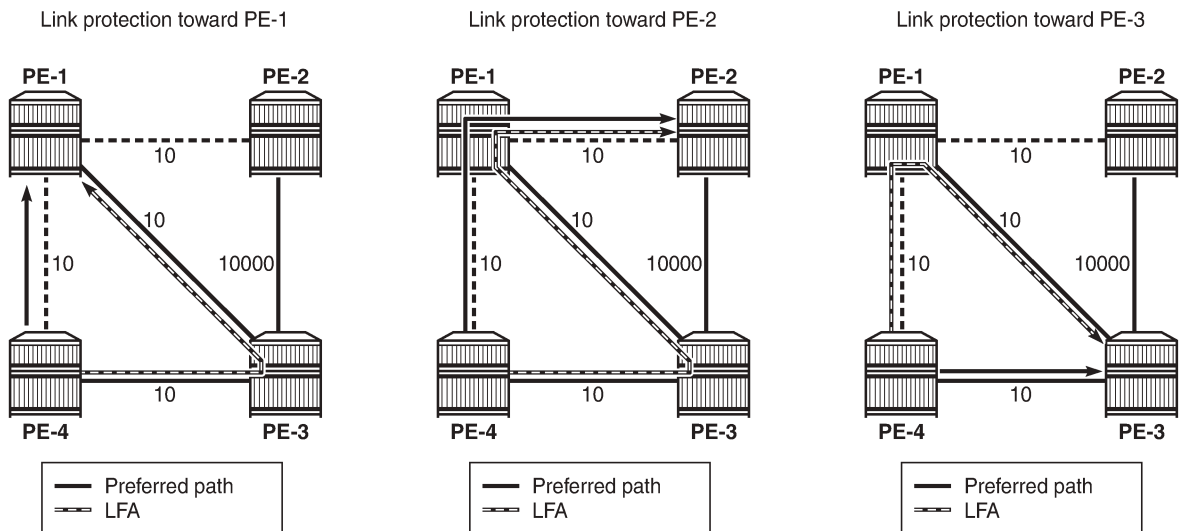
LFA nh    : PE-3
LFA intf  : int-PE-4-PE-3                     LFA Metric : 30
LFA type  : linkProtection

Node      : PE-3.00
Nexthop   : PE-3
Interface : int-PE-4-PE-3
SNPA      : none                               Metric    : 10

LFA nh    : PE-1
LFA intf  : int-PE-4-PE-1                     LFA Metric : 20
LFA type  : linkProtection
=====
    
```

The LFA protection is shown in [Figure 321: LDP FRR LFA link protection on PE-4](#).

Figure 321: LDP FRR LFA link protection on PE-4



25691

The LDP bindings for FRR LFA indicate alternate (**BU**) in the list, as follows:

```

[/]
A:admin@PE-4# show router ldp bindings prefixes ipv4
=====
    
```

```

LDP Bindings (IPv4 LSR ID 192.0.2.4)
(IPv6 LSR ID ::)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch,
  BA - ASBR Backup FEC
=====
LDP IPv4 Prefix Bindings
=====
Prefix
Peer          FEC-Flags
IgrLbl        EgrLbl
EgrNextHop    EgrIntf/LspId
-----
192.0.2.1/32
192.0.2.3:0
524283U          524283BU
192.168.34.1    1/1/c1/2

192.0.2.1/32
192.1.2.1:0
--              524283
Unnumbered      1/1/c1/1

192.0.2.2/32
192.0.2.3:0
524282U          524285BU
192.168.34.1    1/1/c1/2

192.0.2.2/32
192.1.2.1:0
524282N          524280
Unnumbered      1/1/c1/1

192.0.2.3/32
192.0.2.3:0
--              524287
192.168.34.1    1/1/c1/2

192.0.2.3/32
192.1.2.1:0
524286U          524281BU
Unnumbered      1/1/c1/1

192.0.2.4/32
192.0.2.3:0
524287U          --
--              --

192.0.2.4/32
192.1.2.1:0
524287U          --
--              --

192.1.2.1/32
192.0.2.3:0
524285U          524286BU
192.168.34.1    1/1/c1/2

192.1.2.1/32

```

```
192.1.2.1:0
--
Unnumbered
192.1.2.4/32
192.0.2.3:0
104U
--
192.1.2.4/32
192.1.2.1:0
104U
--
-----
No. of IPv4 Prefix Bindings: 12
=====
```

Conclusion

Unnumbered interfaces were initially supported for SONET/SDH/ATM/FR, and later also on Ethernet access ports. IS-IS adjacencies and OSPF neighbors can be established on unnumbered interfaces. This chapter shows that unnumbered interfaces can be added to RSVP or LDP. Most features that are supported on numbered interfaces are also supported on unnumbered interfaces.

Segment Routing and PCE

This section provides segment routing and PCE configuration information for the following topics:

- [BGP Segment Routing Using the Prefix SID Attribute](#)
- [BGP Signaled Segment Routing Policy](#)
- [Inter-AS Model C VPRN Using MPLS Forwarding Policies and Segment Routing Policies](#)
- [Parallel Adjacency Sets in Segment Routing](#)
- [Seamless BFD for SR-TE LSPs](#)
- [Segment Routing – Traffic Engineered Tunnels](#)
- [Segment Routing over IPv6](#)
- [Segment Routing over IPv6 for VPRN](#)
- [Segment Routing with IS-IS Control Plane](#)
- [SR-TE LSP Path Computation Using Local CSPF](#)
- [SRv6 Encapsulation in the Base Routing Instance](#)
- [SRv6 Loop-Free Alternate](#)

BGP Segment Routing Using the Prefix SID Attribute

This chapter describes BGP Segment Routing using the prefix SID attribute.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information and configuration in this chapter are based on SR OS Release 23.3.R1. BGP Segment Routing (SR) is supported in SR OS Release 19.10.R1, and later.

Overview

Segment Routing (SR) has become a foundational technology for Software-Defined Networking (SDN) in Wide Area Networks (WANs). Also, SR is being extended beyond WAN borders into Data Centers (DCs).

SR allows an ingress node to route a packet from the source, by prepending an SR header containing an ordered list of segment identifiers (SIDs). A SID represents a topological or service-based instruction. A SID can have a local meaning for one specific node, or a global meaning within the SR domain, such as the instruction to forward a packet on the Equal-Cost Multipath (ECMP) aware shortest path to reach some prefix.

In WAN networks, infrastructure IP reachability is nearly always conveyed by an IGP protocol, such as OSPF and IS-IS, but in large-scale DCs, BGP has become the protocol of choice. In a typical DC design, BGP is used for endpoint reachability, as follows:

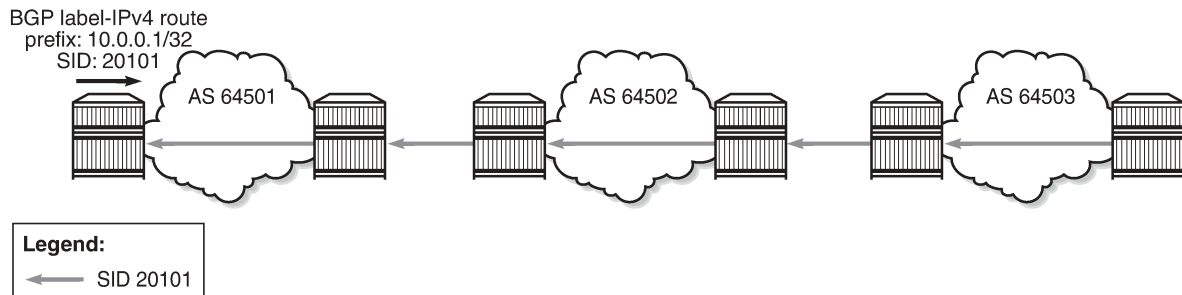
- Each node (Top of Rack (TOR), leaf, spine, and so on) has its own Autonomous System (AS).
- Each node has an eBGP session to each of its directly connected peers.
- Each node originates the IPv4 (or IPv6) address of its loopback interface into BGP and announces it to its neighbors.

To extend SR-MPLS into DCs that use this type of BGP design, the SR OS nodes must advertise their loopback IP prefix in a BGP labeled-unicast (BGP-LU) IPv4 route with a prefix SID attribute. The prefix SID attribute is ignored when attached to other types of BGP routes, including BGP-LU IPv6 routes, but it is still be propagated.

A BGP prefix SID is always a global SID within the SR domain and identifies an instruction to forward the packet along the ECMP-aware BGP-computed best paths to reach the prefix. The BGP prefix SID attribute can also help to create SR paths that transit across multiple administrative domains that do not share IGP SR topology information.

Figure 322: BGP-LU IPv4 route with prefix SID BGP path attribute shows a node in AS 64501 advertising a BGP-LU IPv4 route for prefix 10.0.0.1/32 with SID 20101. The SR-capable nodes forward packets with SID 20101 via the best BGP path to 10.0.0.1, using any of the available multipaths computed by BGP.

Figure 322: BGP-LU IPv4 route with prefix SID BGP path attribute



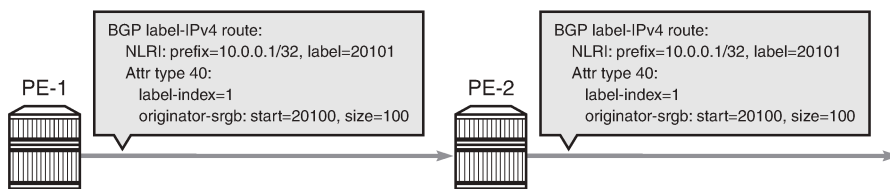
35886

The BGP prefix SID attribute with type code 40 is an optional and transitive BGP path attribute, meaning that the attribute is expected to be propagated by routers that do not recognize the type value. When SR is deployed using an MPLS dataplane (SR-MPLS), the BGP prefix SID encodes:

- A 32-bit label-index Type-Length-Value (TLV) (mandatory TLV)
- An originator Segment Routing Global Block (SRGB) TLV containing one or more SRGB fields (optional TLV). If the SRGB field occurs multiple times in the SRGB TLV, the SRGB space of the ingress node consists of multiple ranges that are concatenated.

Figure 323: BGP signaling overview shows that node PE-1 exports a BGP-LU IPv4 route with prefix 10.0.0.1/32 and label 20101. The BGP prefix SID attribute is attribute type 40 and contains an SR label index of 1 and the originator SRGB with start label 20100 and size 100 (from 20100 to 20199). Node PE-2 imports the BGP-LU IPv4 route and exports it to the next node.

Figure 323: BGP signaling overview



35887

To add, replace, or process a BGP prefix SID, SR must be administratively enabled in the **bgp** context. The BGP prefix SID range can be set to either **global** (that is, equal to the SRGB also used by SR-OSPF or SR-ISIS and defined in the **router "Base" mpls-labels sr-labels** context) or a subset of the SRGB defined by the **start-label** command in combination with **max-index**. All BGP prefix SID values must reside within the global SRGB or the **start-label** command fails. The **prefix-sid-range** is a mandatory requirement.

To originate BGP SR prefixes, two policies are required with an **sr-label-index** action, which may or may not be identical:

- **route-table-import policy-name <policy-name>** used to populate a local BGP-SR table with an SR label index
- **export policy [<policy>]** to advertise a prefix to a neighbor with an SR label index

In the example topology used in this chapter, the import and export policies are identical and have an **action** entry with **action-type accept** with **sr-label-index** with **value 1**, so on PE-1, the prefix SID for the

prefix 10.0.0.1/32 equals 20101, which is the sum of the start label for the prefix SID range 20100 and the SR label index 1.

A unique label index value must be assigned to each different IPv4 prefix that is advertised with a BGP prefix SID. However, in case of a conflict with another SR-programmed Label Forwarding Instance Base (LFIB) entry, the conflict situation is addressed as follows:

- If the conflict is with another BGP-LU IPv4 route for a different prefix with a prefix SID attribute, all the conflicting BGP-LU IPv4 routes for both prefixes are advertised with normal BGP-LU labels from the dynamic label range, not from the dedicated SR label range.
- If the conflict is with an IGP route and the route-table-import policy action does not contain the **prefer-igp** in the **sr-label-index** command, the BGP-LU IPv4 route loses to the IGP route and is advertised with a normal BGP-LU label from the dynamic SR label range.
- If the conflict is with an IGP route and the route-table-import policy action contains the **prefer-igp** in the **sr-label-index** command, this is not considered a conflict and BGP uses the IGP-signaled label index to derive its advertised label. This stitches the BGP SR tunnel to the IGP SR tunnel.

Stitching of SR-ISIS or SR-OSPF to SR-BGP is one of the main advantages of implementing SR-BGP.

Any /32 BGP-LU IPv4 route containing a prefix SID attribute is resolvable and usable in the same way as /32 BGP-LU IPv4 routes without prefix SID attribute. The routes can be installed in the route table and tunnel table, have ECMP next hops or FRR backup next hops, and can be used as transport tunnels.

Receiving a /32 BGP-LU IPv4 route with prefix SID attribute does not create a tunnel in the SR database; it only creates a label swap entry when the route is re-advertised with a new next hop. This means that the first SID in any SID list of an SR policy should not be based on a BGP prefix SID because the data path would not be programmed correctly. However, the BGP prefix SID can be used as a non-first SID in any SR policy.

Each node capable of receiving and propagating the BGP prefix SID attribute can be configured with the **block-prefix-sid** command at the BGP global, group, or neighbor configuration levels to:

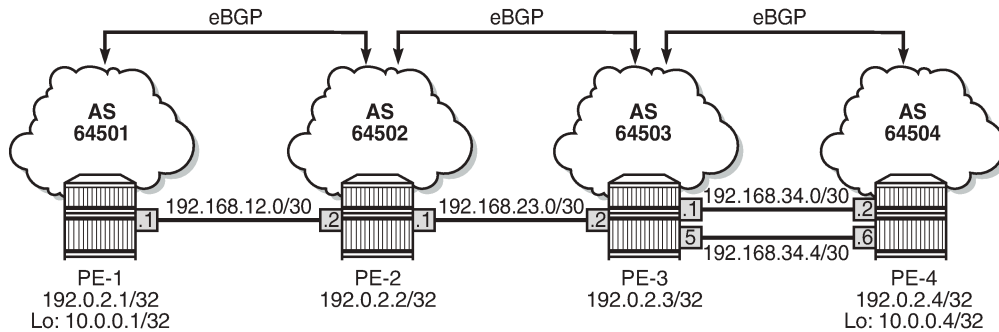
- block the propagation of the attribute outside its local SR domain
- block inbound propagation of the attribute from another SR domain

When **block-prefix-sid** applies to a BGP session, the prefix SID attribute is stripped from all sent and received routes on that session, even if the prefix SID attribute was added to the outbound routes by the local router. By default, this feature is not configured, so the prefix SID is propagated freely to and from all BGP peers.

Configuration

[Figure 324: Example topology](#) shows the example topology with four nodes in different ASs. The loopback addresses 10.0.0.1/32 on PE-1 and 10.0.0.4/32 on PE-4 are exported in BGP-LU IPv4 routes with prefix SID attribute.

Figure 324: Example topology



35888

The initial configuration includes:

- Cards, MDAs, ports
- Router interfaces
- eBGP sessions for the label-IPv4 address family
- PE-3 and PE-4 have **ecmp** and **multipath max-paths** set to 2 for BGP address family **label-ipv4**

No IGP is configured, so SR-OSPF or SR-ISIS cannot be used.

Configure BGP segment routing using prefix SID

BGP SR is enabled on all PEs. Also, the SRGB is configured and the BGP SR labels are defined as a subset of the SRGB, as follows:

```
# on PE-1, PE-2, PE-3, PE-4:
configure exclusive
  router "Base" {
    mpls-labels {
      sr-labels {
        start 20000
        end 20999
      }
    }
    bgp {
      segment-routing {
        admin-state enable
        prefix-sid-range {
          start-label 20100
          max-index 99
        }
      }
    }
  }
}
```

It is possible to define different policies with the **sr-label-index** action for importing and exporting the prefixes, but in this example, the same policy is used. The following policy is used for exporting and importing prefix 10.0.0.1/32 on PE-1:

```
# on PE-1:
configure exclusive
  policy-options {
    prefix-list "10.0.0.1/32" {
      prefix 10.0.0.1/32 type exact {
      }
    }
    policy-statement "prefix-sid-1" {
      entry 10 {
        from {
          prefix-list ["10.0.0.1/32"]
        }
        action {
          action-type accept
          sr-label-index {
            value 1
          }
        }
      }
    }
  }
}
```

Likewise, PE-4 exports prefix 10.0.0.4/32 with SR label index value 4, resulting in a BGP prefix SID 20104 (start label 20100 + index 4 = 20104).

The **route-table-import policy-name** command is used to populate a local BGP-SR table with SR label 20101 (20100 + 1 = 20101), as follows:

```
# on PE-1:
configure exclusive
  router "Base" {
    bgp {
      rib-management {
        label-ipv4 {
          route-table-import {
            policy-name "prefix-sid-1"
          }
        }
      }
    }
  }
}
```

The export policy is configured in the BGP group, as follows:

```
# on PE-1:
configure exclusive
  router "Base" {
    bgp {
      group "eBGP" {
        family {
          label-ipv4 true
        }
      }
      neighbor "192.168.12.2" {
        group "eBGP"
        peer-as 64502
      }
      export {
```


Prefix	Advertised Label	Received Label	Flags
10.0.0.1/32	20101	20101	E
10.0.0.4/32	20104	-	-

Total Labels allocated: 2			
=====			

Because PE-3 and PE-4 have ECMP and BGP multipath configured, traffic flows can be sprayed over two links. The E-flag in the last column indicates that an ECMP next-hop is available for prefix 10.0.0.4/32 on PE-3 and for prefix 10.0.0.1 on PE-4.

The tunnel table on PE-1 shows that a tunnel with ID 262145 is available toward destination 10.0.0.4/32:

```
[/]
A:admin@PE-1# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
10.0.0.4/32      bgp        MPLS  262145   12   192.168.12.2  1000
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
```

The FP-tunnel table provides more information about the label (20104) and next hop (192.168.12.2):

```
[/]
A:admin@PE-1# show router fp-tunnel-table 1

=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination      Protocol      Tunnel-ID
  Lbl/SID
  NextHop
  Lbl/SID (backup)
  NextHop (backup)
-----
10.0.0.4/32      BGP          -
  20104
  192.168.12.2      1/1/c1/1:100
-----
Total Entries : 1
=====
```

On PE-2, two tunnels are available: one toward destination 10.0.0.1/32 with SR label 20101 and another toward destination 10.0.0.4/32 with SR label 20104:

```
[/]
A:admin@PE-2# show router fp-tunnel-table 1

=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol      Tunnel-ID
Lbl/SID
NextHop                                     Intf/Tunnel
Lbl/SID (backup)
NextHop   (backup)
-----
10.0.0.1/32                                BGP           -
20101
192.168.12.1                               1/1/c1/2:100
10.0.0.4/32                                BGP           -
20104
192.168.23.2                               1/1/c1/1:100
-----
Total Entries : 2
=====
```

On PE-3, three tunnels are available: one toward destination 10.0.0.1/32 with SR label 20101 and two toward destination 10.0.0.4/32 with SR label 20104.

```
[/]
A:admin@PE-3# show router fp-tunnel-table 1

=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol      Tunnel-ID
Lbl/SID
NextHop                                     Intf/Tunnel
Lbl/SID (backup)
NextHop   (backup)
-----
10.0.0.1/32                                BGP           -
20101
192.168.23.1                               1/1/c1/2:100
10.0.0.4/32                                BGP           -
20104
192.168.34.2                               1/1/c1/1:100
20104
192.168.34.6                               1/1/c1/3:100
-----
Total Entries : 2
=====
```

On PE-4, two tunnels are available toward destination 10.0.0.1/32 with SR label 20101:

```
[/]
A:admin@PE-4# show router fp-tunnel-table 1

=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                               Protocol      Tunnel-ID
  Lbl/SID
  NextHop                                  Intf/Tunnel
  Lbl/SID (backup)
  NextHop (backup)
-----
10.0.0.1/32                               BGP          -
  20101
  192.168.34.1                             1/1/c1/2:100
  20101
  192.168.34.5                             1/1/c1/3:100
-----
Total Entries : 1
=====
```

PE-1 advertised a BGP-LU IPv4 route for prefix 10.0.0.1/32 with label 20101 to PE-2. The following command on PE-2 shows the received route:

```
[/]
A:admin@PE-2# show router bgp routes 10.0.0.1/32 label-ipv4

=====
BGP Router ID:192.0.2.2      AS:64502      Local AS:64502
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
BGP LABEL-IPV4 Routes
=====
Flag Network                               LocalPref  MED
  Nexthop (Router)                          Path-Id    IGP Cost
  As-Path                                    Label

-----
u*>i 10.0.0.1/32                            None       None
      192.168.12.1                          None       0
      64501                                   None       20101
-----
Routes : 1
=====
```

This route is advertised to PE-3 and finally to PE-4. The following command on PE-4 shows two BGP-LU IPv4 routes for prefix 10.0.0.1/32 with label 20101: one with next hop 192.168.34.1 and another one with next hop 192.168.34.5.

```
[/]
A:admin@PE-4# show router bgp routes 10.0.0.1/32 label-ipv4
```

```

=====
BGP Router ID:192.0.2.4      AS:64504      Local AS:64504
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                  l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP LABEL-IPV4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  10.0.0.1/32                            None       None
      192.168.34.1                          None       0
      64503 64502 64501                      None       20101
u*>i  10.0.0.1/32                            None       None
      192.168.34.5                          None       0
      64503 64502 64501                      None       20101
-----
Routes : 2
=====

```

The detailed output for the BGP-LU IPv4 routes on PE-4 show the prefix SID attribute with index 1 and originator SRGB with start label 20100 and size 100, as follows:

```

[/]
A:admin@PE-4# show router bgp routes 10.0.0.1/32 label-ipv4 detail
=====
BGP Router ID:192.0.2.4      AS:64504      Local AS:64504
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                  l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP LABEL-IPV4 Routes
=====
Original Attributes

Network       : 10.0.0.1/32
Nexthop      : 192.168.34.1
Path Id      : None
From         : 192.168.34.1
Res. Nexthop : 192.168.34.1
Local Pref.  : n/a
Aggregator AS : None
Atomic Aggr. : Not Atomic
AIGP Metric  : None
Connector    : None
Community    : No Community Members
Cluster      : No Cluster Members
Originator Id : None
Fwd Class    : None
IPv4 Label   : 20101
Flags        : Used Valid Best IGP In-TTM In-RTM
Route Source : External
AS-Path      : 64503 64502 64501
Route Tag    : 0
Neighbor-AS  : 64503
Interface Name : int-PE-4-PE-3
Aggregator     : None
MED            : None
IGP Cost       : 0
Peer Router Id : 192.0.2.3
Priority        : None

```



```

DB Orig Val   : NotFound           Final Orig Val : N/A
Source Class  : 0                  Dest Class     : 0
Add Paths Send : Default
RIB Priority   : Normal
Last Modified  : 00h01m18s
Prefix SID    : index 1, originator-srgb [20100/100]
---snip---
-----
Routes : 2
=====
=====

```

The following debug message shows how the prefix SID attribute is advertised in a BGP update:

```

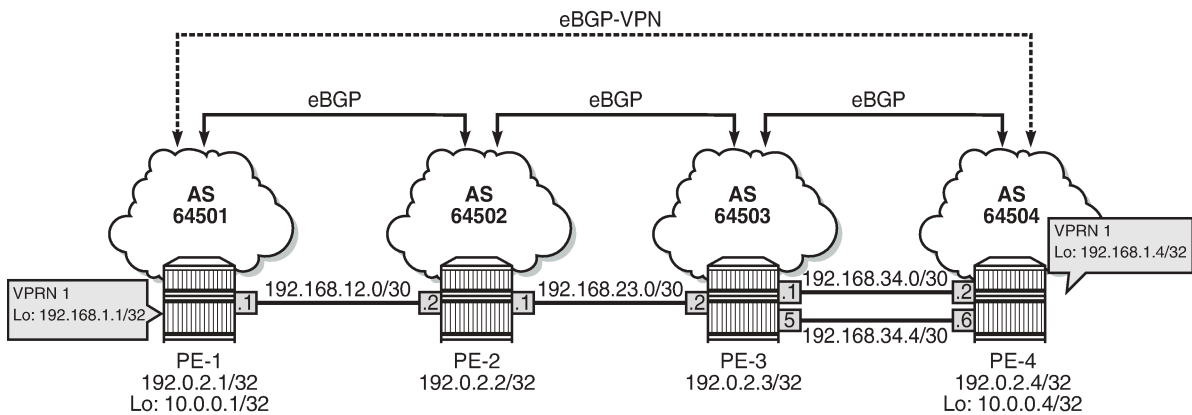
18 2023/04/17 18:16:28.069 CEST MINOR: DEBUG #2001 Base Peer 1: 192.168.34.1
"Peer 1: 192.168.34.1: UPDATE
Peer 1: 192.168.34.1 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 66
  Flag: 0x90 Type: 14 Len: 17 Multiprotocol Reachable NLRI:
    Address Family LBL-IPV4
    NextHop len 4 NextHop 192.168.34.1
    10.0.0.1/32 Label 20101
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 14 AS Path:
    Type: 2 Len: 3 < 64503 64502 64501 >
  Flag: 0xc0 Type: 40 Len: 21 Prefix-SID-attr:
    Label Index TLV (10 bytes):-
      flags: 0x0 label Index: 1
    Originator SRGB TLV (11 bytes):-
      flags: 0x0 start_label: 20100 num_label: 100
"

```

Configure VPRN

Figure 325: Example topology with VPRN 1 shows the example topology with a basic VPRN service to demonstrate the end-to-end control plane signaling and data plane verification.

Figure 325: Example topology with VPRN 1



35890

A BGP multi-hop session for address family VPN-IPv4 is configured between the GRT loopback addresses 10.0.0.1/32 on PE-1 and 10.0.0.4/32 on PE-4. On PE-1, the additional BGP configuration is as follows:

```
# on PE-1:
configure exclusive
router "Base" {
  bgp {
    group "eBGP-VPN" {
      family {
        vpn-ipv4 true
      }
    }
    neighbor "10.0.0.4" {
      group "eBGP-VPN"
      multihop 64
      local-address 10.0.0.1
      peer-as 64504
    }
  }
}
```

In addition, the VPRN 1 service has loopback addresses 192.168.1.1/32 on PE-1 and 192.168.1.4/32 on PE-4. The configuration on PE-1 is as follows:

```
# on PE-1:
configure exclusive
service {
  vprn "VPRN 1" {
    admin-state enable
    service-id 1
    customer "1"
    bgp-ipvpn {
      mpls {
        admin-state enable
        route-distinguisher "1:1"
        vrf-target {
          community "target:1:1"
        }
        auto-bind-tunnel {
          resolution any
        }
      }
    }
    interface "lol" {
      loopback true
      ipv4 {
        primary {
          address 192.168.1.1
          prefix-length 32
        }
      }
    }
  }
}
```

The configuration on PE-4 is similar.

The following VPN-IPv4 route is received on PE-1:

```
[/]
A:admin@PE-1# show router bgp routes vpn-ipv4
=====
BGP Router ID:192.0.2.1      AS:64501      Local AS:64501
=====
Legend -
```

```
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
```

```
=====
BGP VPN-IPv4 Routes
=====
```

Flag	Network	LocalPref	MED
	Nexthop (Router)	Path-Id	IGP Cost
	As-Path		Label
u*>i	4:1:192.168.1.4/32	None	None
	10.0.0.4	None	0
	64504		524286

```
-----
Routes : 1
=====
```

The route table for VPRN 1 on PE-1 is as follows:

```
[/]
A:admin@PE-1# show router 1 route-table

=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]                Type  Proto  Age           Pref
Next Hop[Interface Name]          Metric
-----
192.168.1.1/32                    Local Local  00h01m31s    0
    lo1                             0
192.168.1.4/32                    Remote BGP VPN 00h01m14s    170
    10.0.0.4 (tunneled:BGP)         1000
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```

Conclusion

With BGP SR, it is possible to use SR without the use of an IGP protocol (for example, to cross AS boundaries). It is also possible to stitch SR-IGP and SR-BGP tunnels together. BGP SR uses the prefix SID attribute.

BGP Signaled Segment Routing Policy

This chapter describes BGP Signaled Segment Routing Policy.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information and configuration contained in this chapter are based on MD-CLI of SR OS Release 21.7.R1.

Overview

Segment Routing (SR) allows a head-end node to steer a packet flow along a source-routed path. SR policy is a generic framework that describes the procedures and processes that a head-end node carries out when instantiating such a path. The SR policy consists of an ordered list of segments on a node, sufficient to implement a traffic-engineered path. The segments can have any type of Segment Identifier (SID), including Adjacency-SIDs, Node-SIDs, and Anycast-SIDs. The head-end can then steer traffic, using the SR policy as appropriate.

An SR policy can define one or multiple candidate paths. When explicit candidate paths are used, each path contains one or more segment lists, where each segment list contains the ordered set of segments (identified by their unique SID) required to provide the source-routed path from head-end to destination. When a candidate path contains multiple segment lists, each is assigned a weight for the purpose of weighted load-balancing. Candidate paths can be instantiated using a variety of ways, including Path Computation Element Protocol (PCEP), BGP, or local configuration. This chapter describes the use of BGP to advertise SR policy candidate paths. The term "BGP SR policy" is interchangeably used with "BGP SR TE policy".

SR policy overview

An SR policy is identified through the tuple {head-end, color, endpoint}.

- The head-end is the node where the SR policy is instantiated, and the node that is responsible for steering traffic, using the SR policy with the relevant SID stack. From the perspective of the head-end, the SR policy can be identified using the {color, endpoint} tuple.
- The color is a fundamental part of the SR policy and forms part of the Network Layer Reachability Information (NLRI). The color is a 32-bit numerical value that a head-end uses to associate the SR policy with a characteristic, such as low-latency or high-throughput.
- The endpoint is the destination in the SR policy specified as an IPv4 or IPv6 address, although "wildcard" destinations can be used and are described later in this chapter.

Color is also a 32-bit transitive extended community originally defined in *draft-ietf-idr-tunnel-encaps* that can be attached to a BGP update message, in order to associate it with a corresponding SR policy. For example, if head-end H learns a BGP route R with {next-hop N, color extended community C, and VPN label V} and head-end H has a valid SR policy P to {endpoint N, color C}, it can associate BGP route R with the SR policy P. When H receives packets with a destination matching BGP route R, it forwards them using the instructions contained within SR policy P.

SR policy NLRI

The BGP address family "SR TE policy" (SAFI 73) is defined to advertise a candidate path for an SR policy in BGP and is carried in an update message using BGP multiprotocol extensions. The AFI must be IPv4 (AFI=1) or IPv6 (AFI=2). An SR policy candidate path may be advertised from a centralized controller, or it may be advertised by a router; for example, an egress router advertising paths to itself. [Figure 326: SR TE policy NLRI](#) shows the structure of the SR TE policy NLRI.

Figure 326: SR TE policy NLRI

Path Attribute:	MP_REACH_NLRI <Distinguisher, Policy Color, Endpoint>	
Path Attribute:	Origin, AS_PATH, Local_PREF, and so on	
Path Attribute:	Tunnel_Encapsulation_Attribute: Tunnel Type SR Policy	
	Sub-TLV:	Binding SID
	Sub-TLV:	Preference
	Sub-TLV:	Segment List
	Sub-TLV:	Segment
	Sub-TLV:	Segment

36648

The SR TE policy NLRI is used to identify an SR policy candidate path and, because it uses MP-BGP, it is carried in an MP_REACH/UNREACH_NLRI path attribute. The NLRI contains the color and endpoint values described previously, and a distinguisher. The distinguisher is an integer value in the range 1 to 4294967295 that serves to make the SR policy unique from an NLRI perspective. The SR TE policy NLRI uses standard BGP propagation and best-path selection; a unique distinguisher ensures that best-path selection does not unnecessarily suppress SR policy advertisements.

Multiple candidate paths can exist for an SR policy, although only one path can be selected as the best path of the SR policy and become the active path. If several candidate paths of the same SR policy (endpoint, color) are advertised via BGP SR TE policy to the same head-end, unique distinguishers for each NLRI are recommended. In SR OS Release 21.7.R1, only a single candidate path is supported for an SR policy.

The other parameters of the SR policy candidate path are carried as sub-TLVs of the Tunnel Encapsulation Attribute (*draft-ietf-idr-tunnel-encaps*) using a tunnel-type known as "SR policy", and are described following.

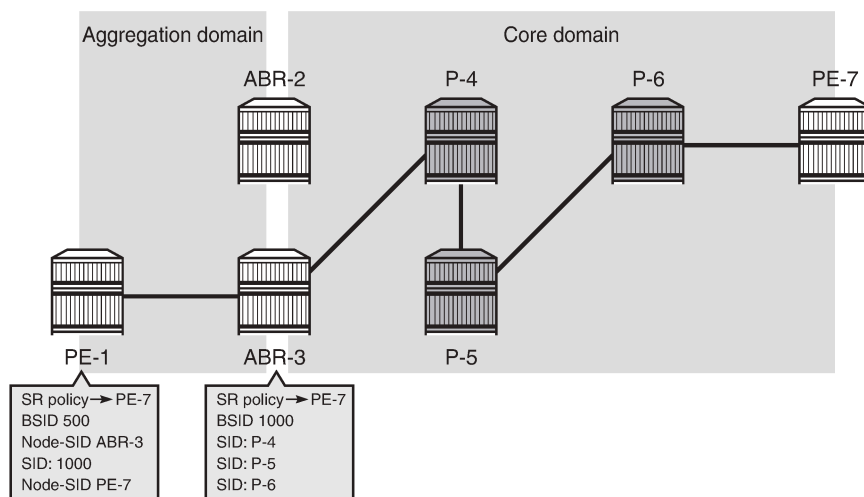
Binding SID

The SR architecture defines the use of a Binding SID (BSID). A BSID is bound to an SR policy, and packets arriving at a node with an active label equal to the BSID are steered using that SR policy. This

action may mean swapping the incoming active label with one or more outgoing labels representing the SR policy path.

When used in this manner, the Binding SID serves as an anchor point, sometimes referred to as a "BSID anchor", that allows one domain to be isolated from another domain. This is shown in [Figure 327: Binding SID \(BSID\) anchor](#), where ABR-3 is acting as a BSID anchor between the aggregation domain and the core domain. ABR-3 has an SR policy to PE-7 with the path P-4-P-5-P-6 and with a BSID of 1000. The PE-1 resulting SR policy to PE-7 consists of the path {Node-SID ABR-3, 1000, Node-SID PE-7} and a BSID of 500. When a packet is forwarded by the SR policy on PE-1 and arrives at ABR-3, it pops the Node-SID ABR-3 label, and swaps label 1000 for the label stack {P-4, P-5, P-6} of the SR policy on ABR-3.

Figure 327: Binding SID (BSID) anchor



36649

The BSID serves as an anchor point, which allows one domain to be isolated from the churn of another domain. If something changes in the path P-4-P-5-P-6, ABR-3 can repair the path locally without needing to change the BSID value known at PE-1. PE-1 is therefore protected from the churn in the core domain. The BSID also serves to reduce the number of segments/labels that the head-end needs to impose an end-to-end traffic-engineered path.

Segment list

A segment list sub-TLV encodes a single path toward the endpoint. Multiple segment list sub-TLVs may be included in each SR policy. Each segment list sub-TLV may contain multiple segment sub-TLVs and may carry a weight sub-TLV. Each segment sub-TLV describes a single segment in a segment list, and multiple segments may be concatenated to constitute an end-to-end path of the SR policy.

There are several types of the segment sub-TLV, allowing for the segment to be expressed as a variant of IPv4/IPv6 node address or local/remote address, and with a SID in the form of an MPLS label or IPv6 address. This chapter focuses only on the Type A encoding, which is represented as a SID in the form of an MPLS label. The SID contained within each segment sub-TLV can be any form of SID, including Node-SID, Adjacency-SID, Anycast-SID, or Binding SID.

The optional weight sub-TLV is used to implement (weighted) load-balancing in the presence of multiple segment lists. By default, SR OS assigns a weight value of 1 to each segment list.

Preference

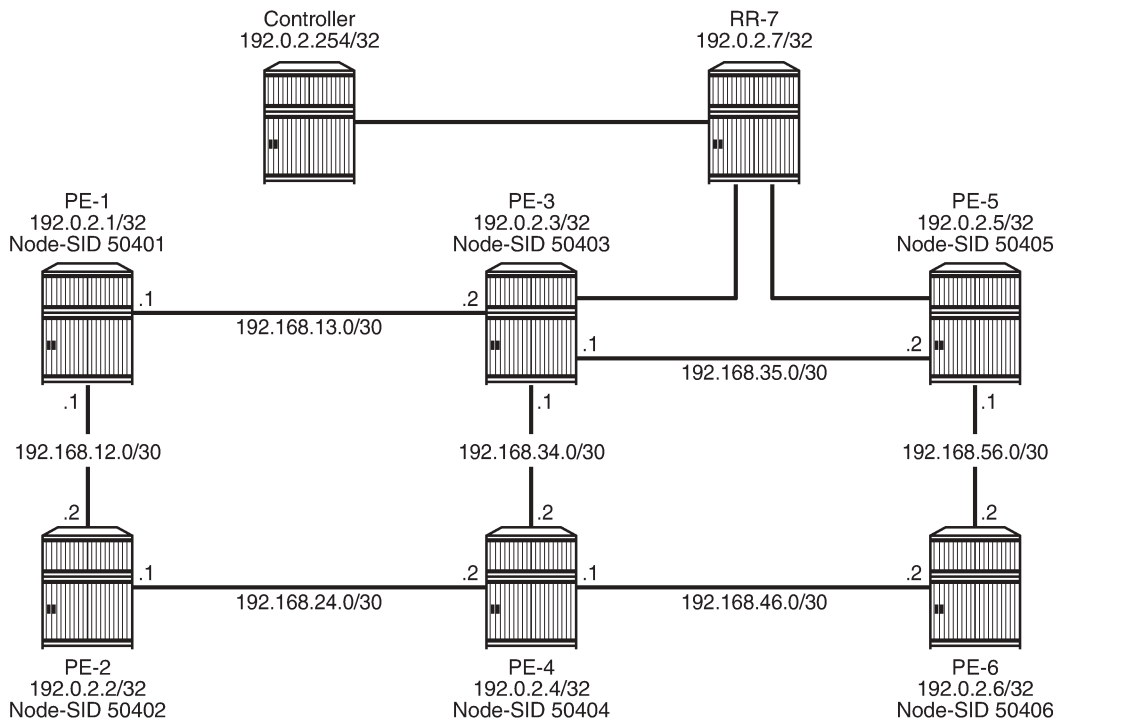
The preference sub-TLV is used to indicate the preference of a candidate path in relation to other candidate paths. Multiple candidate paths can exist in an SR policy, but only one candidate path can be selected as the best and active path. When multiple candidate paths exist that are considered valid, the candidate path with the highest preference is selected. The default value of the preference is 100. If multiple paths have the same preference, the protocol origin (PCEP, BGP, local configuration) may be considered, followed by the lower value of originator, followed by the higher value of discriminator.

Example topology

The topology in [Figure 328: Example topology](#) shows the use of BGP SR TE policy within this chapter. All PE routers within the example topology and the Route Reflector (RR-7) form part of Autonomous System 64496 and belong to the same IS-IS Level-2 area. All IGP link metrics are 100 and are symmetric. SR is enabled within the domain, and the associated Node-SIDs are shown in [Figure 328: Example topology](#) (Adj-SIDs are not shown for the purpose of clarity). The SRGB in use is {50000-54999}. All PE routers are clients of the Route Reflector for multiple address families including SR TE policy.

The example topology also has an additional router simulating a controller, which uses static routing for IP connectivity. This is the point from which SR policies are advertised into BGP, although as previously described, SR policies can be advertised into BGP by a controller or a router. The controller peers in the SR TE policy address family with the Route Reflector, which in turn reflects those routes to its clients.

Figure 328: Example topology



36650

Configuration

An SR policy can be statically (CLI) configured locally on a head-end or dynamically learned by a head-end through BGP SR TE policy route. For SR OS to obtain an SR TE policy route, that route needs to be configured locally as a static SR policy. This chapter provides an example of the instantiation of an SR policy using static configuration on the head-end, but thereafter focuses on the instantiation of SR policies learned by it through BGP SR TE policy. The same static SR policy configuration is used regardless of whether it is for advertising that SR policy into BGP to the head-end, or applying it at that local head-end to forward traffic.

Segment Routing Local Block

A BSID may be either a local SID or a global SID. In general, and for the use-cases in this chapter, BSIDs are local SIDs, so a BSID needs to be within the range of a locally-configured Segment Routing Local Block (SRLB). SRLBs are reserved label blocks used for specific local purposes, such as SR policy BSIDs, Adjacency Set SIDs, and static Adjacency SIDs. A dedicated SRLB is required per application and has only local significance, so the same values can be used on all SR routers in the domain. Ranges for each SRLB are taken from the dynamic label range. The following configuration allocates labels 100000 to 109999 to the SRLB "SRLB-BSID":

```
# on all nodes:
configure {
  router "Base" {
    mpls-labels {
      reserved-label-block "SRLB-BSID" {
        start-label 100000
        end-label 109999
      }
    }
  }
}
```

After the SRLB is defined, it is dedicated to the specific application, which in this case is SR policies. When the **reserved-label-block** is assigned, **sr-policies** must be enabled (**admin-state enable**), as follows:

```
# on all nodes:
configure {
  router "Base" {
    segment-routing {
      sr-policies {
        admin-state enable
        reserved-label-block "SRLB-BSID"
      }
    }
  }
}
```

The preceding configuration is applied to all SR routers in the domain.

Static SR policy

As previously described, SR policies can be statically (CLI) configured locally on a head-end or dynamically learned by a head-end through BGP SR TE policy route. In this section, the necessary steps are shown for the instantiation of an SR policy using static configuration locally on PE-1 as the head-end.

The following output shows the configuration of a static SR policy at PE-1 (192.0.2.1) with an endpoint of PE-5 (192.0.2.5).

```
# on PE-1:
configure {
  router "Base" {
    segment-routing {
      sr-policies {
        static-policy "PE-1-PE-5-color600" {
          admin-state enable      # enable static SR policy
          color 600
          endpoint 192.0.2.5
          head-end local
          binding-sid 100002
          distinguisher 600001005
          segment-list 1 {
            admin-state enable    # enable segment list
            segment 1 {
              mpls-label 50402    # node-SID PE-2
            }
            segment 2 {
              mpls-label 150024   # adj-SID int-PE-2-PE-4
            }
            segment 3 {
              mpls-label 150046   # adj-SID int-PE-4-PE-6
            }
            segment 4 {
              mpls-label 50405    # node-SID PE-5
            }
          }
        }
      }
    }
  }
}
```

The static SR policy is initially created within the `sr-policies` context and begins by assigning a **binding-sid** of 100002. In this example, the SR policy is local to PE-1, and the BSID value is therefore within the range of the PE-1 SRLB. If this static SR policy were to be advertised into BGP, the advertised BSID value must be in the range of the SRLB configured on the target head-end.

The next three parameters are the color, distinguisher, and endpoint that constitute the SR policy NLRI. The SR policy **color** is 600, and is a 4-octet value that can be configured in the range 1 to 4294967295. The **distinguisher** is also a 4-octet value with the same range and is configured as 600001005 (representing the color plus the last octet of the head-end and endpoint addresses). As previously described, the purpose of the distinguisher is to make the SR policy unique from an NLRI perspective, such that if multiple candidate paths of the same SR policy (endpoint, color) are advertised, they are not suppressed by any BGP best-path selection algorithm.

The **endpoint** is the IPv4 or IPv6 address of the destination for the SR policy and is configured as the PE-5 address 192.0.2.5. There are special circumstances where the value 0.0.0.0 or 0::0 is allowed as an endpoint. This is referred to as color-only steering and is described later in this chapter.

The **head-end** is the target node where the SR policy is to be instantiated. If the SR policy is statically configured on the head-end for forwarding of traffic locally using that SR policy, the value **local** is used, as shown in this example. If the SR policy is configured somewhere other than on the head-end, and advertised into BGP toward the head-end, the value of the head-end parameter is the IPv4 address of that head-end. When the SR policy is advertised into BGP, the head-end address is also encoded as an IPv4 address-specific Route-Target Extended Community, which allows for potential constraining of route propagation.

The final parameter is the segment list. The preceding configuration output shows the segment list consisting of four segments, which represent the path using the following SIDs:

- Segment 1 SID is 50402, which is the Node-SID of PE-2
- Segment 2 SID is 150024, representing the PE-2 Adj-SID for the link PE-2-PE-4
- Segment 3 SID is 150046, representing the PE-4 Adj-SID for the link PE-4-PE-6
- Segment 4 SID is 50405, which is the Node-SID of PE-5.

A more optimal SID stack is achievable in this topology, but the configured segment list shows the use of both Node- and Adj-SIDs on a loose or strict hop basis. The segment list has an optional weight parameter used for load-balancing across multiple segment lists. In this example, only a single segment list exists, so the default weight value of 1 is retained.

Finally, both the segment list and the static SR policy are enabled (**admin-state enable**). The following output shows the operational state of the static SR policy. The **active** field shows whether this candidate path is the selected path in the presence of multiple candidate paths. The SR policy segment list is considered valid if the head-end is able to perform path resolution for the first SID in the segment list into one or more outgoing interfaces and next-hops. The segment 1 label is 50402, and the state is shown as **resolved-up**, indicating that this is a valid segment list.

```
[/]
A:admin@PE-1# show router segment-routing sr-policies static

=====
SR-Policies Path
=====
-----
Active           : Yes                Owner           : static
Color            : 600
Head              : 0.0.0.0             Endpoint Addr   : 192.0.2.5
RD               : 600001005          Preference     : 100
BSID             : 100002
TunnelId         : 917506            Age             : 48
Origin ASN       : 0                 Origin          : 0.0.0.0
NumReEval        : 0                 ReEvalReason   : none
NumActPathChange: 0                 Last Change    : 09/14/2021 10:24:05
Maintenance Policy: N/A

Path Segment Lists:
Segment-List     : 1                 Weight          : 1
S-BFD State      : Down             S-BFD Transitio*: 0
Num Segments     : 4                 Last Change    : 09/14/2021 09:37:52
  Seg 1 Label    : 50402             State          : resolved-up
  Seg 2 Label    : 150024            State          : N/A
  Seg 3 Label    : 150046            State          : N/A
  Seg 4 Label    : 50405             State          : N/A

=====
* indicates that the corresponding row element may have been truncated.
```

If the SR policy is considered valid, it is populated in the tunnel table with an owner of sr-policy. The entry indicates the destination and color, and always has a metric value of 0 regardless of how the SR policy is instantiated. The metric value of 0 is used because there is no effective way for the head-end to determine a more reflective value for an SR policy when learned through BGP SR TE policy or statically configured.

```
[/]
A:admin@PE-1# show router tunnel-table 192.0.2.5 protocol sr-policy

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.5/32         sr-policy MPLS  917506   14    192.0.2.2    0
  600
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
```

Traffic steering using SR policies

A head-end can potentially steer traffic using an SR policy as a midpoint (or BSID anchor) or as an ingress router using color-based traffic steering:

- At a midpoint or BSID anchor, if an incoming packet has an active label that matches the BSID of a valid SR policy, the incoming label is swapped for the labels contained in the active path of that SR policy, and traffic is forwarded along that path.
- At an ingress router, if a BGP or service route is received containing a Color Extended Community with a value corresponding to a valid local SR policy, and the endpoint of that SR policy matches the next-hop of the BGP/service route, traffic is forwarded into the associated SR policy.

This sub-section discusses the use of the Color Extended Community to implement traffic steering at an ingress router, and begins with an overview of the structure of the Color Extended Community.

The Color Extended Community has two flags, known as the Color-Only (CO) bits, that allow for a head-end to optionally steer traffic using an SR policy, without the need to explicitly define an SR policy endpoint that matches the next-hop of a BGP or service route. In this case, the endpoint can be the null address (0.0.0.0 for IPv4 and 0::0 for IPv6) and traffic is steered by an SR policy based on correlation of color. [Table 18: Use of CO bits](#) describes the destination steering options based on the setting of the Color-Only (CO) bits.

Table 18: Use of CO bits

CO bits=00	CO bits=01	CO bits=10
If there is a valid SR policy (N, C), where N is the IPv4 or IPv6 endpoint address and C is a color, steer using SR policy (N, C);	If there is a valid SR policy (N, C), where N is the IPv4 or IPv6 endpoint address and C is a color, steer using SR policy (N, C);	If there is a valid SR policy (N, C), where N is the IPv4 or IPv6 endpoint address and C is a color, steer using SR policy (N, C);

CO bits=00	CO bits=01	CO bits=10
Else, steer on the IGP path to the next-hop N	Else, if there is a valid SR policy (null endpoint, C) of the same address family as N, steer using SR policy (null endpoint, C);	Else, if there is a valid SR policy (null endpoint, C) of the same address family as N, steer using SR policy (null endpoint, C);
	Else, if there is any valid SR policy (any address family null endpoint, C), steer using SR policy (any null endpoint, C);	Else, if there is any valid SR policy (any address family null endpoint, C), steer using SR policy (any null endpoint, C);
	Else, steer on the IGP path to the next-hop N	Else, if there is any valid SR policy (any endpoint, C) of the same address family as N, steer using SR policy (any endpoint, C);
		Else, if there is any valid SR policy (any address family endpoint, C), steer using SR policy (any address family endpoint, C);
		Else, steer on the IGP path to the next-hop N

Per-destination traffic steering

When incoming packets match a BGP/service route with a next-hop that resolves to an SR policy, it is referred to as per-destination traffic steering. The previously configured static SR policy at PE-1 with color 600 is used to show how it is applied. A VPRN service (600) is extended between PE-1 and PE-5 with import/export Route-Target 64496:600, and with the auto-bind-tunnel resolution-filter at PE-1 set to SR policy (the complete VPRN service configuration is not shown for conciseness).

```
# on PE-1:
configure {
  service {
    vprn "600" {
      admin-state enable
      service-id 600
      customer "1"
      autonomous-system 64496
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "64496:600"
          vrf-import {
            policy ["vrf600-import"]
          }
          vrf-export {
            policy ["vrf600-export"]
          }
        }
      }
    }
  }
}
```

```

    }
    auto-bind-tunnel {
      resolution filter
      resolution-filter {
        sr-policy true
      }
    }
  }
}

```

A CE router is locally connected to PE-5 and advertises prefix 10.148.5.0/24 to IPv4 BGP, which PE-5 subsequently advertises as a VPN-IPv4 route. In addition to attaching the Route-Target Extended Community to the VPN-IPv4 route, PE-5 also attaches a Color Extended Community with value 600. The PE-5 VRF export policy is shown following. When configuring the Color Extended Community, the syntax "color:co:value" is used. Therefore, in the example configuration, the CO bits are 00 and the color value is 600.

```

# on PE-5:
configure {
  policy-options {
    community "vrf600-export" {
      member "target:64496:600" { }
    }
    community "vrf600-sr-policy" {
      member "color:00:600" { }
    }
  }
  policy-statement "vrf600-export" {
    entry 10 {
      from {
        protocol {
          name [bgp]
        }
      }
      to {
        protocol {
          name [bgp-vpn]
        }
      }
      action {
        action-type accept
        community {
          add ["vrf600-export" "vrf600-sr-policy"]
        }
      }
    }
  }
}

```

At PE-1, the VPN-IPv4 route with {next-hop PE-5, color extended community 600} resolves to the PE-1 static SR policy with {endpoint PE-5, color 600}. It is imported into the VPRN route-table with an indication that it is resolved to the SR policy with tunnel ID 917506, which is the tunnel ID of the previously configured static SR policy. Traffic from PE-1 to PE-5 is therefore forwarded into the SR policy using the label stack defined in the segment list.

```

[/]
A:admin@PE-1# show router 600 route-table 10.148.5.0/24

```

```

=====
Route Table (Service: 600)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]              Metric
-----
10.148.5.0/24                    Remote BGP VPN 00h04m54s 170
    192.0.2.5 (tunneled:SR-Policy:917506) 0
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====

```

Color-Only traffic steering

SR OS provides support for Color-Only traffic steering using a null endpoint SR policy, but its use is limited to unlabeled BGP address families because of the following. When an egress router advertises a downstream label in a labeled BGP update (VPN-IPv4/IPv6, EVPN, BGP Labeled Unicast, and so on) that egress router needs to see that label in received packets to be able to demultiplex into the relevant service/next-hop and forward traffic toward the destination. If a head-end is forwarding traffic using an SR policy with a null endpoint, that head-end is unaware of the egress router, so cannot impose the relevant downstream-advertised BGP/service label into the label stack.

The following example shows the configuration of Color-Only traffic steering. PE-5 advertises an IPv4 prefix 172.16.5.1/32 to PE-1 with the Color Extended Community 01:600. PE-1 intends to use the previously configured static SR policy to resolve this route. As described in [Table 18: Use of CO bits](#), with the CO-bits set to 01, the head-end uses an SR policy with (null endpoint, C) if no valid (N, C) SR policy exists.

```

[~/A:admin@PE-5# show router bgp routes 172.16.5.1/32 hunt | match 'Network|Nexthop|Community'
Network       : 172.16.5.1/32
Nexthop       : 192.0.2.5
Res. Nexthop  : n/a
Community     : color:01:600

```

At PE-1, the static SR policy to PE-5 is reconfigured such that the endpoint is no longer an explicit endpoint of 192.0.2.5 (PE-5), but instead uses a null endpoint (0.0.0.0).

```

# on PE-1:
configure {
  router "Base" {
    segment-routing {
      sr-policies {
        static-policy "PE-1-PE-5-color600" {
          endpoint 0.0.0.0
        }
      }
    }
  }
}

```

Since PE-5 advertised an IPv4 BGP prefix, PE-1 also enables the use of BGP shortcuts, with a resolution filter that only permits the use of SR policy.

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      next-hop-resolution {
        shortcut-tunnel {
          family ipv4 {
            resolution filter
            resolution-filter {
              sr-policy true
            }
          }
        }
      }
    }
  }
}
```

The tunnel table of PE-1 shows that there is a single SR policy active with a destination of 0.0.0.0/32 (null) and color 600.

```
[/]
A:admin@PE-1# show router tunnel-table protocol sr-policy

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
0.0.0.0/32      sr-policy MPLS  917507   14    192.0.2.2    0
  600
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
```

The status of the IPv4 prefix 172.16.5.1/32 received from PE-5 is shown in the following output at PE-1. The output shows that the route is Used/Valid/Best, and that the resolving protocol is SR-POLICY, and the resolving NextHop is 0.0.0.0. Therefore, a BGP next-hop has been resolved to a null endpoint SR policy using the CO-bits.

```
[/]
A:admin@PE-1# show router bgp routes 172.16.5.1/32 detail

=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Original Attributes
```

```

Network      : 172.16.5.1/32
Nexthop     : 192.0.2.5
Path Id     : None
From       : 192.0.2.7
Res. Protocol : SR-POLICY           Res. Metric   : 0
Res. Nexthop : 0.0.0.0 (SR-POLICY)
Local Pref. : 100
Aggregator AS : None                Interface Name : NotAvailable
Atomic Aggr. : Not Atomic           Aggregator    : None
AIGP Metric  : None                 MED           : None
Connector    : None                 IGP Cost      : 0
Community    : color:01:600
Cluster      : 192.0.2.7
Originator Id : 192.0.2.5           Peer Router Id : 192.0.2.7
Fwd Class    : None                 Priority       : None
Flags        : Used Valid Best IGP In-RTM
Route Source : Internal
AS-Path      : No As-Path
Route Tag    : 0
Neighbor-AS  : n/a
Orig Validation: NotFound
---snip---

```

Advertising SR policies into BGP

Before advertising SR policies into BGP, all previous static SR policy configuration is removed. The simulated controller acts as the source of BGP advertised SR policies, and when an SR OS router advertises SR policies into BGP they must first be statically configured to provide the relevant information to populate the BGP path attributes. The following static SR policy is applied at the controller representing a similar SR policy to that previously configured at PE-1. The SR policy has a head-end of PE-1 (192.0.2.1), an endpoint of PE-5 (192.0.2.5), and a color of 600. The segment list is modified slightly to represent a list of strict hops using Adj-SIDs along the path PE-1-PE-2-PE-4-PE-6-PE-5.

```

# on controller:
configure {
  router "Base" {
    segment-routing {
      sr-policies {
        static-policy "color600-PE-1-PE-5" {
          admin-state enable
          color 600
          endpoint 192.0.2.5
          head-end 192.0.2.1
          binding-sid 100002
          distinguisher 600001005
          segment-list 1 {
            admin-state enable
            segment 1 {
              mpls-label 150012 # adj-SID int-PE-1-PE-2
            }
            segment 2 {
              mpls-label 150024 # adj-SID int-PE-2-PE-4
            }
            segment 3 {
              mpls-label 150046 # adj-SID int-PE-4-PE-6
            }
            segment 4 {
              mpls-label 150065 # adj-SID int-PE-6-PE-5
            }
          }
        }
      }
    }
  }
}

```



```

BSID/Pref/TunnType: 100002/100/sr-policy
Nexthop      : 0.0.0.0
From         : BGP
Res. Nexthop : n/a
Local Pref.  : None
Aggregator AS : None
Atomic Aggr. : Not Atomic
AIGP Metric  : None
Connector    : None
Community    : target:192.0.2.1:0
Cluster      : No Cluster Members
Originator Id : None
Flags        : Used Valid Best IGP
Route Source : Internal
AS-Path      : No As-Path
Route Tag    : 0
Neighbor-AS  : n/a
Orig Validation: N/A
Source Class : 0
Add Paths Send : Default
Last Modified : 00h00m51s
Interface Name : NotAvailable
Aggregator     : None
MED            : None
IGP Cost       : 0
Peer Router Id : 0.0.0.0
Dest Class     : 0
    
```

RIB Out Entries

```

RD/Color/End Pt: 600001005/600/192.0.2.5
BSID/Pref/TunnType: 100002/100/sr-policy
Nexthop      : 192.0.2.254
To         : 192.0.2.7
Res. Nexthop : n/a
Local Pref.  : 100
Aggregator AS : None
Atomic Aggr. : Not Atomic
AIGP Metric  : None
Connector    : None
Community    : target:192.0.2.1:0
Cluster      : No Cluster Members
Originator Id : None
Origin       : IGP
AS-Path      : No As-Path
Route Tag    : 0
Neighbor-AS  : n/a
Orig Validation: N/A
Source Class : 0
Dest Class   : 0
Interface Name : NotAvailable
Aggregator     : None
MED            : None
IGP Cost       : n/a
Peer Router Id : 192.0.2.7
    
```

Routes : 2
=====

The following output from PE-1 shows that the SR policy is active and that the first SID in the segment list has been correctly resolved; the owner is bgp.

```

[/]
A:admin@PE-1# show router segment-routing sr-policies bgp color 600 end-point 192.0.2.5
=====
SR-Policies Path
=====
-----
Active      : Yes           Owner       : bgp
Color        : 600
Head         : 0.0.0.0      Endpoint Addr : 192.0.2.5
RD           : 600001005    Preference   : 100
    
```

```

BSID          : 100002
TunnelId      : 917508      Age           : 225
Origin ASN    : 64496      Origin        : 192.0.2.254
NumReEval     : 0          ReEvalReason  : none
NumActPathChange: 0       Last Change   : 09/14/2021 11:33:25
Maintenance Policy: N/A

Path Segment Lists:
Segment-List  : 1          Weight        : 1
S-BFD State   : Down      S-BFD Transitio*: 0
Num Segments  : 4          Last Change   : 09/14/2021 09:37:52
  Seg 1 Label : 150012     State         : resolved-up
  Seg 2 Label : 150024     State         : N/A
  Seg 3 Label : 150046     State         : N/A
  Seg 4 Label : 150065     State         : N/A
    
```

=====
* indicates that the corresponding row element may have been truncated.

Verification is also made at PE-1 that the SR policy is correctly populated in the tunnel table.

```

[/]
A:admin@PE-1# show router tunnel-table 192.0.2.5/32 protocol sr-policy

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.5/32sr-policy MPLS  917508   14    192.168.12.2  0
600
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
    
```

The procedure for traffic steering using an SR policy learned through BGP SR TE policy is the same as traffic steering using a statically configured SR policy, and is therefore not repeated here.

BSID anchor

The statically configured and BGP advertised SR policies used so far in this chapter have been instantiated on a head-end that uses the Color Extended Community to steer traffic. An alternative method of steering traffic using an SR policy is through the use of the BSID. If an incoming packet has an active label that matches the BSID of a valid SR policy, the packet is forwarded using that SR policy and the incoming label is swapped for the labels that the SR policy contains.

Using a BSID in this way is considered useful at domain interconnects such as ABRs or ASBRs. It provides opacity between the domains and protects the churn from one domain from entering another domain. In large networks, it has the additional benefit of reducing the number of labels an ingress router needs to impose, because the BSID can expand a single incoming SID/label stack (the BSID) into a much larger outgoing SID/label stack.

The example topology in [Figure 328: Example topology](#) is entirely IS-IS Level 2, so not constructed of multiple domains. However, it is still sufficient to show the use of BSID traffic steering. In the following

example, PE-3 becomes a BSID anchor for an SR policy path extended between PE-1 and PE-5. This requires the instantiation of two SR policies:

- An SR policy at PE-3 with a segment list that constructs the required path to PE-5. Like every SR policy, it requires a BSID, but in this case the BSID is programmed in the Incoming Label Map (ILM) table and has a next-hop Label Forwarding Entry (NHLFE) that includes the segments (labels) in the segment list.
- An SR policy at PE-1 with a segment list specifying a path to PE-3, followed by a segment that references the relevant BSID programmed at PE-3.

The following output shows the SR policy advertised in BGP to PE-3. It uses color 700 and has an endpoint of PE-5 (192.0.2.5). Since traffic steering at PE-3 using the SR policy is achieved using the BSID, any color value could be used (although different colors may be needed to represent different path characteristics). Packets are classified upstream of PE-3 at PE-1, and the result of that classification selects the relevant BSID to meet the path requirements. The segment list programs a path to PE-5 using Adj-SIDs along the path PE-3-PE-4-PE-6-PE-5. The BSID value is 100001.

```
[/]
A:admin@PE-3# show router segment-routing sr-policies bgp color 700

=====
SR-Policies Path
=====
-----
Active      : Yes           Owner       : bgp
Color      : 700
Head       : 0.0.0.0    Endpoint Addr : 192.0.2.5
RD        : 700003005  Preference  : 100
BSID      : 100001
TunnelId   : 917506    Age         : 22
Origin ASN : 64496      Origin      : 192.0.2.254
NumReEval  : 0         ReEvalReason : none
NumActPathChange: 0    Last Change  : 09/14/2021 11:40:46
Maintenance Policy: N/A

Path Segment Lists:
Segment-List : 1           Weight      : 1
S-BFD State  : Down       S-BFD Transitio*: 0
Num Segments : 3           Last Change : 09/14/2021 09:38:04
  Seg 1 Label : 150034     State       : resolved-up
  Seg 2 Label : 150046     State       : N/A
  Seg 3 Label : 150065     State       : N/A

=====
* indicates that the corresponding row element may have been truncated.
```

The following output shows the SR policy advertised in BGP to PE-1. It uses color 700 and has an endpoint of PE-5 (192.0.2.5). The segment list programs a path that contains the following:

- The Node-SID of PE-3 (50403)
- The BSID programmed at PE-3 for the path to PE-5 (100001). When PE-3 pops its Node-SID and this label is exposed at PE-3, it swaps label 100001 for the label stack contained in the SR policy of that BSID.
- The Node-SID of PE-5 (50405)

```
[/]
A:admin@PE-1# show router segment-routing sr-policies bgp color 700

=====
```

```

SR-Policies Path
=====
-----
Active       : Yes           Owner       : bgp
Color       : 700
Head        : 0.0.0.0       Endpoint Addr : 192.0.2.5
RD          : 700001003     Preference  : 100
BSID        : 100003
TunnelId    : 917509       Age         : 19
Origin ASN  : 64496        Origin      : 192.0.2.254
NumReEval   : 0           ReEvalReason : none
NumActPathChange: 0       Last Change : 09/14/2021 11:40:46
Maintenance Policy: N/A

Path Segment Lists:
Segment-List : 1           Weight      : 1
S-BFD State  : Down       S-BFD Transitio*: 0
Num Segments : 3           Last Change : 09/14/2021 09:37:52
  Seg 1 Label : 50403      State       : resolved-up
  Seg 2 Label : 100001     State       : N/A
  Seg 3 Label : 50405      State       : N/A
=====
* indicates that the corresponding row element may have been truncated.

```

A VPRN service (700) is extended between PE-1 and PE-5 with import/export Route-Target 64496:700, and with the auto-bind-tunnel resolution-filter set to SR policy at PE-1 (the complete VPRN service configuration is not shown for conciseness).

```

# on PE-1:
configure {
  service {
    vprn "VPRN_700" {
      admin-state enable
      service-id 700
      customer "1"
      autonomous-system 64496
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "64496:700"
          vrf-import {
            policy ["vrf700-import"]
          }
          vrf-export {
            policy ["vrf700-export"]
          }
          auto-bind-tunnel {
            resolution filter
            resolution-filter {
              sr-policy true
            }
          }
        }
      }
    }
  }
}

```

PE-5 advertises prefix 172.31.5.1/32 as a VPN-IPv4 route. In addition to the Route-Target Extended Community attached to the VPN-IPv4 route, PE-5 also attaches a Color Extended Community with value 700. PE-5's VRF export policy is as follows:

```
# on PE-5:
configure {
  policy-options {
    community "vrf700-export" {
      member "target:64496:700" { }
    }
    community "vrf700-sr-policy" {
      member "color:00:700" { }
    }
    prefix-list "vrf700-prefixes" {
      prefix 172.31.5.1/32 type exact {
    }
  }
  policy-statement "vrf700-export" {
    entry 10 {
      from {
        prefix-list ["vrf700-prefixes"]
      }
      to {
        protocol {
          name [bgp-vpn]
        }
      }
      action {
        action-type accept
        community {
          add ["vrf700-export" "vrf700-sr-policy"]
        }
      }
    }
  }
}
}
```

PE-1 also advertises prefix 172.31.1.1/32 as a VPN-IPv4 route to allow for connectivity to be validated end-to-end through both SR policies. The first of the following outputs shows PE-1s tunnel-table containing the advertised SR policy to PE-5 with tunnel ID 917509 and next-hop 192.0.2.3.

```
[/]
A:admin@PE-1# show router tunnel-table 192.0.2.5/32 protocol sr-policy

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.5/32     sr-policy MPLS  917508   14    192.168.12.2  0
  600
192.0.2.5/32   sr-policy MPLS  917509  14    192.0.2.3   0
  700
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====
```

The next output shows the route-table of VPRN 700 at PE-1 where the VPN-IPv4 prefix 172.31.5.1/32 advertised by PE-5 is resolved to an SR policy with tunnel ID 917509. As in the previous output, this is the SR policy advertised in BGP containing the BSID at PE-3.

```
[/]
A:admin@PE-1# show router 700 route-table

=====
Route Table (Service: 700)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]                Type   Proto   Age      Pref
Metric
-----
172.31.1.1/32
loopback0-700                          Local  Local   00h07m18s  0
0
172.31.5.1/32
192.0.2.5 (tunneled:SR-Policy:917509)    Remote BGP VPN 00h06m20s 170
0
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
```

The datapath between PE-1 and PE-5 is verified using a ping:

```
[/]
A:admin@PE-1# ping 172.31.5.1 router-instance "VPRN_700" source-address 172.31.1.1
PING 172.31.5.1 56 data bytes
64 bytes from 172.31.5.1: icmp_seq=1 ttl=64 time=3.16ms.
64 bytes from 172.31.5.1: icmp_seq=2 ttl=64 time=3.10ms.
64 bytes from 172.31.5.1: icmp_seq=3 ttl=64 time=3.01ms.
64 bytes from 172.31.5.1: icmp_seq=4 ttl=64 time=2.97ms.
64 bytes from 172.31.5.1: icmp_seq=5 ttl=64 time=2.49ms.

---- 172.31.5.1 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 2.49ms, avg = 2.94ms, max = 3.16ms, stddev = 0.239ms
```

By enabling egress statistics for SR policies at PE-3, it is also possible to see the number of packets and octets being forwarded using the SR policy.

```
# on PE-3:
configure {
  router "Base" {
    segment-routing {
      sr-policies {
        egress-statistics {
          admin-state enable
        }
      }
    }
  }
}

[/]
A:admin@PE-3# show router segment-routing sr-policies egress-statistics color 700 end-point
192.0.2.5

=====
```

SR-Policies Egress Statistics

Egress Statistics:

```

Color          : 700                Endpoint Addr   : 192.0.2.5
Segment-List   : 1
TunnelId       : 917506             BSID           : 100001
Pkt Count    : 10                Octet Count   : 1220
    
```

Weighted Equal Cost Multipath

Support for weighted Equal Cost Multipath (ECMP) is provided with SR policies using multiple segment lists. Each segment list contains a path from the head-end to the endpoint, and each segment list contains a weight used to influence ECMP forwarding. The following output at the controller shows the use of multiple segment lists for an SR policy with a head-end of PE-1 (192.0.2.1), an endpoint of PE-6 (192.0.2.6), and a color of 800. Segment list 1 encodes a path consisting of Node-SIDs along the path PE-1-PE-3-PE-5-PE-6 and has a weight of 40. Segment list 2 encodes a path consisting of Node-SIDs along the path PE-1-PE-2-PE-4-PE-6 and has a weight of 60.

```

# on controller:
configure {
  router "Base" {
    segment-routing {
      sr-policies {
        static-policy "color800-PE-1-PE-6" {
          admin-state enable
          color 800
          endpoint 192.0.2.6
          head-end 192.0.2.1
          binding-sid 100001
          distinguisher 800001006
          segment-list 1 {
            admin-state enable
            weight 40
            segment 1 {
              mpls-label 50403    # node-SID PE-3
            }
            segment 2 {
              mpls-label 50405    # node-SID PE-5
            }
            segment 3 {
              mpls-label 50406    # node-SID PE-6
            }
          }
          segment-list 2 {
            admin-state enable
            weight 60
            segment 1 {
              mpls-label 50402    # node-SID PE-2
            }
            segment 2 {
              mpls-label 50404    # node-SID PE-4
            }
            segment 3 {
              mpls-label 50406    # node-SID PE-6
            }
          }
        }
      }
    }
  }
}
    
```


L = Loop-Free Alternate (LFA) hop available
E = Inactive best-external BGP route
k = RIB-API or Forwarding Policy backup hop

=====

Conclusion

SR policies provide an effective way for instantiating traffic engineered SR tunnels that may be statically configured or advertised into BGP from either a controller or a router. Segments of paths constructed using SR policies can be loose or strict, using any combination of SIDs. The use of BSIDs also provides a way to interconnect domains and reduces the label stack imposition required at ingress routers. BGP can be used to advertise and instantiate SR policies that can be used as a method of steering traffic.

Inter-AS Model C VPRN Using MPLS Forwarding Policies and Segment Routing Policies

This chapter provides information about Inter-AS Model C VPRN using MPLS Forwarding Policies and Segment Routing Policies.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information and configuration in this chapter are based on MD-CLI for SR OS Release 21.7.R1. MPLS label binding forwarding policies and segment routing policies are supported in SR OS Release 16.0.R1, or later. MPLS endpoint forwarding policies and ECMP are supported in SR OS Release 16.0.R4, or later.

Overview

In this configuration, MPLS forwarding policies are combined with segment routing policies. In the remainder of this chapter, SR refers to "Segment Routing", unless specified otherwise. Product and release references, such as 7750 SR and SR OS, continue to refer to "Service Router".

MPLS forwarding policies

SR OS uses the following table management to forward packets:

- Route Table Manager (RTM) for IP packets matching IP route prefixes in the Global Route Table (GRT) that resolve to IP next-hops or tunnel next-hops (for IGP shortcuts).
- Tunnel Table Manager (TTM) containing Next-Hop Label Forwarding Entries (NHLFEs) to forward IP packets for routes in GRT or VPRN using tunnels. The resolved next-hop of the IP packet is matched to the far-end address of the TTM entry.
- Incoming Label Map (ILM) containing labels matching a specific Forwarding Equivalence Class (FEC), such that packets with this label are sent to the destination of the FEC.

The ILM tunnel is programmed via the service module, the MPLS module, and various control plane protocols supporting labeled tunnels or FECs. The GRT and TTM provide some flexibility, but do not allow customization, such as the ability to create specific sets of IP direct next-hops, IP indirect next-hops, or tunnel next-hops for a specific set of flows or prefixes. For more flexibility, the following can be configured:

- static routes
- traffic steering; for example, using Openflow, and Policy-Based Routing (PBR)
- MPLS forwarding policies

MPLS forwarding policies establish Static Label Routes (SLRs). The binding label of the forwarding policy is popped when matched on an incoming packet. If no pushed label is configured, then it becomes a swap to implicit-null, which is essentially a pop operation. After the incoming label is popped, the exposed packet payload (or the next label after the top label is removed) is forwarded via the configured next-hop of the MPLS forwarding policy. The next-hop is looked up in the route table and can be direct or indirect. A direct next-hop is an attached local interface; an indirect next-hop is a resolved route.

MPLS label-binding forwarding policies use labels from a reserved label block also known as a Segment Routing Local Block (SRLB), whereas node SIDs in segment routing use the Segment Routing Global Block (SRGB) instead. An SRLB is used for the following:

- static adjacency SIDs
- adjacency set SIDs
- SR policy binding SIDs (BSIDs)
- MPLS forwarding policy binding labels

MPLS forwarding policies allow the forwarding of packets over a set of user-defined next-hops: either direct next-hops (with option to push a label stack) or indirect next-hops.

MPLS forwarding policies are validated as follows:

- the binding label must be in the label range of the defined reserved label block (SRLB) and it must be unused; the same label cannot be allocated more than once
- the direct next-hop interfaces must be up
- the indirect next-hops must be reachable

MPLS forwarding policies work in one of two modes:

- ILM mode: label binding policy for labeled packets
- LTN mode: endpoint policy for unlabeled packets (this is beyond the scope of this chapter)

The data model of a forwarding policy represents the primary and the backup next-hop as a Next-Hop Group (NHG) and models the ECMP as the set of NHGs. Flows of prefixes can be switched on a per-NHG basis -without disturbing flows forwarded over other NHGs of the policy- from the failing primary next-hop to the backup next-hop or from the backup next-hop to the restored primary next-hop.

SR policies

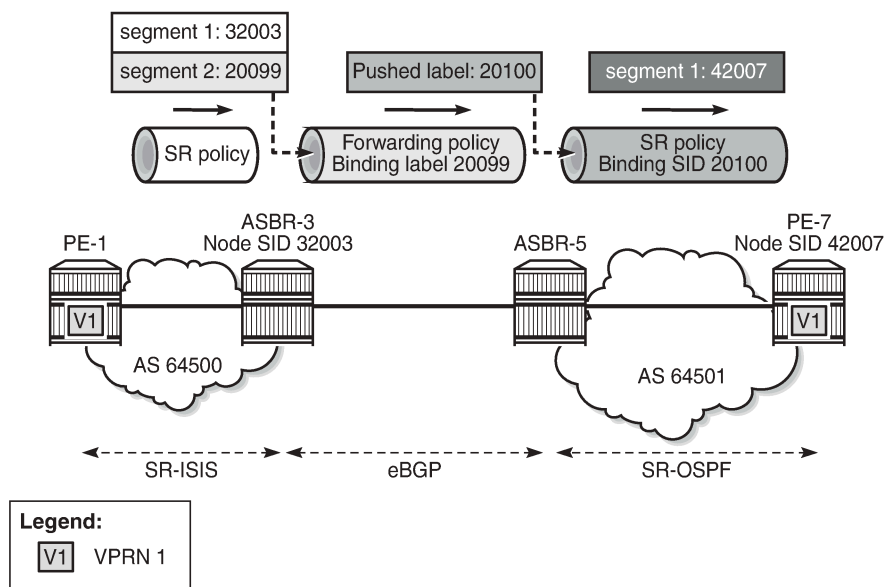
SR policies contain a list of MPLS labels in the form of a segment list that instantiates Segment Routing - Traffic Engineering (SR-TE) Label Switched Paths (LSPs) to a network endpoint and are described in the SR policies chapter.

Inter-AS VPRN Model C using an MPLS forwarding policy and SR policies

One typical application to use MPLS forwarding policies together with SR policies is an example of static Egress Peer Engineering (EPE); a head-end PE in AS1 can steer traffic toward AS2 using a specific AS2 next-hop node.

In the following example, an MPLS forwarding policy is configured on the Autonomous System Border Routers (ASBRs) in an inter-AS VPRN scenario. [Figure 329: Inter-AS VPRN Model C using MPLS forwarding policy and SR policies](#) shows the labels added to a packet sent by VPRN 1 on PE-1 in AS 64500 to VPRN 1 on PE-7 in AS 64501.

Figure 329: Inter-AS VPRN Model C using MPLS forwarding policy and SR policies



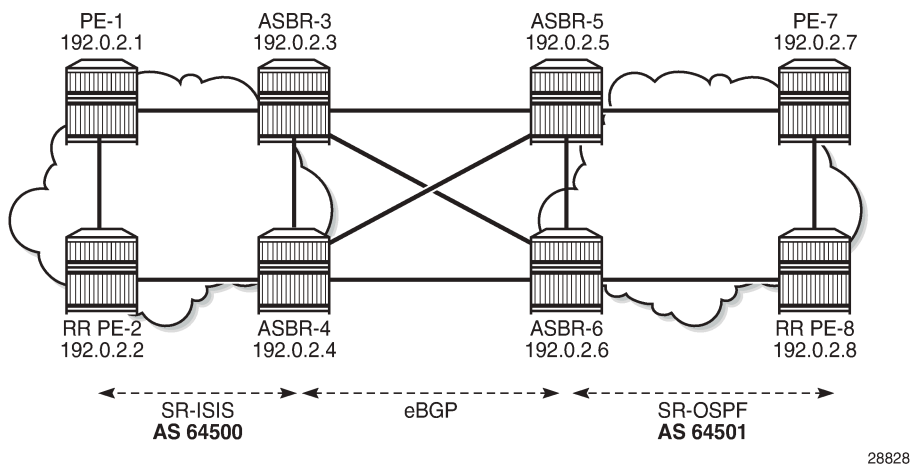
28827

An SR policy on PE-1 pushes two labels: label 32003 from the SRGB for segment routing to SID 32003 of ASBR-3, and label 20099 from the SRLB corresponding to the binding label of the MPLS forwarding policy to be used in ASBR-3. In ASBR-3, these labels are popped and the MPLS forwarding policy is applied. This MPLS forwarding policy forwards the packet to ASBR-5 and pushes a binding label 20100 from the SRLB on ASBR-5, which identifies the SR policy to be used on ASBR-5. In ASBR-5, label 20100 is popped and an SR policy with binding SID 20100 is applied. This SR policy pushes label 42007, which is the SID of PE-7.

Configuration

Figure 330: Example topology shows the example topology with four routers in AS 64500 and four routers in AS 64501. SR-ISIS is configured in AS 64500, while SR-OSPF is configured in AS 64501. The SR policies are configured within the ASs, whereas the MPLS forwarding policies are used to set up tunnels between the ASBRs.

Figure 330: Example topology



The initial configuration includes:

- Cards, MDAs, ports
- Router interfaces
- IS-IS as IGP between the routers in AS 64500; OSPF between the routers in AS 64501

Segment routing

SR-ISIS is configured in AS 64500. The following SR-ISIS configuration on PE-1 shows that the prefix SIDs are taken from the SRGB, which uses labels from 32000 to 32999, and the scope of the router capability advertisement is the area. The system interface has SID index 1, so the node SID label will be start label + index = 32000 + 1 = 32001.



Note:

The SRGB block does not need to have the same start value and end value on each router in the AS, but it must have the same size, that is, the same number of labels in the SRGB.

```
# on PE-1:
configure {
  router "Base" {
    autonomous-system 64500
    mpls-labels {
      sr-labels {
        start 32000
        end 32999 # SRGB block definition AS 64500
      }
    }
  }
  isis 0 { # IS-IS in the AS 64500; OSPF in ASs 64501
    admin-state enable
    advertise-router-capability area
    level-capability 2
    traffic-engineering true
    area-address [49.0001]
    segment-routing {
      admin-state enable
      prefix-sid-range {
```


Inter-AS VPRN Model C

The configuration of inter-AS Model C VPRNs is described in the "Inter-AS VPRN Model C" chapter in the Layer 3 Services volume of *7450 ESS, 7750 SR, and 7950 XRS Advanced Configuration Guide — Book II*. PE-2 acts as the Route Reflector (RR) for the "iBGP_grp" group in AS 64500; in AS 64501, PE-8 acts as the RR. Between AS 64500 and AS 64501, eBGP is configured on the ASBRs.

The RR addresses need to be advertised between the ASs. This can be done using IPv4 or labeled IPv4 (with next-hop resolution enabled). No other PE system addresses need to be advertised.

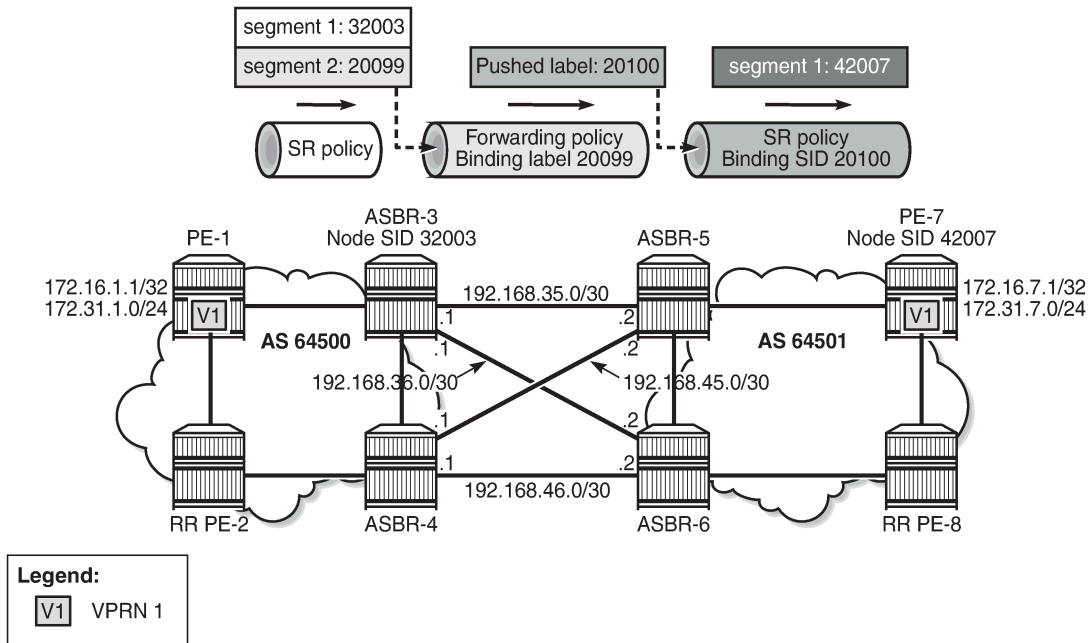
On PE-1, the system IP address 192.0.2.1/32 need not be exported, because no recursive lookup is required. Instead, the VPRN will be configured with auto-bind to the SR policy and the SR policy tunnel can resolve the next-hop of the VPN-IPv4 route. The following configuration shows the BGP configuration for address family VPN-IPv4 on PE-1:

```
# on PE-1:
configure {
  router "Base" {
    bgp {
      split-horizon true
      group "iBGP_grp" {
        type internal
        family {
          vpn-ipv4 true
        }
      }
      neighbor "192.0.2.2" {
        group "iBGP_grp"
      }
    }
  }
}
```

On the ASBRs, BGP is configured for the labeled IPv4 address family only. On ASBR-3 and ASBR-4, the BGP next-hop for the labeled IPv4 address family can be resolved using SR-ISIS within AS 64500; on ASBR-5 and ASBR-6, the next-hop can be resolved using SR-OSPF within AS 64501. The forwarding between the ASBRs is based on label-binding MPLS forwarding policies. On ASBR-3, BGP is configured as follows:

```
# on ASBR-3:
configure {
  router "Base" {
    bgp {
      split-horizon true
      next-hop-resolution {
        labeled-routes {
          transport-tunnel {
            family label-ipv4 {
              resolution-filter {
                sr-isis true
              }
            }
          }
        }
      }
    }
  }
  group "eBGP_grp" {
    peer-as 64501
    advertise-inactive true
    family {
      label-ipv4 true
    }
  }
}
```


Figure 331: Inter-AS VPRN using MPLS forwarding policy and SR policies: Traffic to PE-7



28830

SR policies -like MPLS forwarding policies- use labels from an SRLB, which is a pool of labels defined as follows:

```
# on PE-1:
configure {
  router "Base" {
    mpls-labels {
      reserved-label-block "SRLB1" {
        start-label 20000
        end-label 21999
      }
    }
  }
}
```

SR policies contain MPLS labels in a segment list that instantiates SR-TE LSPs to a network endpoint. This appears in the tunnel table as an SR policy tunnel. On PE-1, the following SR policy-with endpoint 192.0.2.7 in a remote AS-is configured with one segment list including two segments:

- segment 1 contains label 32003 referencing the node SID of ASBR-3.
- segment 2 contains label 20099 referencing the binding label that matches the MPLS forwarding policy used at ASBR-3.

There are two ways to steer a set of flows into an SR policy: either based on the BSID value or based on a match of color and endpoint.

In this case, for a VPRN with auto-bind-tunnel, the payload prefix (VPN-IPv4 route prefix) must contain a color community value that matches the color value of the SR-policy route, and the prefix BGP next-hop must also match the endpoint value of the SR-policy route.

In addition, the reserved label block SRLB1 must be referenced within the SR policy context because the configured BSID (20000) is checked against this label block.

```
# on PE-1:
configure {
  router "Base" {
    segment-routing {
      sr-policies {
        admin-state enable
        reserved-label-block "SRLB1"
        static-policy "SR-static-policy-EP7" {
          admin-state enable
          color 100
          endpoint 192.0.2.7
          head-end local
          binding-sid 20000
          distinguisher 64500
          segment-list 1 {
            admin-state enable
            segment 1 {
              mpls-label 32003 # node SID of ASBR-3
            }
            segment 2 {
              mpls-label 20099 # binding label of fwd-policy
            }
          }
        }
      }
    }
  }
}
```

On the ASBRs, MPLS forwarding policies are configured. Like SR policies (using BSID), the binding label is taken from reserved label block SRLB1, which is a pool of labels defined as follows:

```
# on ASBR-3:
configure {
  router "Base" {
    mpls-labels {
      reserved-label-block "SRLB1" {
        start-label 20000
        end-label 21999
      }
    }
  }
}
```

The reserved label block SRLB1 must be referenced within the MPLS forwarding policy context on each ASBR. The following MPLS forwarding policy is configured on ASBR-3 with binding label 20099, which maps to the segment 2 label defined in the SR policy on PE-1. The resolution type is set to direct, meaning that the next-hops are locally attached interface IP addresses. The primary next-hop 192.168.35.2 is on ASBR-5 and the backup next-hop 192.168.36.2 is on ASBR-6.

Within the MPLS forwarding policy on ASBR-3, the pushed label 20100 matches the BSID identifying the SR policy at the peer ASBRs (PE-5 and PE-6). In the MPLS forwarding policy on ASBR3, both the primary and backup next-hops are configured with the same pushed label 20100:

```
# on ASBR-3:
configure {
  router "Base" {
```

```

mpls {
  admin-state enable
  forwarding-policies {
    admin-state enable
    reserved-label-block "SRLB1"
    forwarding-policy "SLR-ILM-pushed-label" {
      admin-state enable
      binding-label 20099
      revert-timer 5
      next-hop-group 1 {
        admin-state enable
        resolution-type direct
        primary-next-hop {
          next-hop 192.168.35.2
          pushed-labels 1 {
            label 20100
          }
        }
        backup-next-hop {
          next-hop 192.168.36.2
          pushed-labels 1 {
            label 20100
          }
        }
      }
    }
  }
}

```

On ASBR-5 and ASBR-6, the following SR policy with endpoint PE-7 and binding SID 20100 only contains one segment toward the node SID of PE-7:

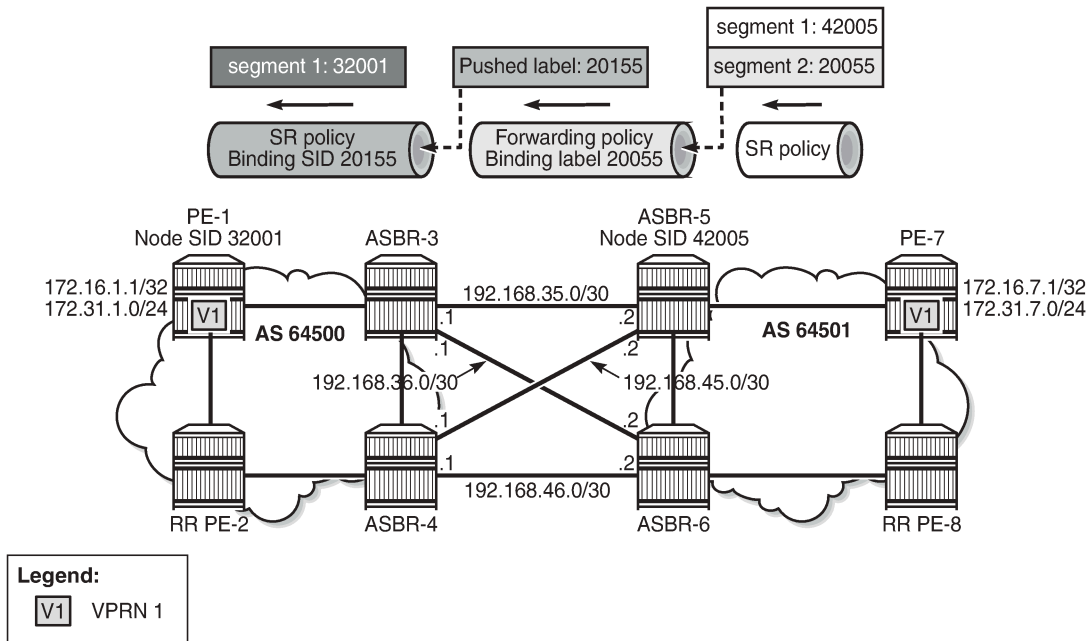
```

# on ASBR-5:
configure {
  router "Base" {
    segment-routing {
      sr-policies {
        admin-state enable
        reserved-label-block "SRLB1"
        static-policy "SR-static-policy-EP7" {
          admin-state enable
          color 100
          endpoint 192.0.2.7
          head-end local
          binding-sid 20100
          distinguisher 64501
          segment-list 1 {
            admin-state enable
            segment 1 {
              mpls-label 42007 # node SID of PE-7
            }
          }
        }
      }
    }
  }
}

```

In the opposite direction, the configuration is similar. [Figure 332: Inter-AS VPRN using MPLS forwarding policy and SR policies: Traffic to PE-1](#) shows the labels used for traffic from PE-7 to PE-1.

Figure 332: Inter-AS VPRN using MPLS forwarding policy and SR policies: Traffic to PE-1



28831

On PE-7, an SR policy is created with endpoint 192.0.2.1 and a segment list with two segments: segment 1 contains label 42005, which is the node SID of ASBR-5, and segment 2 contains label 20055 identifying the binding label in the MPLS forwarding policy at ASBR-5. The configured SR policy color is 100, so this applies for VPN-IPv4 routes with color extended community color:00:100.

```
# on PE-7:
configure {
  router "Base" {
    segment-routing {
      sr-policies {
        admin-state enable
        reserved-label-block "SRLB1"
        static-policy "SR-static-policy-EP1" {
          admin-state enable
          color 100
          endpoint 192.0.2.1
          head-end local
          binding-sid 20001
          distinguisher 64501
          segment-list 1 {
            admin-state enable
            segment 1 {
              mpls-label 42005 # node SID of ASBR-5
            }
            segment 2 {
              mpls-label 20055 # binding label of fwd-policy
            }
          }
        }
      }
    }
  }
}
```

```
}
```

On ASBR-5, both labels (42005 and 20055) are popped and the MPLS forwarding policy with binding label 20055 pushes label 20155 to the primary and backup next-hops. The configuration is as follows:

```
# on ASBR-5:
configure {
  router "Base" {
    mpls {
      admin-state enable
      forwarding-policies {
        admin-state enable
        reserved-label-block "SRLB1"
        forwarding-policy "SLR-ILM-pushed-label" {
          admin-state enable
          binding-label 20055
          revert-timer 5
          next-hop-group 1 {
            admin-state enable
            resolution-type direct
            primary-next-hop {
              next-hop 192.168.35.1
              pushed-labels 1 {
                label 20155
              }
            }
            backup-next-hop {
              next-hop 192.168.45.1
              pushed-labels 1 {
                label 20155
              }
            }
          }
        }
      }
    }
  }
}
}
```

On ASBR-3 and ASBR-4, an SR policy with BSID 20155 is configured. The segment list only contains one segment, which is the node SID of PE-1.

```
# on ASBR-3:
configure {
  router "Base" {
    segment-routing {
      sr-policies {
        admin-state enable
        reserved-label-block "SRLB1"
        static-policy "SR-static-policy-EP1" {
          admin-state enable
          color 100
          endpoint 192.0.2.1
          head-end local
          binding-sid 20155
          distinguisher 64500
          segment-list 1 {
            admin-state enable
            segment 1 {
              mpls-label 32001 # node SID of PE-1
            }
          }
        }
      }
    }
  }
}
```

```

    }
  }
}

```

On PE-1, VPRN 1 is configured with two loopback interfaces to test the traffic, as follows. The tunnel resolution filter within the service is set to **sr-policy**, configured explicitly. The configuration of VPRN 1 on PE-7 is similar, also with two loopback interfaces for test purposes: 17.16.7.1/32 and 172.31.7.1/24.

```

# on PE-1:
configure {
  service {
    vprn "VPRN 1" {
      admin-state enable
      service-id 1
      customer "1"
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "192.0.2.1:1"
          vrf-target {
            import-community "target:64501:1"
            export-community "target:64500:1"
          }
          auto-bind-tunnel {
            resolution filter
            resolution-filter {
              sr-policy true
            }
          }
        }
      }
    }
    interface "lo1" {
      loopback true
      ipv4 {
        primary {
          address 172.31.1.1
          prefix-length 24
        }
      }
    }
    interface "system" {
      loopback true
      ipv4 {
        primary {
          address 172.16.1.1
          prefix-length 32
        }
      }
    }
  }
}

```

The following tunnel table show command on PE-1 returns one tunnel toward PE-7: an SR policy tunnel with color 100.

```

[/]
A:admin@PE-1# show router tunnel-table 192.0.2.7

```



```
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.7/32        sr-policy MPLS  917506   14    192.0.2.3     0
  100
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
```

The following command shows that the SR policy tunnel with tunnel ID 917506 has next-hop 192.0.2.3 and label 20099, which matches the binding label value of the configured MPLS forwarding policy at next-hop ASBR-3.

```
[/]
A:admin@PE-1# show router fp-tunnel-table 1

=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination          Protocol      Tunnel-ID
  Lbl/SID
  NextHop              Intf/Tunnel
  Lbl/SID (backup)
  NextHop (backup)
-----
192.0.2.2/32          SR-ISIS-0    524290
  32002
  192.168.12.2        1/1/2:1000
192.0.2.3/32          SR-ISIS-0    524292
  32003
  192.168.13.2        1/1/1:1000
192.0.2.4/32          SR-ISIS-0    524293
  32004
  192.168.12.2        1/1/2:1000
192.0.2.7/32        SR-Policy   917506
20099
192.0.2.3          SR
192.168.12.2/32      SR            524289
  3
  192.168.12.2        1/1/2:1000
192.168.13.2/32      SR            524291
  3
  192.168.13.2        1/1/1:1000
-----
Total Entries : 6
=====
```

However, on PE-1, the received BGP-VPN routes are not used, as follows:

```
[/]
A:admin@PE-1# show router bgp routes vpn-ipv4
=====
```

```

BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete
=====
BGP VPN-IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
*>i  192.0.2.7:1:172.16.7.1/32              100        None
      192.0.2.7                             None        0
      64501                                  524286
*>i  192.0.2.7:1:172.31.7.0/24             100        None
      192.0.2.7                             None        0
      64501                                  524286
-----
Routes : 2
=====

```

Therefore, the following route table for VPRN 1 does not include any route toward VPRN 1 on PE-7:

```

[/]
A:admin@PE-1# show router 1 route-table
=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]                Type  Proto  Age           Pref
  Next Hop[Interface Name]                Metric
-----
172.16.1.1/32                     Local Local  00h46m33s    0
  system                           0
172.31.1.0/24                     Local Local  00h46m33s    0
  lo1                               0
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====

```

As stated earlier, the following two conditions are required to ensure that the traffic flow from VPRN 1 is steered by the SR policy on PE-1:

- the BGP payload prefix next-hop must match the endpoint value in the SR policy
- the BGP payload prefix must have a color extended community, matching the color value in the SR policy

To match the second condition, the following BGP policy exports the prefixes of VPRN 1 and adds color extended community "color:00:100" and extended community "target:64500:1" to the VPN-IPv4 routes.

```

# on PE-1:
configure {
  policy-options {
    community "Color100_com" {
      member "color:00:100" { }
    }
  }
}

```



```
Flag: 0xc0 Type: 16 Len: 16 Extended Community:
target:64500:1
color:00:100
"
```

The VPN-IPv4 routes are using the SR policy tunnel to PE-7, as follows:

```
[/]
A:admin@PE-1# show router bgp next-hop vpn-ipv4
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
BGP VPN Next Hop
=====
VPN Next Hop      Owner
Autobind          FibProg Reason
Labels            FlexAlgo Metric
Admin-tag-policy (strict-tunnel-tagging)
-----
192.0.2.7          SR-POLICY
  bgp sr-policy    Y
  --              -- 0
  -- (-)
-----
Next Hops : 1
=====
```

In the route table of VPRN 1 on PE-1, routes to VPRN 1 on PE-7 use the SR policy tunnel toward PE-7, as follows:

```
[/]
A:admin@PE-1# show router 1 route-table
=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]      Type  Proto  Age      Pref
Next Hop[Interface Name] Metric
-----
172.16.1.1/32           Local Local  01h12m28s 0
  system                0
172.16.7.1/32           Remote BGP VPN 00h17m00s 170
  192.0.2.7 (tunneled:SR-Policy:917506) 0
172.31.1.0/24           Local Local  01h12m28s 0
  lol                    0
172.31.7.0/24           Remote BGP VPN 00h17m00s 170
  192.0.2.7 (tunneled:SR-Policy:917506) 0
-----
No. of Routes: 4
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```



Note:

Color-only steering can be achieved without the need to match the BGP next-hop endpoint. This is done by setting the "color-only" (CO) bits, which are the two highest order bits of the color extended community. When set to "10" or "01" instead of "00", the **endpoint** value in the SR

policy can be set to "0.0.0.0". In this case, only the color value is checked as a single condition to steer traffic flows using the SR policy.

Show and Debug commands

The following command shows the SR policies on PE-1, with the segment list, color, endpoint address, and so on:

```
[/]
A:admin@PE-1# show router segment-routing sr-policies all

=====
SR-Policies Path
=====
-----
Active           : Yes                Owner           : static
Color            : 100
Head             : 0.0.0.0          Endpoint Addr   : 192.0.2.7
RD               : 64500           Preference      : 100
BSID             : 20000
TunnelId         : 917506          Age             : 218
Origin ASN       : 0              Origin          : 0.0.0.0
NumReEval        : 0              ReEvalReason    : none
NumActPathChange: 0              Last Change     : 09/20/2021 14:18:28
Maintenance Policy: N/A

Path Segment Lists:
Segment-List     : 1                Weight          : 1
S-BFD State      : Down            S-BFD Transitio*: 0
Num Segments     : 2                Last Change     : 09/20/2021 13:31:51
  Seg 1 Label    : 32003            State           : resolved-up
  Seg 2 Label    : 20099            State           : N/A

=====
* indicates that the corresponding row element may have been truncated.
```

For each combination of color and endpoint, the SR database must validate each segment list / candidate path, and choose one to be the active path. The most important checks are:

- The configured BSID is part of the SRLB. In this example, 20000 is part of the SRLB1 block on PE-1, which ranges from 20000 to 21999. The BSID is used uniquely by this policy.
- The first segment of each segment list is resolved to a set of one or more next-hops. This means matching an SR-ISIS or SR-OSPF node SID, matching an SR-ISIS or SR-OSPF adjacency SID, or matching an SR-ISIS or SR-OSPF adjacency-set SID. In this example, on PE-1, label 32003 resolves to the SR-ISIS node SID of ASBR-3.

The following command shows the MPLS forwarding policy on ASBR-3, including the binding label, NHG, and pushed labels. If, for example, the reserved label block was not defined or not referenced by the MPLS forwarding policy, the validation would fail and the MPLS forwarding policy would remain operationally down.

```
[/]
A:admin@ASBR-3# show router mpls forwarding-policies forwarding-policy detail

=====
Forwarding Policy (Detail)
=====
-----
```

```

Policy : SLR-ILM-pushed-label
-----
Admin State      : Up                Oper State       : Up
Binding Label   : 20099             Preference        : 255
Revert Timer      : 5 sec
Last Change       : 09/20/2021 14:19:00
Ingress Stats     : Disabled
Metric            : 0                  Tunnel Table Pref: 255
Endpoint Address  : N/A

Next-hop Group  : 1
Admin State       : Up                Oper State        : Up
Resolution Type   : direct            Load Balancing Wt: 0
Last Change       : 09/20/2021 14:19:00
Primary
NH Address        : 192.168.35.2
Oper State        : Up                Last Change       : 09/20/2021 14:19:00
Pushed Labels   : 20100
Backup
NH Address        : 192.168.36.2
Oper State        : Up                Last Change       : 09/20/2021 14:19:00
Pushed Labels   : 20100
=====

```

The following command shows the details of the MPLS binding label forwarding policy:

```

[/]
A:admin@ASBR-3# show router mpls forwarding-policies binding-label detail

=====
Binding Label (Detail)
=====
Label          : 20099                Preference     : 255
Policy Name    : SLR-ILM-pushed-label
Oper State     : Up                    OperDownReason : notApplicable
Up Time       : 09/20/2021 14:07:03   NumNextHopGrps : 1
Ingress Stats : Disabled              IngrOperState  : Down
Egress Stats  : Disabled              EgrOperState   : Down
Revert Timer   : 5
Retry Count    : 0                    Next Retry In  : 0

Next-hop Group : 1                    Resolution Type: direct
Oper State     : Up                    OperDownReason : notApplicable
Num Revert     : 0                    Num Failover   : 0
Next Revert In : 0
Primary nexthop: 192.168.35.2
Resolved      : True                  NHopDownReason : notApplicable
EgrOperState  : Down
Pushed Labels  : 20100
Backup nexthop: 192.168.36.2
Resolved      : True                  NHopDownReason : notApplicable
EgrOperState  : Down
Pushed Labels  : 20100
-----
=====

```

The following tools command on ASBR-3 shows that the MPLS forwarding policy is validated and the ILM is programmed with the binding label value. The output also shows which router interfaces are used toward the configured next-hops and what label stack is pushed.

```

[/]
A:admin@ASBR-3# tools dump router mpls forwarding-policies binding-label 20099

```

```
Db Mgr flags 0x80 ilmStatsFailCnt 0
-----
dbOwner FWD PLCY routeOwner 48 rsvdBlkId 2 flags 0x3 numPolicies 1 numInstalled 1
-----

Label DB 20099
dbFlags 0xd PathCount 1 srTunnelId 851970 ilmStatsIdx[MGMT] 0x0 ilmStatsIdx[API] 0x0
LABEL RESERVED: PROGRAMMED
Path bitmap 0
Label Retry time left : 0 retrycount : 0, SR Retry time left : 0 SR retrycount : 0

Best Db Path owner 0 path name vrId:1, dbOwner:0, pathName:SLR-ILM-pushed-label Last Modified
09/20/2021 14:19:00 Up Time 0d 00:33:31
Preference 255 flags 0x245 Status FWDPLCY_ERR_NA SR status SR_ERR_OK
PrimResolved NH's 1 BkupResolved NH's 1
NHGroup 1
flags 0x3bf9 : weight 0 normalized weight 0
Revert timer 5 Time left 0 NumOfReverts 0

Hold timer 0 Time left 0
DIRECT NH: PRIM PGMED: PRIM RESOLVED: BKUP RESOLVED: BKUP PGMED:
primaryNH 192.168.35.2 egrStatsIdx 0x0 Status FWDPLCY_NHERR_NA
Label Stack:20100 0
Nexthop 1 192.168.35.2 outIf 3 globalIfIndex 2 globaIfInNHgrp 2
PG ID 4
PG ID 0
backupNH 192.168.36.2 egrStatsIdx 0x0 Status FWDPLCY_NHERR_NA
Label stack:20100 0
Nexthop 1 192.168.36.2 outIf 4 globalIfIndex 3 globaIfInNHgrp 3
PG ID 5
PG ID 0
-----
```

Conclusion

MPLS forwarding policies provide the customization of next-hops as well as ECMP, weighted ECMP, Class-Based Forwarding, and backup support. MPLS forwarding policies can be combined with SR policies.

Parallel Adjacency Sets in Segment Routing

This chapter describes the Parallel Adjacency Sets in Segment Routing.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information and configuration in this chapter are based on SR OS Release 21.7.R1. They apply for MD-CLI.

Overview

SR OS supports segment routing as described in RFC 8402, *Segment Routing Architecture*. In the remainder of this chapter, SR refers to "Segment Routing", unless specified otherwise. Product and release references, such as 7750 SR and SR OS, continue to refer to "Service Router".

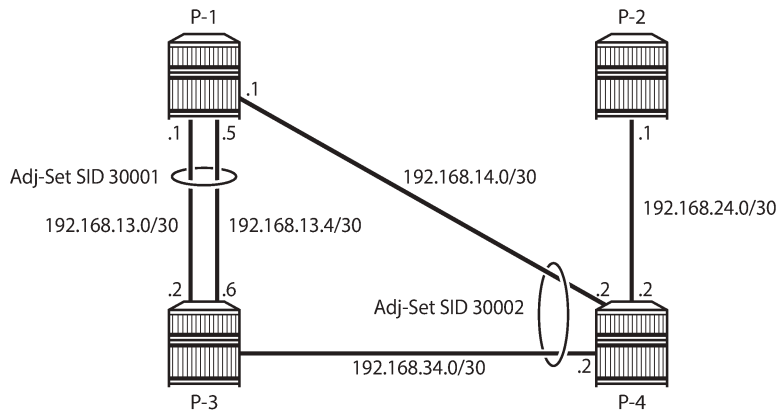
SR provides operators the means to provision paths or tunnels, encoded as a sequential list of sub-paths or segments without requiring a dedicated signaling protocol, by advertising the identities of the segments across the SR domain using extensions to the link state Interior Gateway Protocols (IGPs), such as IS-IS and OSPF.

When defining source-routed traffic-engineered end-to-end SR paths, routing constraints such as loose and strict hops can be used to control the data path through a network; a node SID is used for a loose hop, and an adjacency SID is used for a strict hop. See the [Segment Routing – Traffic Engineered Tunnels](#) chapter for more information.

Parallel links between adjacent nodes can be grouped into adjacency sets, and a single adjacency set is identified using a locally significant adjacency set SID. Traffic can be load shared across the links in the set and is based on traffic flow identifiers; for example, source and destination IP addresses, and entropy label.

In [Figure 333: Parallel and non-parallel adjacency sets](#), two adjacency sets are defined. A first adjacency set is defined on P-1 with adjacency set SID 30001. Two parallel links are available between P-1 and P-2, and by combining them into an adjacency set, traffic can be shared across both links. A second set is defined on P-4, with adjacency set SID 30002. However, the member links of that set are not terminated on the same router pair, so traffic cannot be shared.

Figure 333: Parallel and non-parallel adjacency sets

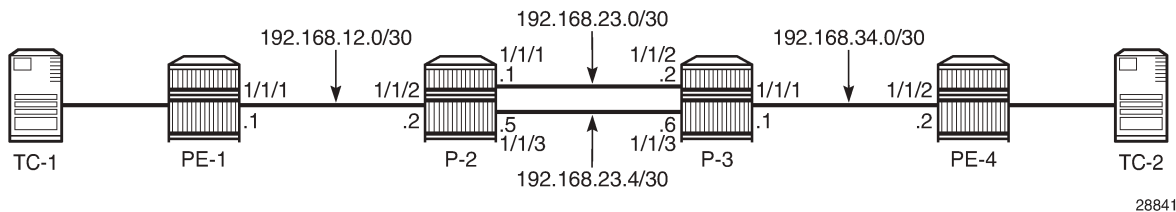


28840

Configuration

The topology used in this chapter is shown in [Figure 334: Parallel adjacency set](#). All nodes are configured for SR and IS-IS level 2. If test center TC-1 is connected at PE-1 and test center TC-2 is connected at PE-4, traffic can be sent from TC-1 to TC-2 following the PE-1, P-2, P-3, PE-4 path. Two links are active between P-2 and P-3, and these links belong to the same adjacency set.

Figure 334: Parallel adjacency set



28841

The initial configuration on the PE nodes includes the following:

- Cards, MDAs, ports
- Router interfaces
- IS-IS

Segment routing configuration

In the topology from [Figure 334: Parallel adjacency set](#), all nodes are configured with a common Segment Routing Global Block (SRGB), which is defined as follows:

```
configure {
  router "Base" {
    mpls-labels {
      sr-labels {
        start 20000
      }
    }
  }
}
```

```

    end 20099
  }
}
}
}

```

In this example, prefix SID allocation is using global mode, and the node SIDs are defined by index on the system interfaces in the **isis** context, where PE-1, P-2, P-3, and PE-4 take the indices 1, 2, 3, and 4, respectively. The **advertise-router-capability area** command enables the IS-IS extensions so that the SID values are advertised throughout the SR domain. The configuration on PE-1 is as follows; the configuration on the other nodes is similar.

```

# on PE-1
configure {
  router "Base" {
    isis 0 {
      admin-state enable
      advertise-router-capability area
      level-capability 2
      traffic-engineering true
      area-address [49.0001]
      segment-routing {
        admin-state enable
        prefix-sid-range {
          global
        }
      }
      interface "int-PE-1-P-2" {
        interface-type point-to-point
      }
      interface "system" {
        ipv4-node-sid {
          index 1
        }
      }
    }
  }
}
}

```

With this configuration, each node floods the SIDs in link state packets (shown as "LSP") across the domain. For P-2, prefix 192.0.2.2 has index 2 in the SRGB. The adjacency SIDs 524285, 524286, and 524287 are taken from the dynamic range, as follows:

```

[/]
A:admin@P-2# show router isis database P-2.00-00 detail

=====
Rtr Base ISIS Instance 0 Database (detail)
=====

Displaying Level 1 database
-----
Level (1) LSP Count : 0

Displaying Level 2 database
-----
LSP ID   : P-2.00-00          Level    : L2
Sequence : 0x3                Checksum : 0xeb4b    Lifetime : 1103
Version  : 1                  Pkt Type  : 20       Pkt Ver  : 1
Attributes: L1L2             Max Area  : 3         Alloc Len : 1492
SYS ID   : 1920.0000.2002    SysID Len : 6         Used Len  : 330

```

```
TLVs :
Area Addresses:
  Area Address : (3) 49.0001
Supp Protocols:
  Protocols    : IPv4
IS-Hostname   : P-2
Router ID     :
  Router ID    : 192.0.2.2
Router Cap    : 192.0.2.2, D:0, S:0
  TE Node Cap : B E M P
  SR Cap: IPv4 MPLS-IPv6
    SRGB Base:20000, Range:100
  SR Alg: metric based SPF
  Node MSD Cap: BMI : 12 ERLD : 15
IS Neighbors  :
  Virtual Flag : 0
  Default Metric: (I) 10
  Delay Metric : (I) 0
  Expense Metric: (I) 0
  Error Metric : (I) 0
  Neighbor     : PE-1.00
IS Neighbors  :
  Virtual Flag : 0
  Default Metric: (I) 10
  Delay Metric : (I) 0
  Expense Metric: (I) 0
  Error Metric : (I) 0
  Neighbor     : P-3.00
Internal Reach:
  Default Metric: (I) 10
  Delay Metric : (I) 0
  Expense Metric: (I) 0
  Error Metric : (I) 0
  IP Address   : 192.168.12.0
  IP Mask      : 255.255.255.252
  Default Metric: (I) 10
  Delay Metric : (I) 0
  Expense Metric: (I) 0
  Error Metric : (I) 0
  IP Address   : 192.168.23.0
  IP Mask      : 255.255.255.252
  Default Metric: (I) 0
  Delay Metric : (I) 0
  Expense Metric: (I) 0
  Error Metric : (I) 0
  IP Address   : 192.0.2.2
  IP Mask      : 255.255.255.255
  Default Metric: (I) 10
  Delay Metric : (I) 0
  Expense Metric: (I) 0
  Error Metric : (I) 0
  IP Address   : 192.168.23.4
  IP Mask      : 255.255.255.252
I/F Addresses :
  I/F Address  : 192.168.23.1
  I/F Address  : 192.0.2.2
  I/F Address  : 192.168.12.2
  I/F Address  : 192.168.23.5
TE IS Nbrs   :
  Nbr         : PE-1.00
  Default Metric : 10
  Sub TLV Len  : 19
  IF Addr     : 192.168.12.2
```

```

Nbr IP      : 192.168.12.1
Adj-SID: Flags:v4VL Weight:0 Label:524287
TE IS Nbrs :
Nbr        : P-3.00
Default Metric : 10
Sub TLV Len  : 26
IF Addr     : 192.168.23.1
Nbr IP      : 192.168.23.2
Adj-SID: Flags:v4VL Weight:0 Label:524286
Adj-SID: Flags:v4VLSP Weight:0 Label:30000
TE IS Nbrs :
Nbr        : P-3.00
Default Metric : 10
Sub TLV Len  : 26
IF Addr     : 192.168.23.5
Nbr IP      : 192.168.23.6
Adj-SID: Flags:v4VL Weight:0 Label:524285
Adj-SID: Flags:v4VLSP Weight:0 Label:30000
TE IP Reach :
Default Metric : 10
Control Info:   , prefLen 30
Prefix        : 192.168.12.0
Default Metric : 10
Control Info:   , prefLen 30
Prefix        : 192.168.23.0
Default Metric : 0
Control Info:  S, prefLen 32
Prefix        : 192.0.2.2
Sub TLV       :
Prefix-SID Index:2, Algo:0, Flags:NnP
Default Metric : 10
Control Info:   , prefLen 30
Prefix        : 192.168.23.4

Level (2) LSP Count : 1
-----
Control Info      : D = Prefix Leaked Down
                  S = Sub-TLVs Present
Attribute Flags   : N = Node Flag
                  R = Re-advertisement Flag
                  X = External Prefix Flag
                  E = Entropy Label Capability (ELC) Flag
Adj-SID Flags     : v4/v6 = IPv4 or IPv6 Address-Family
                  B = Backup Flag
                  V = Adj-SID carries a value
                  L = value/index has local significance
                  S = Set of Adjacencies
                  P = Persistently allocated
Prefix-SID Flags  : R = Re-advertisement Flag
                  N = Node-SID Flag
                  nP = no penultimate hop POP
                  E = Explicit-Null Flag
                  V = Prefix-SID carries a value
                  L = value/index has local significance
Lbl-Binding Flags: v4/v6 = IPv4 or IPv6 Address-Family
                  M = Mirror Context Flag
                  S = SID/Label Binding flooding
                  D = Prefix Leaked Down
                  A = Attached Flag
SABM-flags Flags: R = RSVP-TE
                  S = SR-TE
                  F = LFA
                  X = FLEX-ALGO
FAD-flags Flags:  M = Prefix Metric

```

Adjacency set configuration

Adjacency set SIDs are allocated from a reserved label block. Because the adjacency SIDs have a local significance only, the same block can be defined on each node. In this example, a different label block is defined on P-2 and P-3 respectively, as follows. The start-label and end-label values must be in the dynamic range.

```
# on P-2
configure {
  router "Base" {
    mpls-labels {
      reserved-label-block "adjset_block_on_P-2" {
        start-label 30000
        end-label 30099
      }
    }
  }
}

# on P-3
configure {
  router "Base" {
    mpls-labels {
      reserved-label-block "adjset_block_on_P-3" {
        start-label 40000
        end-label 40099
      }
    }
  }
}
```

This range is listed in the **show router mpls-labels label-range** command, as follows:

```
[/]
A:admin@P-2# show router mpls-labels label-range

=====
Label Ranges
=====
Label Type      Start Label  End Label  Aging    Available  Total
-----
Static          32           18431     -        18400     18400
Dynamic         18432        524287    0        505653    505856
  Seg-Route     20000        20099     -         0         100
-----

Reserved Label Blocks
-----
Reserved Label  Start      End      Total
Block Name     Label     Label
-----
adjset_block_on_P-2    30000    30099    100
-----

No. of Reserved Label Blocks: 1
=====
```

The reserved label block range is then defined as a Segment Routing Local Block (SRLB) in the segment-routing context. Label values for adjacency sets must be allocated from the SRLB; otherwise, an error is raised. The adjacency set is identified by number, and on P-2 adjacency set 1 has a SID label value of 30000. A similar configuration is used on P-3. If no SID label value is configured, the system will allocate a value from the SRLB range.

```
# on P-2
configure {
  router "Base" {
    isis 0 {
      segment-routing {
        admin-state enable
        srlb "adjset_block_on_P-2"
        adjacency-set 1 {
          sid {
            label 30000
          }
        }
      }
    }
  }
}

# on P-3
configure {
  router "Base" {
    isis 0 {
      segment-routing {
        admin-state enable
        srlb "adjset_block_on_P-3"
        adjacency-set 1 {
          sid {
            label 40000
          }
        }
      }
    }
  }
}
```

On P-2, the *int-P-2-P-3-a* and *int-P-2-P-3-b* interfaces have addresses 192.168.23.1/30 and 192.168.23.5/30, respectively, and these interfaces are included in adjacency set 1 by applying the adjacency set index to the individual interfaces, as follows. A similar configuration is present on P-3.

```
configure {
  router "Base" {
    isis 0 {
      interface "int-P-2-P-3-a" {
        interface-type point-to-point
        adjacency-set 1 { }
      }
      interface "int-P-2-P-3-b" {
        interface-type point-to-point
        adjacency-set 1 { }
      }
      interface "int-P-2-PE-1" {
        interface-type point-to-point
      }
      interface "system" {
        ipv4-node-sid {
          index 2
        }
      }
    }
  }
}
```



```
IP Mask      : 255.255.255.252
Default Metric: (I) 10
Delay Metric : (I) 0
Expense Metric: (I) 0
Error Metric  : (I) 0
IP Address   : 192.168.23.0
IP Mask      : 255.255.255.252
Default Metric: (I) 0
Delay Metric : (I) 0
Expense Metric: (I) 0
Error Metric  : (I) 0
IP Address   : 192.0.2.2
IP Mask      : 255.255.255.255
Default Metric: (I) 10
Delay Metric : (I) 0
Expense Metric: (I) 0
Error Metric  : (I) 0
IP Address   : 192.168.23.4
IP Mask      : 255.255.255.252
I/F Addresses :
I/F Address  : 192.168.23.1
I/F Address  : 192.0.2.2
I/F Address  : 192.168.12.2
I/F Address  : 192.168.23.5
TE IS Nbrs   :
Nbr          : PE-1.00
Default Metric : 10
Sub TLV Len   : 19
IF Addr      : 192.168.12.2
Nbr IP       : 192.168.12.1
Adj-SID: Flags:v4VL Weight:0 Label:524287
TE IS Nbrs   :
Nbr          : P-3.00
Default Metric : 10
Sub TLV Len   : 26
IF Addr      : 192.168.23.1
Nbr IP       : 192.168.23.2
Adj-SID: Flags:v4VL Weight:0 Label:524286
Adj-SID: Flags:v4VLSP Weight:0 Label:30000
TE IS Nbrs   :
Nbr          : P-3.00
Default Metric : 10
Sub TLV Len   : 26
IF Addr      : 192.168.23.5
Nbr IP       : 192.168.23.6
Adj-SID: Flags:v4VL Weight:0 Label:524285
Adj-SID: Flags:v4VLSP Weight:0 Label:30000
TE IP Reach  :
Default Metric : 10
Control Info:   , prefLen 30
Prefix       : 192.168.12.0
Default Metric : 10
Control Info:   , prefLen 30
Prefix       : 192.168.23.0
Default Metric : 0
Control Info:  S, prefLen 32
Prefix       : 192.0.2.2
Sub TLV      :
Prefix-SID Index:2, Algo:0, Flags:NnP
Default Metric : 10
Control Info:   , prefLen 30
Prefix       : 192.168.23.4
```

Level (2) LSP Count : 1


```

-----
Control Info      : D = Prefix Leaked Down
                  S = Sub-TLVs Present
Attribute Flags  : N = Node Flag
                  R = Re-advertisement Flag
                  X = External Prefix Flag
                  E = Entropy Label Capability (ELC) Flag
Adj-SID Flags    : v4/v6 = IPv4 or IPv6 Address-Family
                  B = Backup Flag
                  V = Adj-SID carries a value
                  L = value/index has local significance
                  S = Set of Adjacencies
                  P = Persistently allocated
Prefix-SID Flags : R = Re-advertisement Flag
                  N = Node-SID Flag
                  nP = no penultimate hop POP
                  E = Explicit-Null Flag
                  V = Prefix-SID carries a value
                  L = value/index has local significance
Lbl-Binding Flags: v4/v6 = IPv4 or IPv6 Address-Family
                  M = Mirror Context Flag
                  S = SID/Label Binding flooding
                  D = Prefix Leaked Down
                  A = Attached Flag
SABM-flags Flags: R = RSVP-TE
                  S = SR-TE
                  F = LFA
                  X = FLEX-ALGO
FAD-flags Flags:  M = Prefix Metric
=====

```

SR traffic engineered label switched path configuration

For traffic from PE-1 to PE-4 to use the adjacency set between P-2 and P-3, a label switched path is required. This path can be defined using SR policies or using SR traffic engineered (SR-TE) tunnels (see the [Segment Routing – Traffic Engineered Tunnels](#) chapter).

This chapter uses SR-TE tunnels, with label switched path *lsp-adj-set* using *path-adj-set* as the primary path. A loose hop translates to a node SID for that hop. A strict hop translates to an adjacency set SID, if an adjacency set is available. If no adjacency set is configured, an adjacency SID is used. The MPLS configuration on PE-1 is as follows; the configuration on PE-4 is similar.

```

# on PE-1
configure {
  router "Base" {
    mpls {
      admin-state enable
      path "path-adj-set" {
        admin-state enable
        hop 1 {
          ip-address 192.0.2.2
          type loose
        }
        hop 2 {
          ip-address 192.0.2.3
          type strict
        }
        hop 3 {
          ip-address 192.0.2.4
          type loose
        }
      }
    }
  }
}

```



```

PCE Control      : Disabled          Oper PCE Control : Disabled
Include Groups   :                   Oper IncludeGroups:
None                                                     None
Exclude Groups   :                   Oper ExcludeGroups:
None                                                     None
Last Resignal    : n/a

IGP/TE Metric    : 16777215          Oper Metric       : 16777215
Oper MTU         : 1552              Path Trans       : 1
Degraded         : False
Failure Code     : noError
Failure Node     : n/a
Explicit Hops    :
                  192.0.2.2(L)
                  -> 192.0.2.3(S)
                  -> 192.0.2.4(L)
Actual Hops      :
    192.0.2.2(192.0.2.2) (N-SID)      Record Label      : 20002
-> 192.0.2.3(192.0.2.3) (A-SID)      Record Label      : 30000
-> 192.0.2.4(192.0.2.4) (N-SID)      Record Label      : 20004

BFD Configuration and State
Template         : None              Ping Interval     : N/A
Enable          : False             State             : notApplicable
WaitForUpTimer  : 4 sec             OperWaitForUpTimer: 0 sec
WaitForUpTmLeft : 0
StartFail Rsn   : N/A
=====

```

Service configuration

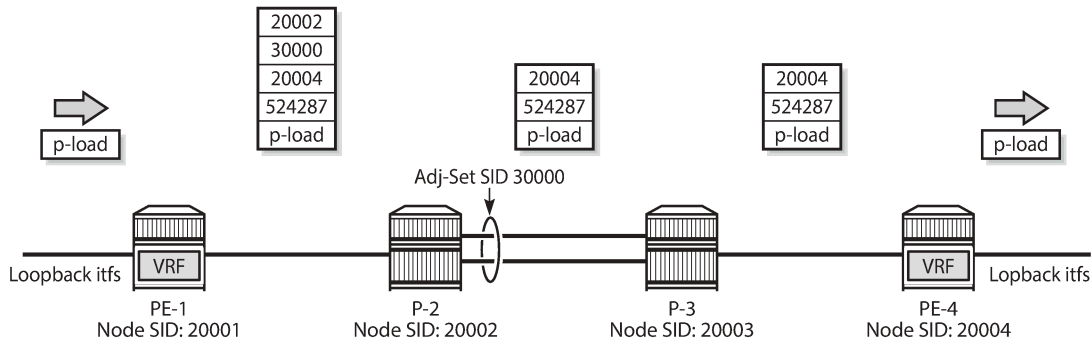
A VPRN service is configured on PE-1 and PE-4, providing multiple loopback interfaces that simulate the TCs. This VPRN is configured to use the SR-TE tunnel defined in the previous section. The configuration on PE-1 is as follows; the configuration on PE-4 is similar.

```

# on PE-1
configure {
  service {
    vprn "svc-1" {
      admin-state enable
      description "runs between PE-1 and PE-4"
      service-id 1
      customer "1"
      autonomous-system 64496
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "64496:1"
          vrf-target {
            community "target:64496:1"
          }
          auto-bind-tunnel {
            resolution filter
            resolution-filter {
              sr-te true
            }
          }
        }
      }
    }
  }
}

```


Figure 335: MPLS label stack



28842

The traffic that is sent in this example is a burst of successive pings (8000) in multiple flows (5) from a loopback interface on PE-1 to the different loopback interfaces on PE-4. So, the traffic flows have a variety of source/destination IP-address pairs. Additionally, for the load to be sprayed across the adjacency set members, load balancing must be enabled. On P-2, this is enabled as follows:

```
# on P-2
configure system load-balancing lsr-load-balancing lbl-ip
```

P-2 hashes the traffic (ping requests) based on the source and destination IP addresses, thereby spraying the traffic across the *int-P-2-P-3-a* interface (on port 1/1/1) and the *int-P-2-P-3-b* interface (on port 1/1/3). P-3 hashes the return traffic (ping responses) similarly across the *int-P-3-P-2-a* interface (on port 1/1/2) and the *int-P-3-P-2-b* interface (on port 1/1/3). Because two links are available, both carry a part of the traffic, as follows. Only the monitoring outcome for P-2 is shown; P-3 has a corresponding monitoring outcome.

```
A:P-2# monitor port 1/1/2 1/1/1 1/1/3 interval 5 repeat 25 absolute
```

```
=====
Monitor statistics for Ports
=====
Input                               Output
-----
At time t = 0 sec (Base Statistics)
-----
Port 1/1/2
-----
Octets                               11416                               12136
Packets                              124                                124
Errors                                0                                  0

Port 1/1/1
-----
Octets                               11969                               11703
Packets                              125                                124
Errors                                0                                  0

Port 1/1/3
-----
Octets                               10102                               10102
Packets                              99                                 99
Errors                                0                                  0
```

```
-----  
At time t = 5 sec (Mode: Absolute)  
-----  
Port 1/1/2  
-----  
Octets          75846          72206  
Packets         670           670  
Errors          0             0  
  
Port 1/1/1  
-----  
Octets          44492          51523  
Packets         421           486  
Errors          0             0  
  
Port 1/1/3  
-----  
Octets          37859          30626  
Packets         352           286  
Errors          0             0  
  
-----  
---snip---  
-----  
At time t = 120 sec (Mode: Absolute)  
-----  
Port 1/1/2  
-----  
Octets          4734681         4415401  
Packets         40159          40159  
Errors          0             0  
  
Port 1/1/1  
-----  
Octets          1775114         2654968  
Packets         16159          24159  
Errors          0             0  
  
Port 1/1/3  
-----  
Octets          2652795         1772795  
Packets         24127          16127  
Errors          0             0  
  
-----  
At time t = 125 sec (Mode: Absolute)  
-----  
Port 1/1/2  
-----  
Octets          4734964         4415538  
Packets         40161          40161  
Errors          0             0  
  
Port 1/1/1  
-----  
Octets          1775371         2655251  
Packets         16162          24161  
Errors          0             0  
  
Port 1/1/3  
-----
```

```

Octets                2653262                1773262
Packets            24131                 16131
Errors                0                  0
    
```

```
=====
A:P-2#
```

The relevant information is available after monitoring all bursts (after 125 seconds): 3 out of 5 flows use the *int-P-2-P-3-a* interface; 2 out of 5 flows use the *int-P-2-P-3-b* interface.

With an additional burst to a loopback interface that is reached above over the *int-P-2-P-3-a* interface, 4 (=3+1) out of now 6 flows use the *int-P-2-P-3-a* interface; the initial 2 out of 6 flows keep on using the *int-P-2-P-3-b* interface, as follows:

```
A:P-2# monitor port 1/1/2 1/1/1 1/1/3 interval 5 repeat 25 absolute
```

```
=====
Monitor statistics for Ports
=====
```

	Input	Output
-----snip-----		
At time t = 125 sec (Mode: Absolute)		

Port 1/1/2		
Octets	5675055	5291655
Packets	48120	48119
Errors	0	0

Port 1/1/1		
Octets	2651441	3531342
Packets	24120	32120
Errors	0	0

Port 1/1/3		
Octets	2649909	1769836
Packets	24097	16096
Errors	0	0

```
=====
A:P-2#
```

With a further additional burst to a loopback interface that is reached above over the *int-P-2-P-3-b* interface, the initial 4 out of now 7 flows keep on using the *int-P-2-P-3-a* interface; 3 (=2+1) out of 7 flows use the *int-P-2-P-3-b* interface, as follows:

```
A:P-2# monitor port 1/1/2 1/1/1 1/1/3 interval 5 repeat 25 absolute
```

```
=====
Monitor statistics for Ports
=====
```

	Input	Output
-----snip-----		

```
-----  
At time t = 125 sec (Mode: Absolute)  
-----  
Port 1/1/2  
-----  
Octets                6618935                6171269  
Packets               56117                 56115  
Errors                 0                      0  
  
Port 1/1/1  
-----  
Octets                2651175                3531102  
Packets             24117                 32116  
Errors                 0                      0  
  
Port 1/1/3  
-----  
Octets                3529677                2649844  
Packets             32094                 24095  
Errors                 0                      0  
  
=====
```

A:P-2#

Conclusion

By defining adjacency sets in SR-enabled networks, operators can apply load sharing to parallel links between adjacent nodes, thereby optimizing the use of network resources.

Seamless BFD for SR-TE LSPs

This chapter describes seamless BFD for SR-TE LSPs.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written based on SR OS Release 19.10.R1, but the configuration in the current edition is based on SR OS Release 23.3.R3. BFD for RSVP-TE LSPs is supported in SR OS Release 13.0, and later. Seamless BFD for SR-TE LSPs is supported in SR OS Release 19.10.R1, and later.

Overview

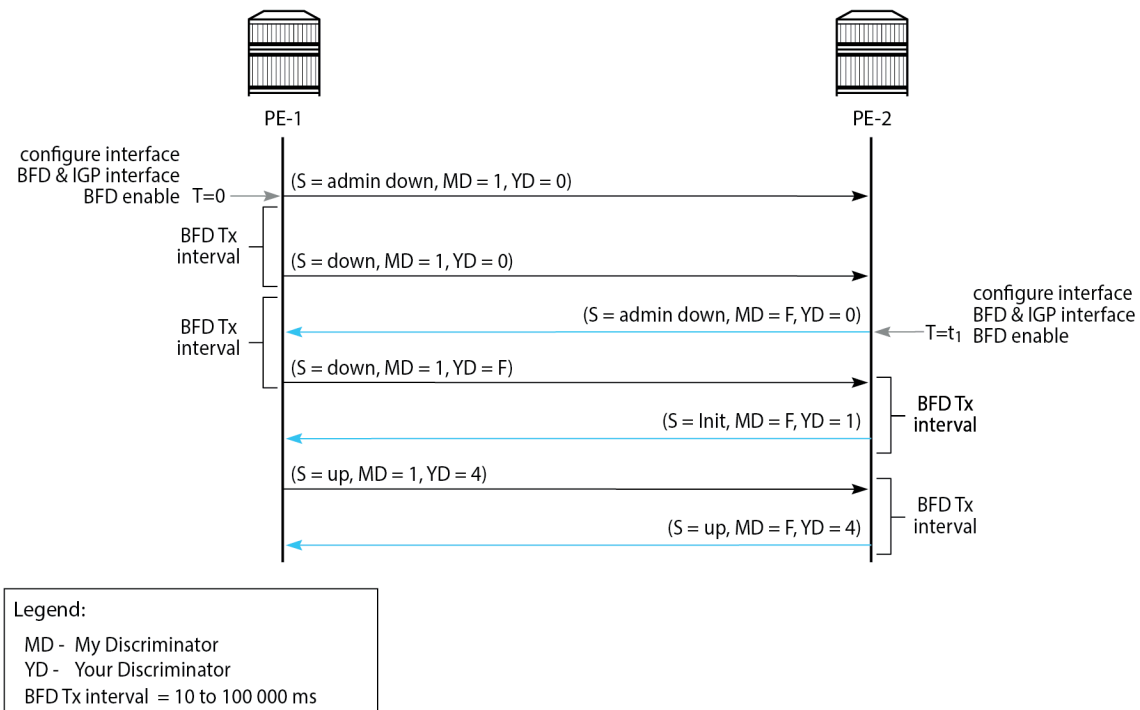
Bidirectional Forwarding Detection (BFD) is widely deployed in IP/MPLS networks to rapidly detect failures in the forwarding path between network elements. In this chapter, a comparison is made between classical BFD and seamless BFD (S-BFD).

Classical BFD

Classical BFD, described in RFC 5880, requires little overhead. However, the handshake mechanism to negotiate and set up two-way BFD sessions between network elements can take several seconds. RFC 5880 specifies two modes of operation: asynchronous mode and on-demand mode. Additionally, the BFD echo function loops back BFD echo packets to the sender.

Classical BFD is applied to the interface. In asynchronous mode, sessions are established. Network elements periodically send BFD control packets to one another. Discriminators are used as a session demultiplexer to distinguish between BFD sessions. The transmitting network element generates a unique non-zero discriminator value, which is exchanged as part of the session handshake establishment. [Figure 336: Classical BFD handshake](#) shows the classical BFD handshake for a single hop across an IP link.

Figure 336: Classical BFD handshake



35626

BFD for MPLS LSPs

BFD is supported for RSVP-TE LSPs and for LDP LSPs, as described in the [BFD for RSVP-TE and LDP LSPs](#) chapter.

BFD for MPLS LSPs is described in RFC 5884. For continuity checks in MPLS LSPs, BFD packets are transmitted using the MPLS encapsulation, so they share fate with the LSP data path.

BFD is bootstrapped using an LSP ping. An MPLS echo request packet is transmitted along the LSP path, including a BFD discriminator TLV containing the head-end BFD discriminator value. The tail end responds with an echo reply packet, using the IP forwarding path, including the tail-end BFD discriminator value.

Afterward, BFD control packets establish a BFD session between the head end and tail end using the discriminator values from the bootstrap session. The egress LER will send a BFD control packet upon receipt.

Each session has its own pair of discriminators, so multiple discriminators are allocated by the system.

S-BFD for SR-TE LSPs

S-BFD is described in RFC 7880. Unlike classical BFD, S-BFD does not rely on the BFD bootstrapping process (handshake) or session state at the tail end of a session. Instead, when S-BFD is initialized, a pair of discriminators are selected by the system for specific purposes (reflector or initiator). S-BFD minimizes the time required to establish BFD sessions, which contributes to its seamless operation. S-BFD relies

on the fact that the discriminators are already known by the endpoints for each session, either through configuration or advertisement using unicast protocols.

There are two discriminators, one for each end of the BFD/S-BFD session. From the perspective of the S-BFD initiator (or BFD head end) there is a local 'my discriminator' and a remote 'your discriminator'. The 'your discriminator' matches the remote node's local discriminator, which for BFD is allocated to the session endpoint, and for S-BFD is the reflector discriminator.

Terminology

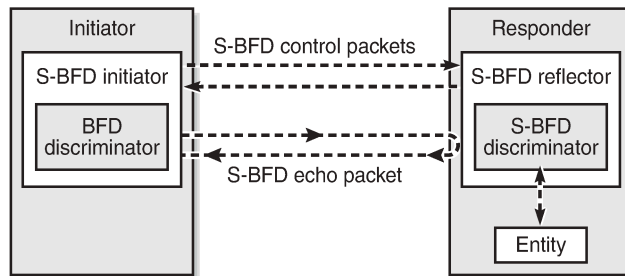
[Table 19: RFC 7880 S-BFD terms](#) describes the S-BFD terms, as defined by RFC 7880.

Table 19: RFC 7880 S-BFD terms

S-BFD term	Description
Entity	A function on a network node to which the S-BFD mechanism allows remote network nodes to perform continuity tests. An entity can be abstract (for example, reachability) or specific (for example, IP addresses, router IDs, functions).
S-BFD initiator	An S-BFD session on a network node that performs a continuity test to a remote entity by sending S-BFD packets.
BFD discriminator	An identifier for a BFD session at an endpoint of a BFD session.
Initiator	A network node hosting an S-BFD initiator.
S-BFD reflector	An S-BFD session on a network node that listens for incoming S-BFD control packets to local entities and generates response S-BFD control packets.
Responder	A network node hosting an S-BFD reflector.
S-BFD discriminator	A BFD discriminator allocated for an endpoint of an S-BFD session.

[Figure 337: Relationship between S-BFD terms](#) shows the relationship between the S-BFD terms described in [Table 19: RFC 7880 S-BFD terms](#).

Figure 337: Relationship between S-BFD terms



35628

S-BFD implementation in SR OS

Before an application can request the establishment of an S-BFD session, a mapping table of remote discriminators to peer far-end IP addresses must exist. These correspond to the discriminators of the reflector nodes. The mapping can be accomplished in two ways:

- automatically learned (using opaque OSPF or IS-IS routing extensions) or
- statically configured

A single S-BFD discriminator is allocated to a reflector in a router instance. The local reflector S-BFD discriminator is statically configured in the CLI and must be in the range from 524288 to 526335. The S-BFD discriminator must not be the same as any discriminator used for classical BFD.

As per RFC 5884, the destination IP address of explicitly label-switched S-BFD control packets must be chosen from the 127/8 range for IPv4 and the TTL of the IP header must be set to 1. The source IP address is a routable address of the sender.

The initiator node uses the following UDP ports for S-BFD control packets:

- UDP destination port 7784
- UDP source port, which can be any valid port except 7784, as follows:
 - the same UDP source port for all S-BFD control packets to the same reflector
 - different UDP source ports for S-BFD control packets to different reflectors
 - packets with UDP source port 7784 will be discarded by the reflector

The responder node swaps the UDP source and destination port when sending S-BFD control packets back to the initiator node:

- received UDP source port = transmitted UDP destination port
- received UDP destination port = transmitted UDP source port

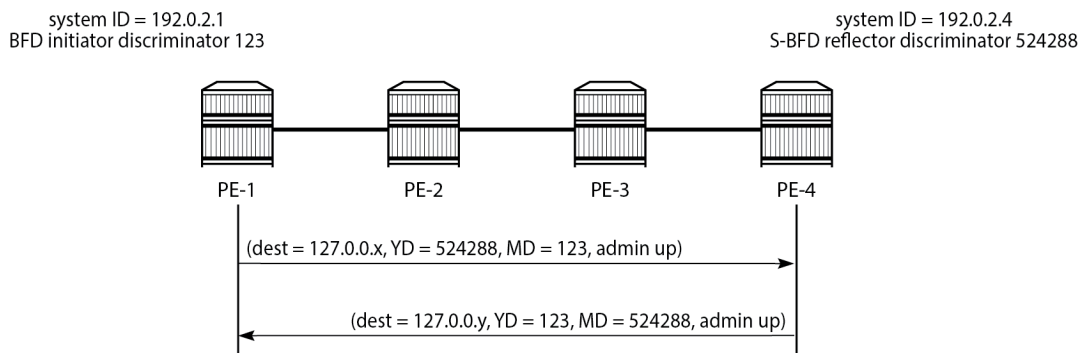
It also exchanges the 'my discriminator' and 'your discriminator' values in the reflected S-BFD packet.

S-BFD can be applied to SR-TE LSPs and the SR-TE LSP state can depend on the S-BFD session state.

S-BFD session establishment - continuity check

Figure 338: S-BFD session establishment - continuity check shows the continuity check S-BFD control packets between PE-1 and PE-4. On PE-1, the BFD (initiator) discriminator equals 123; on PE-4, the S-BFD (reflector) discriminator equals 524288. Head-end router PE-1 has a mapping table of remote discriminators to far-end IP addresses; for PE-4, the system ID is 192.0.2.4 and the S-BFD discriminator 524288. There is no INIT state in S-BFD. The mapping between the remote discriminators and the far-end IP addresses is required when the BFD return path is routed; when the BFD return path is controlled, no remote discriminators are used.

Figure 338: S-BFD session establishment - continuity check



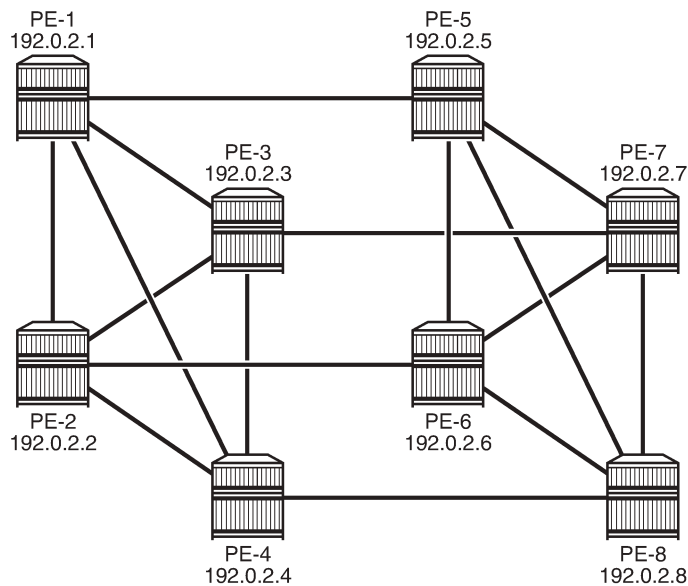
The session initiator node PE-1 generates an S-BFD control packet with destination PE-4 (but with an IP DA from the 127/8 range), YourDiscriminator 524288 (= S-BFD (reflector) discriminator value), MyDiscriminator 123 (= BFD (initiator) discriminator value), and admin state up.

The responder node PE-4 responds to PE-1 with an IP DA from the 127/8 range, YourDiscriminator 123, MyDiscriminator 524288, and admin state up. The admin state of the reflector reflects the configured S-BFD local state.

Configuration

Figure 339: Example topology shows the example topology with eight nodes.

Figure 339: Example topology



35630

The initial configuration includes:

- Cards, MDAs, ports
- Router interfaces
- IS-IS as IGP (alternatively, OSPF can be used) with traffic engineering (TE) enabled
- Segment routing enabled on all nodes
- MPLS and RSVP enabled on all router interfaces

The following will be configured:

- [S-BFD for SR-TE LSPs with routed return path](#) between PE-4 and PE-5
- [S-BFD for SR-TE LSPs with controlled return path](#) between PE-1 and PE-8



Note:

Even though BFD can use intervals smaller than 1000 ms, the used example setup has its limitations. The nodes in the used example setup are sims and the simulation for CPM-NP or central BFD sessions has the limitation that intervals that are configured with a value smaller than 1000 ms are always negotiated to intervals of 1000 ms. To avoid confusion when the configured intervals differ from the negotiated intervals on sims, a BFD template with intervals of 1000 ms is configured and used in this chapter.

S-BFD for SR-TE LSPs with routed return path

For S-BFD, the S-BFD (reflector) discriminator on the responder (tail-end) node must be known by both end nodes. The mapping between the remote discriminators and the far-end IP addresses can be configured statically or it can be learned dynamically from IGP. On each node, the reflector S-BFD discriminator must be in the range from 524288 to 526335 and the local state must be set to **up**.

Automated S-BFD distribution

In this example, one SR-TE LSP is established between head end PE-4 and tail end PE-5. On tail end PE-5, the global S-BFD configuration is as follows:

```
# on PE-5:
configure {
  bfd {
    seamless-bfd {
      reflector "PE-5" {
        admin-state enable
        discriminator 524292
        local-state up
      }
    }
  }
}
```

The S-BFD configuration on the other PEs is similar; in this example, it is sufficient to have the global S-BFD configuration on tail end PE-5 only. When the IGP is configured with **advertise-router-capability area** and **traffic-engineering true**, IGP routing protocol extensions provide the encodings to advertise the S-BFD discriminators as opaque information within the IGP link state information. This way, the remote IP addresses and the S-BFD discriminators are automatically mapped.

When PE-4 sets up an SR-TE LSP to PE-5, it will use a BFD discriminator—for example, 3—and S-BFD (reflector) discriminator 524292 for PE-5. For different LSPs toward PE-5, PE-4 will use different BFD discriminators combined with the same S-BFD (reflector) discriminator 524292.

Static S-BFD configuration

If **advertise-router-capability** or **traffic-engineering** are not configured, the S-BFD far-end IP address and its discriminator are statically mapped, as follows. When all SR-TE LSPs have far end PE-5, the mapping for PE-5 is sufficient.

```
# on PE-4:
configure {
  router "Base" {
    bfd {
      seamless-bfd {
        peer 192.0.2.1 {
          discriminator 524288
        }
        peer 192.0.2.2 {
          discriminator 524289
        }
        peer 192.0.2.3 {
          discriminator 524290
        }
        peer 192.0.2.5 {
          discriminator 524292
        }
        peer 192.0.2.6 {
          discriminator 524293
        }
        peer 192.0.2.7 {
          discriminator 524294
        }
        peer 192.0.2.8 {
          discriminator 524295
        }
      }
    }
  }
}
```

```
}
}
```

If the initiator receives a valid response from the reflector with an Up state, the initiator declares the S-BFD session as Up.



Note: Traffic engineering is not supported in VPRN or in OSPF3, so S-BFD discriminators cannot be automatically distributed in such cases.

Examples

S-BFD is only supported in CPM-NP on SR OS nodes, so the BFD type must be set to *cpm-np*. SR-TE LSPs can use CPM-NP BFD templates with a transmit and receive interval of minimum 10 ms. However, due to the simulation limitations on the sims in the example topology, the intervals are configured with a value of 1000 ms, as follows:

```
# on PE-4:
configure {
  bfd {
    bfd-template "bfd-cpm-np-1s" {
      receive-interval 1000
      transmit-interval 1000
      type cpm-np
    }
  }
}
```

On PE-4, the following paths and SR-TE LSPs are configured:

- "LSP-PE-4-PE-5_empty_localCSPF" with primary path "empty", which does not contain any explicit hops
- "LSP-PE-4-PE-5_viaPE-2_localCSPF" with primary path "via-PE-2", which contains 192.0.2.2 as a loose hop
- "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd" with primary path "via-PE-2" and secondary path "via-PE-3", which contains 192.0.2.3 as a loose hop

Any path computation method can be used. In the following example, the path computation method is local CSPF, as described in the [SR-TE LSP Path Computation Using Local CSPF](#) chapter. BFD can be configured per LSP or per path (primary or secondary) in the LSP.

```
# on PE-4:
configure {
  router "Base" {
    mpls {
      path "empty" {
        admin-state enable
      }
      path "via-PE-2" {
        admin-state enable
        hop 10 {
          ip-address 192.0.2.2
          type loose
        }
      }
      path "via-PE-3" {
        admin-state enable
        hop 10 {
          ip-address 192.0.2.3
          type loose
        }
      }
    }
  }
}
```



```

    }
  }
  lsp "LSP-PE-4-PE-5_empty_localCSPF" {
    admin-state enable
    type p2p-sr-te
    to 192.0.2.5
    pce-report true
    path-computation-method local-cspf
    max-sr-labels {
      additional-frr-labels 2
    }
    bfd {
      bfd-liveness true
      bfd-template "bfd-cpm-np-1s"
    }
    primary "empty" {
    }
  }
  lsp "LSP-PE-4-PE-5_viaPE-2_localCSPF" {
    admin-state enable
    type p2p-sr-te
    to 192.0.2.5
    pce-report true
    path-computation-method local-cspf
    max-sr-labels {
      additional-frr-labels 2
    }
    primary "via-PE-2" {
      bfd {
        bfd-liveness true
        bfd-template "bfd-cpm-np-1s"
      }
    }
  }
}
lsp "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd" {
  admin-state enable
  type p2p-sr-te
  to 192.0.2.5
  pce-report true
  path-computation-method local-cspf
  max-sr-labels {
    additional-frr-labels 2
  }
  bfd {
    bfd-liveness true
    bfd-template "bfd-cpm-np-1s"
  }
  primary "via-PE-2" {
  }
  secondary "via-PE-3" {
    standby true
  }
}
}

```

The head-end or initiator node PE-4 learned the S-BFD reflector discriminator for PE-5 (524292), so the BFD control packets can be sent with both a BFD and S-BFD discriminator value. The BFD control packets follow the data path from head end to tail end. The return path is native IP.

The first S-BFD session on initiator node PE-4 gets BFD discriminator 1, the second BFD discriminator 2, and so on. The S-BFD discriminator for PE-5 remains the same: 524292. For "LSP-PE-4-

PE-5_viaPE-2_localCSPF_2nd", with primary and secondary path, two S-BFD sessions are established: one with BFD discriminator 3 and another with BFD discriminator 4, as follows:

```
[/]
A:admin@PE-4# show router bfd seamless-bfd session lsp-name "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd" detail

=====
BFD Session
=====
Prefix          : 192.0.2.5/32
Local Address   : 192.0.2.4
LSP Name        : LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd
LSP Index       : 65538                Path LSP ID      : 33792
Fec Type        : srTe
Oper State      : Up
Up Time         : 0d 00:07:07          Protocols        : mplsLsp
Last Down Time  : 0d 00:00:01          Up Transitions   : 1
                                          Down Transitions : 0
                                          Version Mismatch : 0

Forwarding Information

Local Discr   : 3                    Local State      : Up
Local Diag      : 0 (None)
Local Mode      : Demand
Local Min Tx    : 1000
Last Sent (ms) : 0                    Local Mult       : 3
Type           : cpm-np               Local Min Rx     : 0
Remote Discr  : 524292                Remote State     : Up
Remote Diag     : 0 (None)            Remote Mode      : Async
Remote Min Tx   : 1000                Remote Mult      : 3
Remote C-flag   : 1
Last Recv (ms) : 0                    Remote Min Rx    : 3

=====
Prefix          : 192.0.2.5/32
Local Address   : 192.0.2.4
LSP Name        : LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd
LSP Index       : 65538                Path LSP ID      : 33794
Fec Type        : srTe
Oper State      : Up
Up Time         : 0d 00:07:07          Protocols        : mplsLsp
Last Down Time  : 0d 00:00:01          Up Transitions   : 1
                                          Down Transitions : 0
                                          Version Mismatch : 0

Forwarding Information

Local Discr   : 4                    Local State      : Up
Local Diag      : 0 (None)
Local Mode      : Demand
Local Min Tx    : 1000
Last Sent (ms) : 0                    Local Mult       : 3
Type           : cpm-np               Local Min Rx     : 0
Remote Discr  : 524292                Remote State     : Up
Remote Diag     : 0 (None)            Remote Mode      : Async
Remote Min Tx   : 1000                Remote Mult      : 3
Remote C-flag   : 1
Last Recv (ms) : 0                    Remote Min Rx    : 3

=====
=====
```

In the preceding **show** command, "Local Discr: 3" and "Local Discr: 4" refer to the BFD discriminator values on the initiator node PE-4, while "Remote Discr: 524292" refers to the S-BFD reflector discriminator value on the responder node PE-5.

The following command shows that the primary path "via-PE-2" goes from PE-4 via PE-2 and PE-1 to PE-5; the secondary path "via-PE-3" goes from PE-4 via PE-3 and PE-7 to PE-5:

```
[/]
A:admin@PE-4# show router mpls sr-te-lsp "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd" path detail

=====
MPLS SR-TE LSP LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd
Path (Detail)
=====
Legend :
  S      - Strict                L      - Loose
  A-SID  - Adjacency SID        N-SID  - Node SID
  +      - Inherited
=====
-----
LSP SR-TE LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd
Path via-PE-2
-----
LSP Name      : LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd
Path LSP ID   : 33792
From          : 192.0.2.4
To            : 192.0.2.5
Admin State   : Up                Oper State    : Up
Path Name     : via-PE-2
Path Type     : Primary
Path Admin    : Up                Path Oper     : Up
---snip---

Explicit Hops :
              : 192.0.2.2(L)
Actual Hops   :
  192.168.24.1(192.0.2.2) (A-SID)      Record Label : 524286
-> 192.168.12.1(192.0.2.1) (A-SID)      Record Label : 524287
-> 192.168.15.2(192.0.2.5) (A-SID)      Record Label : 524284

BFD Configuration and State
Template      : None                Ping Interval  : N/A
Enable       : False                State          : up
ReturnPathLabel : None
WaitForUpTimer : 4 sec              OperWaitForUpTimer: 4 sec
WaitForUpTmLeft : 0
StartFail Rsn : N/A

-----
LSP SR-TE LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd
Path via-PE-3
-----
LSP Name      : LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd
Path LSP ID   : 33794
From          : 192.0.2.4
To            : 192.0.2.5
Admin State   : Up                Oper State    : Up
Path Name     : via-PE-3
Path Type     : Standby
Path Admin    : Up                Path Oper     : Up
---snip---

Explicit Hops :
```

```

192.0.2.3(L)
Actual Hops      :
  192.168.34.1(192.0.2.3) (A-SID)      Record Label      : 524285
-> 192.168.37.2(192.0.2.7) (A-SID)      Record Label      : 524284
-> 192.168.57.1(192.0.2.5) (A-SID)      Record Label      : 524286
Srlg             : Disabled             Srlg Disjoint      : False

BFD Configuration and State
Template         : None                 Ping Interval      : N/A
Enable          : False                 State              : up
ReturnPathLabel : None
WaitForUpTimer  : 4 sec                 OperWaitForUpTimer: 4 sec
WaitForUpTmLeft: 0
StartFail Rsn   : N/A
=====

```

The following OAM LSP trace from PE-4 shows that the path goes via PE-2 and PE-1 to PE-5:

```

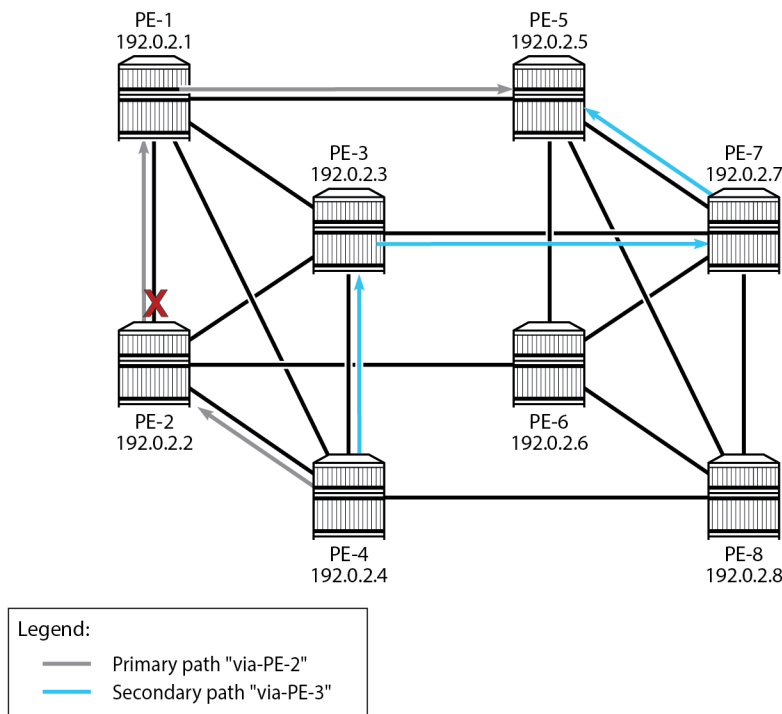
[/]
A:admin@PE-4# oam lsp-trace sr-te lsp-name "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd"
lsp-trace to LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd: 1 hops min, 30 hops max, 176 byte packets
1 192.0.2.2 rtt=2.21ms rc=3(EgressRtr) rsc=3
1 192.0.2.2 rtt=2.85ms rc=8(DSRtrMatchLabel) rsc=2
2 192.0.2.1 rtt=3.03ms rc=3(EgressRtr) rsc=2
2 192.0.2.1 rtt=3.38ms rc=8(DSRtrMatchLabel) rsc=1
3 192.0.2.5 rtt=4.40ms rc=3(EgressRtr) rsc=1

```

S-BFD session down without failure action

[Figure 340: Failure on remote link in primary path](#) shows the two paths of "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd" with a failure on the link between PE-2 and PE-1, which is part of the primary path "via-PE-2". The broken link is remote to the head-end node PE-4. The failure is emulated on PE-2 by disabling the port toward PE-1.

Figure 340: Failure on remote link in primary path



35631

As a result, the BFD session associated with the primary path "via-PE-2" goes down, as follows:

```
[/]
A:admin@PE-4# show router bfd seamless-bfd session lsp-name "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd"

=====
Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path   pp = Protecting path
=====
BFD Session
=====
```

Session Id	State	Tx Pkts	Rx Pkts
Rem Addr/Info/SdpId:VcId	Multipl	Tx Intvl	Rx Intvl
Protocols	Type	LAG Port	LAG ID
Loc Addr			LAG name
192.0.2.5/32	Down	N/A	N/A
192.0.2.5	3	1000	0
mplsLsp	cpm-np	N/A	N/A
192.0.2.4			
192.0.2.5/32	Up	N/A	N/A
192.0.2.5	3	1000	1000
mplsLsp	cpm-np	N/A	N/A
192.0.2.4			

```
-----
No. of BFD sessions: 2
=====
```

By default, there is no failure action on the BFD session, so the primary path remains up even when the BFD session on that path is down, as follows:

```
[/]
A:admin@PE-4# show router mpls sr-te-lsp "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd" path detail

=====
MPLS SR-TE LSP LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd
Path (Detail)
=====
Legend :
  S      - Strict                L      - Loose
  A-SID  - Adjacency SID        N-SID  - Node SID
  +      - Inherited
=====
-----
LSP SR-TE LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd
Path via-PE-2
-----
LSP Name      : LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd
Path LSP ID   : 33792
From          : 192.0.2.4
To            : 192.0.2.5
Admin State   : Up                Oper State    : Up
Path Name     : via-PE-2
Path Type     : Primary
Path Admin    : Up                Path Oper     : Up
---snip---

Explicit Hops :
              : 192.0.2.2(L)
Actual Hops   :
  192.168.24.1(192.0.2.2) (A-SID)      Record Label   : 524286
-> 192.168.12.1(192.0.2.1) (A-SID)      Record Label   : 524287
-> 192.168.15.2(192.0.2.5) (A-SID)      Record Label   : 524284

BFD Configuration and State
Template      : None                Ping Interval   : N/A
Enable        : False              State           : down
ReturnPathLabel : None
WaitForUpTimer : 4 sec             OperWaitForUpTimer: 4 sec
WaitForUpTmLeft : 0
StartFail Rsn  : N/A
-----
---snip---
```

The LSP and its paths remain up and the corresponding SR-TE tunnel in the tunnel table remains unchanged, so the traffic using the LSP will be blackholed. The following tunnel table lists three SR-TE tunnels, corresponding to:

- "LSP-PE-4-PE-5_empty_localCSPF", with next-hop 192.168.48.2 (PE-8)
- "LSP-PE-4-PE-5_viaPE-2_localCSPF", using path "via-PE-2", with next-hop 192.168.24.1
- "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd", using path "via-PE-2" (the primary path is used, not the secondary), with next-hop 192.168.24.1.

```
[/]
A:admin@PE-4# show router tunnel-table protocol sr-te

=====
IPv4 Tunnel Table (Router: Base)
```

```
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.5/32     sr-te     MPLS  655362    8    192.168.48.2  20
192.0.2.5/32     sr-te     MPLS  655363    8    192.168.24.1  30
192.0.2.5/32     sr-te     MPLS  655364    8    192.168.24.1  30
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
```

The OAM LSP ping command using the SR-TE LSP "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd" fails, as follows:

```
[/]
A:admin@PE-4# oam lsp-ping sr-te lsp-name "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd"
LSP-PING LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd: 96 bytes MPLS payload
Request timed out.

---- LSP LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd PING Statistics ----
1 packet sent, 0 packets received, 100% packet loss
```

The OAM LSP trace command shows that the LSP trace stops at PE-2 (192.0.2.2):

```
[/]
A:admin@PE-4# oam lsp-trace sr-te lsp-name "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd"
lsp-trace to LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd: 1 hops min, 30 hops max, 176 byte packets
1 192.0.2.2 rtt=2.15ms rc=3(EgressRtr) rsc=3
1 192.0.2.2 rtt=2.40ms rc=11(DSNoLabelEntry) rsc=2
```

S-BFD session down with failure action

To force a failover to the secondary path or to bring the LSP down when the BFD session goes down, a failure action needs to be configured in the BFD context of the LSP, as follows:

```
# on PE-4:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd" {
        bfd {
          bfd-liveness true
          bfd-template "bfd-cpm-np-ls"
          failure-action failover-or-down
          wait-for-up-timer 4 # default; applicable for failure action
        }
      }
    }
  }
}
```

The failure action **failover-or-down** is the only failure action that is allowed for SR-TE LSPs. An error is raised when attempting to configure failure action **down** or failure action **failover**, as follows:

```
*[ex:/configure router "Base" mpls lsp "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd" bfd]
A:admin@PE-4# commit
MINOR: MGMT_CORE #3001: configure router "Base" mpls lsp "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd"
bfd failure-action - 'failure-action down' is not allowed for this LSP type
```

```
*[ex:/configure router "Base" mpls lsp "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd" bfd]
A:admin@PE-4# commit
MINOR: MGMT_CORE #3001: configure router "Base" mpls lsp "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd"
bfd failure-action - 'failure-action failover' is not allowed for this LSP type
```

When the failure action is configured, the primary path "via-PE-2" goes down and a failover takes place to the secondary path "via-PE-3" (if available). When no secondary paths are available, the LSP is operational down.

When a link or node fails on the primary path, the BFD state goes down for the primary path. The head-end node switches to the best preference standby that is up. When the LSP retry timer expires, the MPLS module initiates a local CSPF request to calculate a new SR-TE path. When it is possible to calculate a new path meeting the path constraints for the primary path, the new path is added to the SR-TE tunnel, and S-BFD for the primary path is started. S-BFD comes up and the LSP metric is set.

By default, the revert timer is zero, so no reversion to the primary path takes place. However, if the revert timer is configured to a non-zero value, the revert timer starts when the S-BFD session comes up. When the revert timer expires, the active path is reprogrammed from secondary to primary. If **pce-report-enable** is configured, a PCEP status report is sent for each path, so two reports are sent.

The following command shows that the primary path "via-PE-2" is down and the list of actual hops is empty. Therefore, the S-BFD session state is not applicable. The secondary path remains up and the LSP is up.

```
[/]
A:admin@PE-4# show router mpls sr-te-lsp "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd" path detail

=====
MPLS SR-TE LSP LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd
Path (Detail)
=====
Legend :
  S      - Strict                L      - Loose
  A-SID  - Adjacency SID        N-SID  - Node SID
  +      - Inherited
=====
-----
LSP SR-TE LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd
Path via-PE-2
-----
LSP Name      : LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd
Path LSP ID   : 33796
From          : 192.0.2.4
To            : 192.0.2.5
Admin State   : Up                Oper State    : Up
Path Name     : via-PE-2
Path Type     : Primary
Path Admin    : Up                Path Oper     : Down
---snip---

Failure Code   : bfdDown
Failure Node   : 192.0.2.4
Explicit Hops  :
                192.0.2.2(L)
Actual Hops    :
  No Hops Specified

BFD Configuration and State
Template       : None                Ping Interval  : N/A
Enable        : False                State          : notApplicable
ReturnPathLabel : None
```



```

WaitForUpTimer   : 4 sec                OperWaitForUpTimer: 4 sec
WaitForUpTmLeft  : 0
StartFail Rsn    : N/A

-----
LSP SR-TE LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd
Path via-PE-3
-----
LSP Name       : LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd
Path LSP ID    : 33794
From           : 192.0.2.4
To             : 192.0.2.5
Admin State    : Up                    Oper State      : Up
Path Name     : via-PE-3
Path Type     : Standby
Path Admin    : Up                    Path Oper      : Up
---snip---

Failure Code    : noError
Failure Node    : n/a
Explicit Hops   :
                192.0.2.3(L)
Actual Hops     :
  192.168.34.1(192.0.2.3) (A-SID)      Record Label    : 524285
-> 192.168.37.2(192.0.2.7) (A-SID)      Record Label    : 524284
-> 192.168.57.1(192.0.2.5) (A-SID)      Record Label    : 524286
Srlg           : Disabled              Srlg Disjoint   : False

BFD Configuration and State
Template       : None                  Ping Interval   : N/A
Enable        : False                 State           : up
ReturnPathLabel : None
WaitForUpTimer : 4 sec                OperWaitForUpTimer: 4 sec
WaitForUpTmLeft : 0
StartFail Rsn  : N/A

=====

```

The tunnel table shows an entry with tunnel ID 655364, which corresponds to "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd", with next-hop 192.168.34.1 (PE-3):

```

[/]
A:admin@PE-4# show router tunnel-table protocol sr-te

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.5/32     sr-te     MPLS  655362   8    192.168.48.2  20
192.0.2.5/32     sr-te     MPLS  655363   8    192.168.24.1  30
192.0.2.5/32     sr-te     MPLS  655364   8    192.168.34.1  30
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====

```

The OAM LSP trace using "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd" shows that the active path goes via PE-3 and PE-7 to PE-5, as follows:

```
[/]
A:admin@PE-4# oam lsp-trace sr-te lsp-name "LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd"
lsp-trace to LSP-PE-4-PE-5_viaPE-2_localCSPF_2nd: 1 hops min, 30 hops max, 176 byte packets
1 192.0.2.3 rtt=1.82ms rc=3(EgressRtr) rsc=3
1 192.0.2.3 rtt=2.49ms rc=8(DSRtrMatchLabel) rsc=2
2 192.0.2.7 rtt=4.91ms rc=3(EgressRtr) rsc=2
2 192.0.2.7 rtt=4.18ms rc=8(DSRtrMatchLabel) rsc=1
3 192.0.2.5 rtt=4.11ms rc=3(EgressRtr) rsc=1
```

S-BFD for SR-TE LSPs with controlled return path

In this mode, a controlled return path for BFD reply packets is configured at the initiating node. The reflector function at the far end of the SR-TE LSP is bypassed, so there is no need to configure reflector discriminators for these sessions.

The initiating node pushes an additional MPLS label on S-BFD packets at the bottom of the stack and the BFD session operates in echo mode. The return path label refers to an MPLS binding SID of an SR policy programmed at the far end of the SR-TE LSP. The SR policy can be used to forward BFD reply packets along an explicit TE path back to the initiator, avoiding the IGP shortest path.

It is possible to configure a specific TE return path for each S-BFD session on an SR-TE LSP at the initiating node. The SR policies can have segments lists with different paths, ensuring the BFD reply packets from different LSP paths do not share the same outcome.

In the following example, initiating node PE-1 has three SR-TE LSPs to far end PE-8:

- SR-TE LSP "LSP-PE-1-PE-8_empty_localCSPF" with an empty primary path and return path label 20041
- SR-TE LSP "LSP-PE-1-PE-8_viaPE-2_localCSPF" with primary path "via-PE-2" and return path label 20621
- SR-TE LSP "LSP-PE-1-PE-8_viaPE-2_localCSPF_2nd" with primary path "via-PE-2" and return path label 20621 and secondary path "via-PE-3" and return path label 20051

The configuration of the paths and the SR-TE LSPs on PE-1 is as follows:

```
# on PE-1:
configure {
  bfd {
    bfd-template "bfd-cpm-np-1s" {
      receive-interval 1000
      transmit-interval 1000
      type cpm-np
    }
  }
  router "Base" {
    mpls {
      admin-state enable
      interface "int-PE-1-PE-2" {
      }
      interface "int-PE-1-PE-3" {
      }
      interface "int-PE-1-PE-4" {
      }
      interface "int-PE-1-PE-5" {
      }
    }
  }
}
```

```
path "empty" {
    admin-state enable
}
path "via-PE-2" {
    admin-state enable
    hop 10 {
        ip-address 192.0.2.2
        type loose
    }
}
path "via-PE-3" {
    admin-state enable
    hop 10 {
        ip-address 192.0.2.3
        type loose
    }
}
lsp "LSP-PE-1-PE-8_empty_localCSPF" {
    admin-state enable
    type p2p-sr-te
    to 192.0.2.8
    pce-report true
    path-computation-method local-cspf
    max-sr-labels {
        additional-frr-labels 2
    }
    bfd {
        bfd-liveness true
        bfd-template "bfd-cpm-np-1s"
        failure-action failover-or-down
        return-path-label 20041
    }
    primary "empty" {
    }
}
lsp "LSP-PE-1-PE-8_viaPE-2_localCSPF" {
    admin-state enable
    type p2p-sr-te
    to 192.0.2.8
    pce-report true
    path-computation-method local-cspf
    max-sr-labels {
        additional-frr-labels 2
    }
    bfd {
        failure-action failover-or-down
    }
    primary "via-PE-2" {
        bfd {
            bfd-liveness true
            bfd-template "bfd-cpm-np-1s"
            return-path-label 20621
        }
    }
}
lsp "LSP-PE-1-PE-8_viaPE-2_localCSPF_2nd" {
    admin-state enable
    type p2p-sr-te
    to 192.0.2.8
    pce-report true
    path-computation-method local-cspf
    max-sr-labels {
        additional-frr-labels 2
    }
}
```

```

bfd {
    failure-action failover-or-down
}
primary "via-PE-2" {
    bfd {
        bfd-liveness true
        bfd-template "bfd-cpm-np-1s"
        return-path-label 20621
    }
}
secondary "via-PE-3" {
    standby true
    bfd {
        bfd-liveness true
        bfd-template "bfd-cpm-np-1s"
        return-path-label 20051
    }
}
}

```

The return path labels correspond to binding SIDs in SR policies on PE-8, as follows:

```

# on PE-8:
configure {
    router "Base" {
        mpls-labels {
            sr-labels {
                start 32000
                end 32999
            }
            reserved-label-block "SRLB1" {
                start-label 20000
                end-label 21999
            }
        }
        segment-routing {
            sr-policies {
                admin-state enable
                reserved-label-block "SRLB1"
                static-policy "SR-static-policy-PE-4-PE-1" {
                    admin-state enable
                    color 810
                    endpoint 192.0.2.1
                    head-end local
                    binding-sid 20041
                    distinguisher 10020041
                    segment-list 1 {
                        admin-state enable
                        segment 1 {
                            mpls-label 32004    # node SID for PE-4
                        }
                        segment 2 {
                            mpls-label 32001    # node SID for PE-1
                        }
                    }
                }
                static-policy "SR-static-policy-PE-5-PE-1" {
                    admin-state enable
                    color 820
                    endpoint 192.0.2.1
                    head-end local
                    binding-sid 20051
                    distinguisher 10020051
                }
            }
        }
    }
}

```



```

Legend:
  Session Id = Interface Name | LSP Name | Prefix | RSVP Sess Name | Service Id
  wp = Working path   pp = Protecting path
=====
BFD Session
=====
Session Id          State      Tx Pkts  Rx Pkts
Rem Addr/Info/SdpId Multipl   Tx Intvl Rx Intvl
Protocols          Type     LAG Port  LAG ID
Loc Addr          LAG name
-----
192.0.2.8/32       Up        N/A      N/A
192.0.2.8         3        1000    1000
mplsLsp           cpm-np   N/A      N/A
192.0.2.1
192.0.2.8/32       Up        N/A      N/A
192.0.2.8         3        1000    1000
mplsLsp           cpm-np   N/A      N/A
192.0.2.1
192.0.2.8/32       Up        N/A      N/A
192.0.2.8         3        1000    1000
mplsLsp           cpm-np   N/A      N/A
192.0.2.1
192.0.2.8/32       Up        N/A      N/A
192.0.2.8         3        1000    1000
mplsLsp           cpm-np   N/A      N/A
192.0.2.1
-----
No. of BFD sessions: 4
=====

```

When the SR policies on PE-8 are down, the corresponding BFD sessions on PE-1 go down.

On PE-1, SR-TE LSP "LSP-PE-1-PE-8_viaPE-2_localCSPF_2nd" has a primary path and a standby secondary path. The local discriminator for the primary path is 7; for the secondary path 5. No remote discriminators are used when the return path corresponds to an SR policy, so the remote discriminators equal zero. The return path label is the binding SID of the SR policy in the far end node.

```

[/]
A:admin@PE-1# show router bfd seamless-bfd session lsp-name "LSP-PE-1-PE-8_viaPE-2_localCSPF_
2nd" detail
=====
BFD Session
=====
Prefix           : 192.0.2.8/32
Local Address    : 192.0.2.1
LSP Name         : LSP-PE-1-PE-8_viaPE-2_localCSPF_2nd
LSP Index        : 65538          Path LSP ID      : 54784
Fec Type         : srTe
Return Path      : 20621
Oper State       : Up
Up Time          : 0d 00:05:18   Protocols        : mplsLsp
Last Down Time  : 0d 00:00:01   Up Transitions   : 1
                                   Down Transitions : 0
                                   Version Mismatch  : 0

Forwarding Information

Local Discr      : 7          Local State       : Up
Local Diag       : 0 (None)
Local Mode       : Demand
Local Min Tx     : 1000      Local Mult        : 3
Last Sent (ms)  : 0          Local Min Rx      : 1000

```

```

Type           : cpm-np
Remote Discr   : 0
Remote Diag    : 0 (None)
Remote Min Tx  : 1000
Remote C-flag  : 1
Last Recv (ms) : 0
Remote State   : Up
Remote Mode    : Demand
Remote Mult    : 3
Remote Min Rx  : 1000
=====
Prefix         : 192.0.2.8/32
Local Address  : 192.0.2.1
LSP Name       : LSP-PE-1-PE-8_viaPE-2_localCSPF_2nd
LSP Index      : 65538
Path LSP ID    : 54786
Fec Type       : srTe
Return Path    : 20051
Oper State     : Up
Up Time        : 0d 00:05:27
Last Down Time : 0d 00:00:01
Protocols      : mplsLsp
Up Transitions : 1
Down Transitions : 0
Version Mismatch : 0

Forwarding Information

Local Discr    : 5
Local Diag     : 0 (None)
Local Mode     : Demand
Local Min Tx   : 1000
Last Sent (ms) : 0
Type          : cpm-np
Remote Discr   : 0
Remote Diag    : 0 (None)
Remote Min Tx  : 1000
Remote C-flag  : 1
Last Recv (ms) : 0
Local State    : Up
Local Mult     : 3
Local Min Rx   : 1000
Remote State   : Up
Remote Mode    : Demand
Remote Mult    : 3
Remote Min Rx  : 1000
=====
=====

```

Conclusion

Seamless BFD for SR-TE LSPs allows fast connectivity checking of the data plane of the LSP. This can be used to trigger fast failover from the currently active to a standby path.

Segment Routing – Traffic Engineered Tunnels

This chapter provides information about Segment Routing – Traffic Engineered Tunnels.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 14.0.R7, but the MD-CLI in the current edition corresponds to SR OS Release 21.2.R1.

Overview

Segment Routing (SR) is described in the chapter [Segment Routing with IS-IS Control Plane](#), where the advertisement of node prefix segment identifiers (SIDs) cause the automatic creation of ECMP-aware shortest path MPLS tunnels on each SR-aware router. Each node prefix SID is a globally unique value and becomes an MPLS label in the MPLS data plane. The label is advertised and learned by each SR-capable router using control plane extensions to the IS-IS and OSPF protocols.

It is also possible to create source-routed traffic-engineered end-to-end segment routing paths, where routing constraints such as strict or loose hops can be used to determine a data path to be taken through a network.

These are known as Segment Routing Traffic Engineered (SR-TE) Label Switched Paths (LSPs) and use the same command line construct as that used in configuring RSVP-TE LSPs. However, SR-TE LSPs differ in that there is no mid-point state; each intermediate and tail-end router is unaware of the presence of the LSP because there is no signaling protocol used to create the path. The path can be computed locally by the ingress PE or by offloading the path computation to an external controller.

If a packet is forwarded through the SR tunnel, each router along the path will read the top label and forward the packet according to the SR tunnel table entry for that label.

This chapter describes the configuration of SR-TE LSPs with locally-computed source-routed paths and how they can be used in the data plane of Layer 2 and Layer 3 services. In the cases described, an SR-TE LSP containing a number of strict or loose hops is created at the head-end router and used to construct an LSP by translating the IP addresses configured in the MPLS path to an SID. This results in an MPLS path with state at the head end only, comprising a stack of SIDs, where each SID is an MPLS label.

In this chapter, OSPF is used to advertise the SIDs and a set of extensions to OSPF have been defined, which require additional configuration on each network router.

The LSP is instantiated—the state is operationally "up"—and a tunnel table entry is created that is owned by the SR-TE protocol. Any data packet that is resolved to use the resulting tunnel has the label stack imposed at the head-end router and is forwarded out of the appropriate next-hop interface. This interface is determined by the topmost label in the stack.

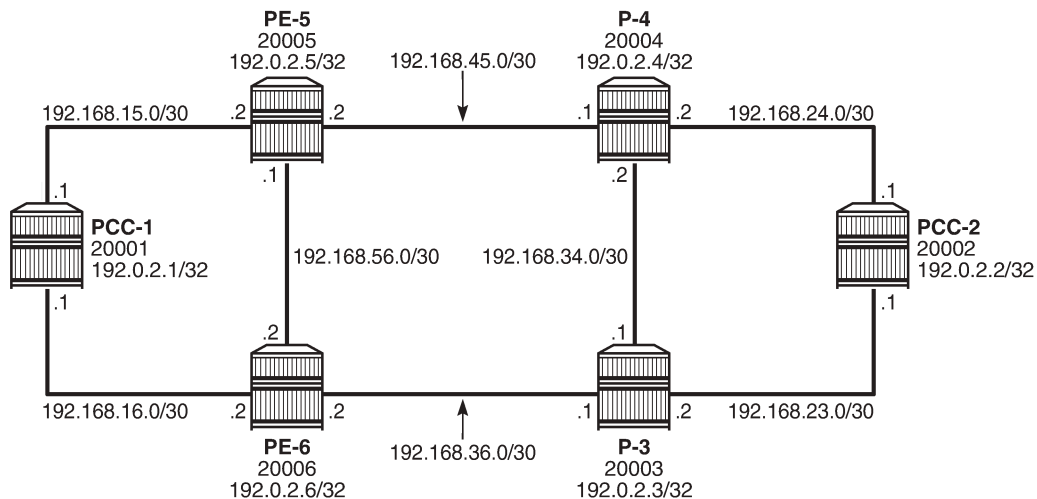
If the label is a node SID, the outgoing interface is determined by the IGP—the shortest path to the router that the node SID represents.

If the label is a local adjacency SID, the outgoing interface is the local interface for which this SID is generated by the IGP.

The segments referenced can be a prefix segment, such as a node segment or an adjacency segment, which represents a specific adjacency between two nodes. The SIDs are used as MPLS labels.

In the following configuration examples, the LSP path is created at the head-end router, and computed by translating a list of hops containing IP addresses into a list of SIDs, by examining the OSPF TE database. The head-end router is referred to as a Path Computation Client (PCC). [Figure 341: Segment routing network schematic](#) shows the example topology used, and a pair of bidirectional connected SR-TE LSPs between PCC-1 and PCC-2 will be configured to illustrate SR-TE LSPs. All interfaces between PCC-1 and its neighbors have the OSPF metric set to 1000. Similarly, for PCC-2, the OSPF metric is also set to 1000 between itself and its neighbors. The OSPF metric on router interfaces between the core routers P-3, P-4, PE-5, and PE-6 are set to 100.

Figure 341: Segment routing network schematic



26381

Configuration

MPLS label range

The MPLS label range must be configured. This represents the Segment Routing Global Block (SRGB) from which node SIDs are allocated. The choice of SRGB in this example is the same as that chosen in the chapter [Segment Routing with IS-IS Control Plane](#), where the label block is the same for each router. The SRGB is a contiguous range within the dynamic range 18432 to 524287, as shown in the following output:

```
[/]
A:admin@PCC-1# show router mpls-labels label-range
```

```
=====
Label Ranges
```

```
=====
```

Label Type	Start Label	End Label	Aging	Available	Total
Static	32	18431	-	18400	18400
Dynamic	18432	524287	0	505856	505856
Seg-Route	0	0	-	0	0

```
=====
```

In this example, a range of 1000 labels is chosen. For operational simplicity, Nokia recommends that the same label range is chosen for each router. However, this is not an explicit requirement.

A label range of 20000 to 20999 for SR is configured with the following command:

```
# on all nodes:
configure {
  router "Base" {
    mpls-labels {
      sr-labels {
        start 20000
        end 20999
      }
    }
  }
}
```

When the SRGB label range has been configured, the MPLS label range looks as follows:

```
[/]
A:admin@PCC-1# show router mpls-labels label-range

=====
Label Ranges
=====
```

Label Type	Start Label	End Label	Aging	Available	Total
Static	32	18431	-	18400	18400
Dynamic	18432	524287	0	504856	505856
Seg-Route	20000	20999	-	0	1000

```
=====
```

Global OSPF configuration

The first step is to configure OSPF on each router, as shown in [Figure 341: Segment routing network schematic](#). All router interfaces are members of a single backbone area: area 0.0.0.0.

The configuration for PCC-1 to enable OSPF is:

```
# on PCC-1:
configure {
  router "Base" {
    ospf 0 {
      admin-state enable
      area 0.0.0.0 {
        interface "int-PCC-1-PE-5" {
          interface-type point-to-point
          metric 1000
        }
        interface "int-PCC-1-PE-6" {
          interface-type point-to-point
          metric 1000
        }
      }
      interface "system" {
```

```

    }
  }
}

```

The configuration for all other nodes is the same, apart from the IP addresses. The IP addresses can be derived from [Figure 341: Segment routing network schematic](#).

For each router to be segment-routing capable, additional configuration within the OSPF context is required. For PCC-1, this is as follows:

```

# on PCC-1:
configure {
  router "Base" {
    ospf 0 {
      admin-state enable
      advertise-router-capability area
      traffic-engineering true
      segment-routing {
        admin-state enable
        prefix-sid-range {
          global
        }
      }
    }
    area 0.0.0.0 {
      interface "system" {
        node-sid {
          label 20001
        }
      }
    }
  }
}

```

The router capability is enabled using the **advertise-router-capability area** command, which defines the flooding scope of the opaque LSA used for this purpose as area. Traffic engineering is also enabled.

Also, MPLS and RSVP must be enabled on each router interface to ensure that OSPF opaque LSAs are generated.

A node SID is manually configured as a label, equivalent to the absolute node SID value. It is possible to configure the node SID as an index. Indexing is explained in the chapter [Segment Routing with IS-IS Control Plane](#).

Finally, segment routing is enabled, along with the **prefix-sid-range** command that states that the node prefix SID values of all routers within the network will be within the range of the global block.

The value of the **prefix-sid-range** must be the same for all routers; in this case, the range is always 1000.

The following output taken from PCC-1 shows the prefix SIDs configured on the routers in the network and advertised using OSPF. This will be identical for all routers in the network.

```

[/]
A:admin@PCC-1# show router ospf prefix-sids
=====
Rtr Base OSPFv2 Instance 0 Prefix-Sids
=====
Prefix                               Area          RtType        SID
Adv-Rtr                               SRMS          Flags
-----
192.0.2.1/32                          0.0.0.0       INTRA-AREA 1
                                           192.0.2.1     N           NnP
192.0.2.2/32                          0.0.0.0       INTRA-AREA 2
                                           192.0.2.2     N           NnP
192.0.2.3/32                          0.0.0.0       INTRA-AREA 3

```

```

192.0.2.3      N      NnP
192.0.2.4/32  0.0.0.0  INTRA-AREA 4
192.0.2.4      N      NnP
192.0.2.5/32  0.0.0.0  INTRA-AREA 5
192.0.2.5      N      NnP
192.0.2.6/32  0.0.0.0  INTRA-AREA 6
192.0.2.6      N      NnP
-----
No. of Prefix/SIDs: 6
SRMS      : Y/N = prefix SID advertised by SR Mapping Server (Y) or not (N)
           S = SRMS prefix SID is selected to be programmed
SID Flags : N = Node-SID

nP = no penultimate hop POP

M = Mapping server

E = Explicit-Null

V = Prefix-SID carries a value
L = value/index has local significance

I = Inter Area flag

A = Attached flag

B = Backup flag

=====

```

The prefix SID for each node is displayed as an index; for example, 1. The absolute value of the node SID is obtained by adding the (label_base) + (advertised SID index) = node prefix SID. The base label value for each router is chosen to be 20000, so the node prefix SID for PCC-1, for example, is 20000 + 1 = 20001.

Adjacency SIDs are generated by OSPF for each interface link, and are advertised within the extended link opaque LSA using the adjacency SID sub-TLV. The following output shows the extended link opaque LSAs of PCC-1. There are two network links, so there are two LSAs, with link state IDs of 8.0.0.2 and 8.0.0.3.

```

[/]
A:admin@PCC-1# show router ospf opaque-database adv-router 192.0.2.1 detail

=====
Rtr Base OSPFv2 Instance 0 Opaque Link State Database (type: All) (detail)
=====
---snip---
-----
Opaque LSA
-----
Area Id       : 0.0.0.0          Adv Router Id  : 192.0.2.1
Link State Id : 8.0.0.2            LSA Type       : Area Opaque
Sequence No   : 0x80000001      Checksum       : 0x2f79
Age           : 166             Length         : 48
Options       : E
Advertisement : Extended Link
  TLV Extended link (1) Len 24 :
    link Type=P2P (1) Id=192.0.2.5 Data=192.168.15.1
  Sub-TLV Adj-SID (2) len 7 :
    Flags=Value Local (0x60)
    MT-ID=0 Weight=0 SID/Index/Label=524287
-----
Opaque LSA
-----
Area Id       : 0.0.0.0          Adv Router Id  : 192.0.2.1

```

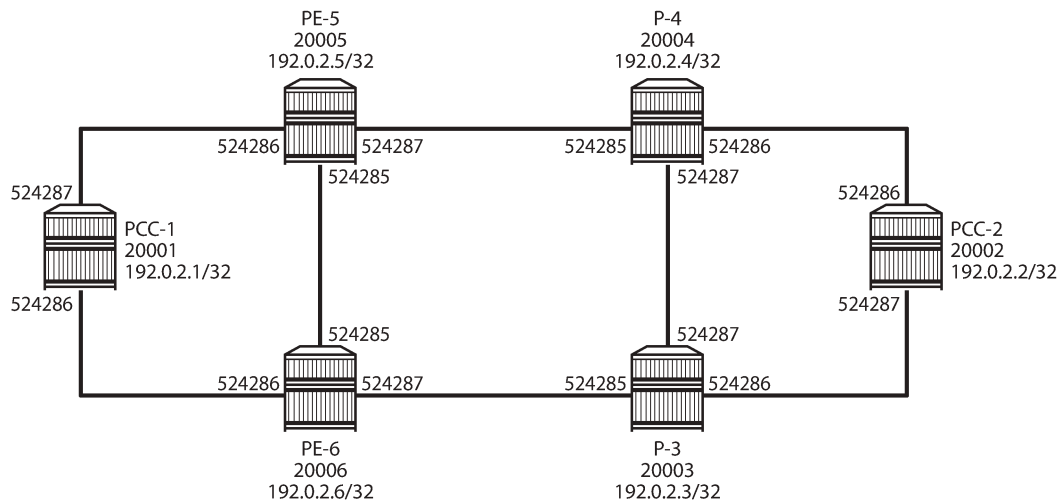
```

Link State Id   : 8.0.0.3           LSA Type      : Area Opaque
Sequence No    : 0x80000001        Checksum      : 0x277f
Age            : 166                Length        : 48
Options        : E
Advertisement   : Extended Link
  TLV Extended link (1) Len 24 :
    link Type=P2P (1) Id=192.0.2.6 Data=192.168.16.1
  Sub-TLV Adj-SID (2) len 7 :
    Flags=Value Local (0x60)
    MT-ID=0 Weight=0 SID/Index/Label=524286
=====
    
```

The adjacency SID for interface on PCC-1 toward PE-5 is 524287, and the adjacency SID for the interface toward PE-6 is 524286.

A full collection of SIDs for the whole network is shown in [Figure 342: Node and adjacency SIDs](#).

Figure 342: Node and adjacency SIDs



26382

Segment routing TE-LSPs

This section describes SR-TE LSPs that are configured on the head-end router (the PCC). The path taken through the network is computed locally by the PCC. To influence the path taken, a series of strict and/or loose hops are configured in an MPLS path.



Note:

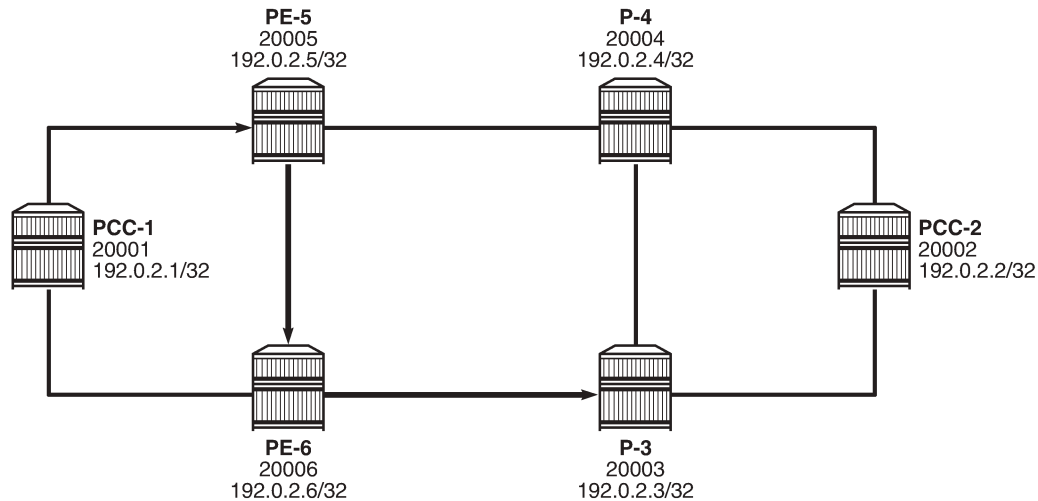
SR-TE LSPs configured with a loose path that contains no hops is effectively a shortest path tunnel to the destination node. The destination address is resolved to the node SID of the tail-end router.

PCC-initiated and computed LSP – strict path

Consider an SR-TE LSP configured on PCC-1, with tail end at PCC-2. Assume there is a requirement for the LSP to avoid the link from PE-5 to P-4 during normal working, so a strict path from PCC-1 via PE-5 to

PE-6, and then on to P-3 is required before being forwarded to PCC-2. This is shown in [Figure 343: PCC computed strict path between PCC-1 and PCC-2](#).

Figure 343: PCC computed strict path between PCC-1 and PCC-2



26383

To meet these requirements, an MPLS path is configured containing the following strict hops, using the system addresses to identify the hops. The following configures the MPLS path required on PCC-1. This uses the identical CLI construct as an MPLS path used in configuring an RSVP-TE LSP.

```

# on PCC-1:
configure {
  router "Base" {
    mpls {
      path "PCC-controlled-strict-path" {
        admin-state enable
        hop 1 {
          ip-address 192.0.2.5
          type strict
        }
        hop 2 {
          ip-address 192.0.2.6
          type strict
        }
        hop 3 {
          ip-address 192.0.2.3
          type strict
        }
      }
    }
  }
}
  
```

The SR-TE LSP is configured on PCC-1 as follows:

```

# on PCC-1:
configure {
  router "Base" {
    mpls {
      lsp "PCC-1-PCC-2-PCC-strict-lsp" {
        admin-state enable
        type p2p-sr-te
        to 192.0.2.2
      }
    }
  }
}
  
```

```

        primary "PCC-controlled-strict-path" {
        }
    }

```

Again, the same CLI construct as an RSVP-TE LSP is used, except for the type **p2p-sr-te**. If the type is not **p2p-sr-te**, the LSP is signaled as an RSVP-TE LSP. The LSP configuration references the previously-created MPLS path as the primary path.

When **enabled**, the LSP path status is as shown in the following output:

```

[/]
A:admin@PCC-1# show router mpls sr-te-lsp "PCC-1-PCC-2-PCC-strict-lsp" path detail
=====
MPLS SR-TE LSP PCC-1-PCC-2-PCC-strict-lsp
Path (Detail)
=====
Legend :
  S      - Strict                L      - Loose
  A-SID  - Adjacency SID        N-SID  - Node SID
  +      - Inherited
=====
LSP SR-TE PCC-1-PCC-2-PCC-strict-lsp
Path PCC-controlled-strict-path
-----
LSP Name      : PCC-1-PCC-2-PCC-strict-lsp
Path LSP ID   : 41984
From          : 192.0.2.1
To            : 192.0.2.2
Admin State   : Up                Oper State    : Up
Path Name     : PCC-controlled-strict-path
Path Type     : Primary
Path Admin    : Up                Path Oper     : Up
Path Up Time  : 0d 00:00:23        Path Down Time : 0d 00:00:00
Retry Limit   : 0                  Retry Timer    : 30 sec
Retry Attempt : 0                  Next Retry In  : 0 sec

PathCompMethod : none                OperPathCompMethod: none
MetricType     : igp                  Oper MetricType  : igp
LocalSrProt    : preferred            Oper LocalSrProt : N/A
LabelStackRed  : Disabled              Oper LabelStackRed: N/A

Bandwidth      : No Reservation        Oper Bandwidth   : 0 Mbps
Hop Limit      : 255                   Oper HopLimit    : 255
Setup Priority  : 7                     Oper SetupPriority: 7
Hold Priority   : 0                     Oper HoldPriority : 0
Inter-area     : N/A

PCE Updt ID    : 0                    PCE Updt State  : None
PCE Upd Fail Code: noError

PCE Report     : Disabled+            Oper PCE Report  : Disabled
PCE Control    : Disabled              Oper PCE Control : Disabled

Include Groups :                       Oper IncludeGroups:
None                                                    None
Exclude Groups :                       Oper ExcludeGroups:
None                                                    None
Last Resignal  : n/a

IGP/TE Metric  : 16777215              Oper Metric      : 16777215
Oper MTU       : 1548                   Path Trans      : 1
Failure Code    : noError

```

```

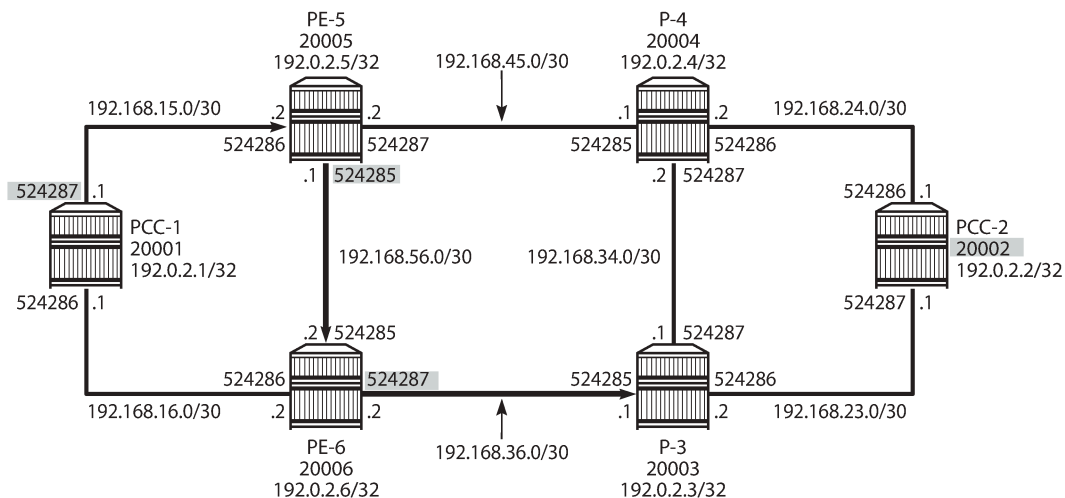
Failure Node      : n/a
Explicit Hops     :
                  192.0.2.5(S)
                  -> 192.0.2.6(S)
                  -> 192.0.2.3(S)
Actual Hops       :
                  192.168.15.2(192.0.2.5) (A-SID)
                  -> 192.168.56.2(192.0.2.6) (A-SID)
                  -> 192.168.36.1(192.0.2.3) (A-SID)
                  -> 192.0.2.2(192.0.2.2) (N-SID)
Record Label      : 524287
Record Label      : 524285
Record Label      : 524287
Record Label      : 20002

BFD Configuration and State
Template          : None
Enable           : False
WaitForUpTimer   : 4 sec
WaitForUpTmLeft  : 0
StartFail Rsn    : N/A
Ping Interval    : N/A
State            : notApplicable
OperWaitForUpTimer: 0 sec
    
```

The Actual Hops output shows the address of the upstream router facing the configured strict hop (in brackets) referenced in the MPLS path, plus a loose hop for the destination hop of 192.0.2.2.

The interface addresses are translated into SIDs to be used as MPLS labels, by the head-end PCC router, PCC-1, by examining the OSPF TE database. Each strict hop is always translated into an adjacency SID (A-SID), and a loose hop is always translated into a node SID (N-SID). This is shown in [Figure 344: PCC computed LSP hop-to-label translation](#).

Figure 344: PCC computed LSP hop-to-label translation



26384

When the LSP is connected, the Tunnel Table Manager (TTM) adds an entry for the SR-TE LSP. This LSP is available for the provisioning of services that use the TTM. The following output shows the tunnel table for PCC-1, which includes the shortest-path tunnels to all other routers in the network, plus the entry for the provisioned SR-TE LSP. The default preference for an SR-TE LSP in the tunnel table is 8.

```

[/]
A:admin@PCC-1# show router tunnel-table
    
```



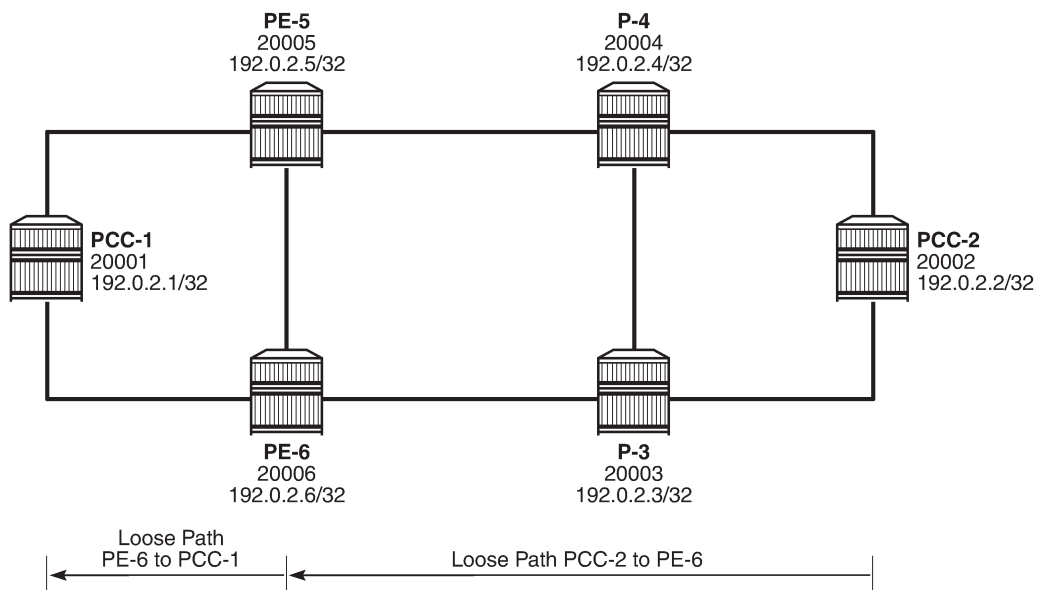
```
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.2/32     sr-te     MPLS  655362    8    192.168.15.2 16777215
192.0.2.2/32     ospf (0)  MPLS  524291   10    192.168.15.2 2100
192.0.2.3/32     ospf (0)  MPLS  524294   10    192.168.16.2 1100
192.0.2.4/32     ospf (0)  MPLS  524292   10    192.168.15.2 1100
192.0.2.5/32     ospf (0)  MPLS  524293   10    192.168.15.2 1000
192.0.2.6/32     ospf (0)  MPLS  524295   10    192.168.16.2 1000
192.168.15.2/32  ospf (0)  MPLS  524289   10    192.168.15.2 0
192.168.16.2/32  ospf (0)  MPLS  524290   10    192.168.16.2 0
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
```

The value of the metric is set to 16777215 (infinity – 1), because there is no CSPF and the head-end router is unaware of the full topology between head- and tail-end router.

PCC-initiated and computed LSP – loose path

Consider an LSP configured on PCC-2, with the tail end at PCC-1. There is a requirement for traffic on the LSP to pass through PE-6 before reaching PCC-1, so a loose path of PCC-2 to PE-6 before being forwarded to PCC-1 is required.

Figure 345: SR-TE LSP with loose path



26385

Figure 345: SR-TE LSP with loose path shows the concept of the loose path. The following configures the MPLS path containing a loose hop on PCC-2:

```
# on PCC-2:
configure {
  router "Base" {
    mpls {
      path "PCC-controlled-loose-path" {
        admin-state enable
        hop 1 {
          ip-address 192.0.2.6
          type loose
        }
      }
    }
  }
}
```

The SR-TE LSP configuration, which references the previously created MPLS path as the primary path, is as follows:

```
# on PCC-2:
configure {
  router "Base" {
    mpls {
      lsp "PCC-2-PCC-1-PCC-loose-lsp" {
        admin-state enable
        type p2p-sr-te
        to 192.0.2.1
        primary "PCC-controlled-loose-path" {
        }
      }
    }
  }
}
```

When enabled, the LSP path status becomes operationally up, as in the following output:

```
[/]
A:admin@PCC-2# show router mpls sr-te-lsp "PCC-2-PCC-1-PCC-loose-lsp" path detail

=====
MPLS SR-TE LSP PCC-2-PCC-1-PCC-loose-lsp
Path (Detail)
=====
Legend :
  S      - Strict                L      - Loose
  A-SID  - Adjacency SID        N-SID  - Node SID
  +      - Inherited
=====
-----
LSP SR-TE PCC-2-PCC-1-PCC-loose-lsp
Path PCC-controlled-loose-path
-----
LSP Name      : PCC-2-PCC-1-PCC-loose-lsp
Path LSP ID   : 31744
From          : 192.0.2.2
To            : 192.0.2.1
Admin State   : Up                Oper State    : Up
Path Name     : PCC-controlled-loose-path
Path Type     : Primary
Path Admin    : Up                Path Oper     : Up
Path Up Time  : 0d 00:00:22        Path Down Time : 0d 00:00:00
Retry Limit   : 0                  Retry Timer    : 30 sec
Retry Attempt : 0                  Next Retry In  : 0 sec

PathCompMethod : none                OperPathCompMethod: none
MetricType     : igp                  Oper MetricType  : igp
```

```

LocalSrProt      : preferred          Oper LocalSrProt : N/A
LabelStackRed   : Disabled           Oper LabelStackRed: N/A

Bandwidth       : No Reservation      Oper Bandwidth    : 0 Mbps
Hop Limit       : 255                 Oper HopLimit     : 255
Setup Priority   : 7                   Oper SetupPriority: 7
Hold Priority    : 0                   Oper HoldPriority : 0
Inter-area      : N/A

PCE Updt ID     : 0                   PCE Updt State   : None
PCE Upd Fail Code: noError

PCE Report      : Disabled+          Oper PCE Report   : Disabled
PCE Control     : Disabled          Oper PCE Control  : Disabled

Include Groups  :                     Oper IncludeGroups:
None                                                    None
Exclude Groups  :                     Oper ExcludeGroups:
None                                                    None
Last Resignal   : n/a

IGP/TE Metric   : 16777215           Oper Metric       : 16777215
Oper MTU        : 1556               Path Trans       : 1
Failure Code    : noError
Failure Node    : n/a
Explicit Hops   :
                192.0.2.6(L)
Actual Hops      :
    192.0.2.6(192.0.2.6) (N-SID)      Record Label      : 20006
    -> 192.0.2.1(192.0.2.1) (N-SID)  Record Label      : 20001

BFD Configuration and State
Template        : None                Ping Interval     : N/A
Enable         : False                State             : notApplicable
WaitForUpTimer : 4 sec                OperWaitForUpTimer: 0 sec
WaitForUpTmLeft : 0
StartFail Rsn  : N/A
=====

```

The Actual Hops in the MPLS path are the configured loose hop plus a hop for the destination of 192.0.2.1. Again, the configured hop addresses are translated into labels by the head-end PCC router, PCC-2, by examining the OSPF TE database. The hop-to-label translation always translates a loose hop to a node SID (N-SID).

The LSP is installed by the TTM into the tunnel table, alongside OSPF advertised shortest path tunnels, for use by the TTM users.

```

[/]
A:admin@PCC-2# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId Pref  Nexthop      Metric
Color
-----
192.0.2.1/32    sr-te    MPLS 655362  8    192.0.2.6   16777215
192.0.2.1/32    ospf (0)  MPLS 524291   10    192.168.23.2 2100
192.0.2.3/32    ospf (0)  MPLS 524292   10    192.168.23.2 1000
192.0.2.4/32    ospf (0)  MPLS 524293   10    192.168.24.2 1000
192.0.2.5/32    ospf (0)  MPLS 524294   10    192.168.24.2 1100

```

```

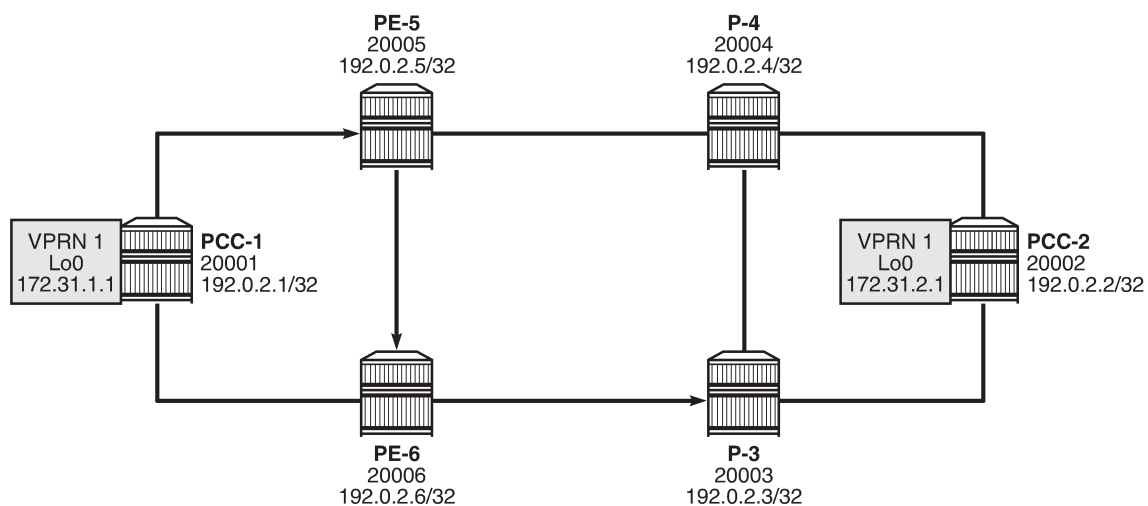
192.0.2.6/32      ospf (0) MPLS 524295 10 192.168.23.2 1100
192.168.23.2/32  ospf (0) MPLS 524289 10 192.168.23.2 0
192.168.24.2/32  ospf (0) MPLS 524290 10 192.168.24.2 0
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====

```

Service provisioning – VPRN

SR-TE tunnels are another MPLS tunnel type, and can be used in the context of **auto-bind-tunnel** for resolving BGP next hops for IPv4 routes within a VPRN.

Figure 346: VPRN service schematic



26386

Figure 346: VPRN service schematic shows a VPRN service, configured on PCC-1 and PCC-2. The following configures the VPRN 1 on PCC-1. It includes a local interface using a /32 loopback address, which will be used to verify that routing is working correctly.

```

# on PCC-1:
configure {
  service {
    vprn "VPRN 1" {
      admin-state enable
      service-id 1
      customer "1"
      autonomous-system 65545
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "65545:1"
          vrf-target {
            community "target:65545:1"
          }
        }
        auto-bind-tunnel {

```

```

        resolution filter
        resolution-filter {
            sr-te true
        }
    }
}
interface "loopback" {
    loopback true
    ipv4 {
        primary {
            address 172.31.1.1
            prefix-length 32
        }
    }
}
}
}

```



Note:

The **auto-bind-tunnel** command has the **resolution-filter** option set to **sr-te**, so that any BGP routes received will have the next-hop resolved to an SR-TE LSP. The VPRN configuration on PCC-2 also uses **auto-bind-tunnel sr-te**.

```

# on PCC-2:
configure {
    service {
        vprn "VPRN 1" {
            admin-state enable
            service-id 1
            customer "1"
            autonomous-system 65545
            bgp-ipvpn {
                mpls {
                    admin-state enable
                    route-distinguisher "65545:1"
                    vrf-target {
                        community "target:65545:1"
                    }
                }
                auto-bind-tunnel {
                    resolution filter
                    resolution-filter {
                        sr-te true
                    }
                }
            }
        }
    }
    interface "loopback" {
        loopback true
        ipv4 {
            primary {
                address 172.31.2.1
                prefix-length 32
            }
        }
    }
}
}
}

```

Examination of the VPRN route table shows that the route prefix representing the IP address of the loopback address configured in VPRN 1 is shown, and is resolved via the SR-TE tunnel.

```
[/]
A:admin@PCC-1# show router 1 route-table

=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]                Type   Proto   Age      Pref
  Next Hop[Interface Name]                Metric
-----
172.31.1.1/32                      Local  Local   00h02m01s  0
  loopback                            0
172.31.2.1/32                      Remote BGP VPN 00h00m57s 170
  192.0.2.2 (tunneled:SR-TE:655362)      16777215
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
=====
```

Connectivity is verified by sending a ping from the loopback interface within VPRN 1 on PCC-1 to the loopback address within VPRN 1 on PCC-2, as follows:

```
[/]
A:admin@PCC-1# ping 172.31.2.1 router-instance "VPRN 1" source-address 172.31.1.1
PING 172.31.2.1 56 data bytes
64 bytes from 172.31.2.1: icmp_seq=1 ttl=64 time=4.77ms.
64 bytes from 172.31.2.1: icmp_seq=2 ttl=64 time=4.92ms.
64 bytes from 172.31.2.1: icmp_seq=3 ttl=64 time=4.91ms.
64 bytes from 172.31.2.1: icmp_seq=4 ttl=64 time=4.68ms.
64 bytes from 172.31.2.1: icmp_seq=5 ttl=64 time=4.17ms.

---- 172.31.2.1 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 4.17ms, avg = 4.69ms, max = 4.92ms, stddev = 0.273ms
```

For completeness, a ping is sent in the opposite direction, between the PCC-2 VPRN 1 interface to PCC-1 VPRN 1, as follows:

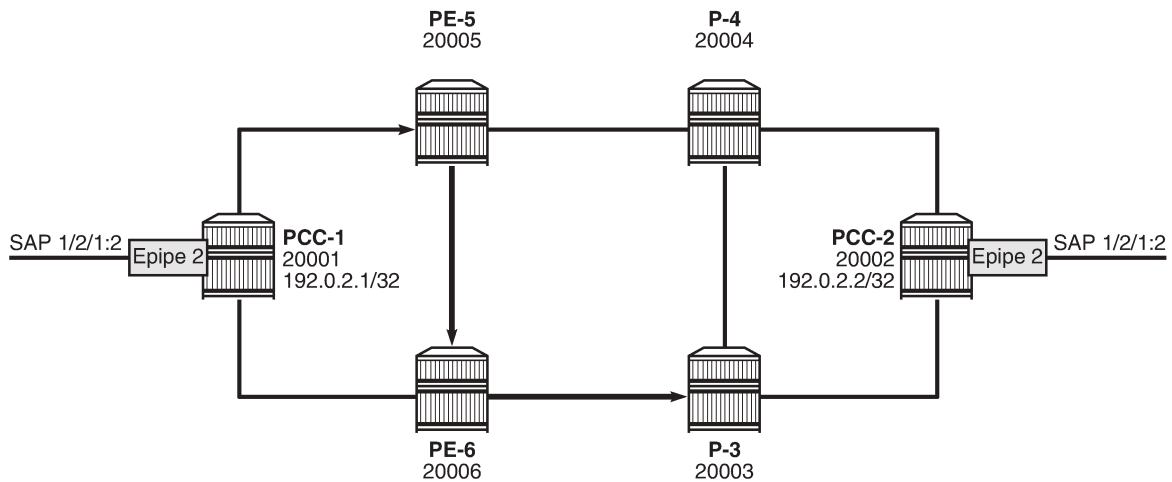
```
[/]
A:admin@PCC-2# ping 172.31.1.1 router-instance "VPRN 1" source-address 172.31.2.1
PING 172.31.1.1 56 data bytes
64 bytes from 172.31.1.1: icmp_seq=1 ttl=64 time=4.98ms.
64 bytes from 172.31.1.1: icmp_seq=2 ttl=64 time=5.42ms.
64 bytes from 172.31.1.1: icmp_seq=3 ttl=64 time=5.06ms.
64 bytes from 172.31.1.1: icmp_seq=4 ttl=64 time=4.99ms.
64 bytes from 172.31.1.1: icmp_seq=5 ttl=64 time=4.79ms.

---- 172.31.1.1 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 4.79ms, avg = 5.05ms, max = 5.42ms, stddev = 0.207ms
```

Layer 2 service provisioning – SR-TE

SR-TE tunnels can also be bound as a transport tunnel within SDPs. To illustrate this, consider the following example of a simple Epipe connected between PCC-1 and PCC-2, as shown in [Figure 347: Epipe service schematic](#).

Figure 347: Epipe service schematic



26387

Configure an SDP on PCC-1, with far end on PCC-2, and bind it to the previously configured SR-TE LSP:

```
# on PCC-1:
configure {
  service {
    sdp 12 {
      admin-state enable
      delivery-type mpls
      far-end {
        ip-address 192.0.2.2
      }
      lsp "PCC-1-PCC-2-PCC-strict-lsp" { }
    }
  }
}
```

Configure an Epipe on PCC-1:

```
# on PCC-1:
configure {
  service {
    epipe "Epipe 2" {
      admin-state enable
      service-id 2
      customer "1"
      spoke-sdp 12:2 {
      }
      sap 1/2/1:2 {
      }
    }
  }
}
```

Similarly, for PCC-2, configure an MPLS SDP and explicitly bind the SR-TE LSP, as follows:

```
# on PCC-2:
configure {
  service {
    sdp 21 {
      admin-state enable
      delivery-type mpls
      far-end {
        ip-address 192.0.2.1
      }
      lsp "PCC-2-PCC-1-PCC-loose-lsp" { }
    }
  }
}
```

Configure Epipe 2 on PCC-2, referencing the SDP as a spoke-SDP:

```
# on PCC-2:
configure {
  service {
    epipe "2" {
      admin-state enable
      service-id 2
      customer "1"
      spoke-sdp 21:2 {
      }
      sap 1/2/1:2 {
      }
    }
  }
}
```

Service verification

The state of SDP 12 on PCC-1 is shown in the following output:

```
[/]
A:admin@PCC-1# show service sdp

=====
Services: Service Destination Points
=====
SdpId  AdmMTU  OprMTU  Far End          Adm  Opr      Del  LSP  Sig
-----
12     0       1544    192.0.2.2        Up   Up       MPLS T    TLDP
-----
Number of SDPs : 1
-----
Legend: R = RSVP, L = LDP, B = BGP, M = MPLS-TP, n/a = Not Applicable
       I = SR-ISIS, 0 = SR-OSPF, T = SR-TE, F = FPE
=====
```

The output shows the LSP type as an SR-TE LSP - "T".

On PCC-1, the following output shows the base state of the Epipe service entities:

```
[/]
A:admin@PCC-1# show service id 2 base

=====
Service Basic Information
=====
```



```

Service Id       : 2                Vpn Id          : 0
Service Type    : Epipe
MACSec enabled  : no
Name            : Epipe 2
Description     : (Not Specified)
Customer Id     : 1                Creation Origin  : manual
Last Status Change: 04/08/2021 16:22:56
Last Mgmt Change : 04/08/2021 16:22:29
Test Service    : No
Admin State     : Up                Oper State      : Up
MTU             : 1514
Vc Switching   : False
SAP Count      : 1                SDP Bind Count  : 1
Per Svc Hashing : Disabled
Vxlan Src Tep Ip : N/A
Force QTag Fwd : Disabled
Oper Group     : <none>
    
```

Service Access & Destination Points

Identifier	Type	AdmMTU	OprMTU	Adm	Opr
sap:1/2/1:2	q-tag	1518	1518	Up	Up
sdp:12:2 S(192.0.2.2)	Spok	0	1544	Up	Up

Similarly, on PCC-2, the status of SDP 21 is as follows:

```

[/]
A:admin@PCC-2# show service sdp

=====
Services: Service Destination Points
=====
SdpId  AdmMTU  OprMTU  Far End          Adm  Opr          Del   LSP   Sig
-----
21     0       1552   192.0.2.1       Up  Up           MPLS  T     TLDP
-----
Number of SDPs : 1

Legend: R = RSVP, L = LDP, B = BGP, M = MPLS-TP, n/a = Not Applicable
        I = SR-ISIS, 0 = SR-OSPF, T = SR-TE, F = FPE
=====
    
```

The state of the Epipe service on PCC-2 is shown in the following output:

```

[/]
A:admin@PCC-2# show service id 2 base

=====
Service Basic Information
=====
Service Id       : 2                Vpn Id          : 0
Service Type    : Epipe
MACSec enabled  : no
Name            : Epipe 2
Description     : (Not Specified)
Customer Id     : 1                Creation Origin  : manual
Last Status Change: 04/08/2021 16:22:56
Last Mgmt Change : 04/08/2021 16:22:53
Test Service    : No
Admin State     : Up                Oper State      : Up
    
```

```
MTU : 1514
Vc Switching : False
SAP Count : 1
Per Svc Hashing : Disabled
Vxlan Src Tep Ip : N/A
Force QTag Fwd : Disabled
Oper Group : <none>
```

```
SDP Bind Count : 1
```

Service Access & Destination Points

Identifier	Type	AdmMTU	OprMTU	Adm	Opr
sap:1/2/1:2	q-tag	1518	1518	Up	Up
sdp:21:2 S(192.0.2.1)	Spok	0	1552	Up	Up

=====

Conclusion

Segment routing LSPs extend the use of MPLS labels into traffic engineering applications. This chapter provides the configuration for router instantiated and controlled SR-TE LSPs along with some examples of the application in a VPRN and Epipe. The chapter also shows the associated commands and outputs that can be used for verifying and troubleshooting.

Segment Routing over IPv6

This chapter provides information about Segment Routing over IPv6.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information and MD-CLI configuration in this chapter are based on SR OS Release 21.10.R1. Segment Routing over IPv6 (SRv6) is supported on FP4-based equipment in SR OS Release 21.5.R2 and later.

Overview

Segment Routing (SR) provides control over the forwarding paths without any need for path signaling, as described in chapter [Segment Routing with IS-IS Control Plane](#) for SR over IPv4. An SR tunnel contains a list of one or more segments. Each segment is identified by a segment identifier (SID). For SR over IPv4, the SIDs are MPLS labels from a configured SR-label range.

SRv6 provides IPv6 transport with both shortest path and source routing capabilities. SRv6 is a framework for the programmability of IPv6, which utilizes the large IPv6 address space. SRv6 data path encapsulation models each SID using a 128-bit IPv6 address, with differences for shortest-path routing and source routing.

In shortest-path routing, the destination SID is encoded in the Destination Address (DA) field of the outer IPv6 header, as shown in [Table 20: SRv6 shortest path routing](#).

Table 20: SRv6 shortest path routing

Header type	Parameter encoding
IPv6	Next header = IP SA = 2001:db8::2:1 DA= 2001:db8:aaaa:101:0:1000::
IP	Version 4, IHL=20 SA= 10.1.2.1 DA = 10.3.2.1 Protocol = UDP ...

In source routing, the SIDs of the nodes the packet must traverse are encoded as a SID list in the Segment Routing Header (SRH). The next SID in a segment list to forward the packet to is copied from the SRH into the DA field of the outer IPv6 header. The SID in the DA field determines the termination of the current segment. At the segment endpoint node, the next header (in this case, SRH) is examined and the next active SID is copied to the DA field. [Table 21: SRv6 source routing](#) shows an example with SRv6 source routed path segment list in the SRH.

Table 21: SRv6 source routing

Header type	Parameter encoding
IPv6	Next header = SRH SA = 2001:db8::2:1 DA= 2001:db8:aaaa:111:0:1000::
SRH	Segments left = 2 Segment 0 - 2001:db8:aaaa:102:0:1000:: Segment 1 - 2001:db8:aaaa:112:0:1000:: Segment 2 - 2001:db8:aaaa:111:0:1000:: Next Header = IP

SRv6 SID

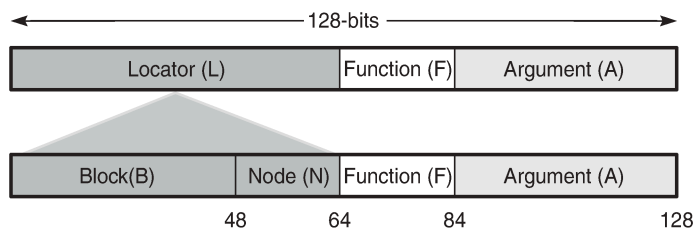
An SRv6 SID is a routable IPv6 prefix when it is set as the IPv6 header DA.



Note:
IPv6 router interface addresses are not SRv6 SIDs.

[Figure 348: SRv6 SID encoding](#) shows that the 128-bit address of an SRv6 SID is split into three constituent parts: locator, function, argument.

Figure 348: SRv6 SID encoding



37192

- The locator is a summary IPv6 prefix for a set of SIDs instantiated on an SRv6-capable router. The locator:
 - must be explicitly configured
 - is advertised using IS-IS
 - can be associated with a topology and/or Flex-Algorithm

- provides reachability to all SIDs originated by a router if the locator part of the SRv6 SID is routable
- comprises the L most significant bits of the SID, with L ranging from 4 to 96 bits
- has format B:N
 - All routers in a domain have the same block address B.
 - Each router in the domain has its own node-specific address N.
- The function is an opaque identification of a local behavior bound to the segment, as described in RFC 8986. [Table 22: SRv6 endpoint behaviors supported in SR OS Release 21.10.R1](#) lists the SRv6 endpoint behaviors supported in SR OS Release 21.10.R1.

Table 22: SRv6 endpoint behaviors supported in SR OS Release 21.10.R1

Function name	Role or behavior	Description
End	Endpoint	Equivalent to a node SID
End.X	Endpoint with an L3 cross-connect (X-connect)	Equivalent to an adjacency SID
LAN-End.X	Endpoint with an L3 cross-connect (X-connect)	Equivalent to an adjacency SID associated with a broadcast interface
End.DT4	De-encapsulate and perform an IPv4 table lookup	<ul style="list-style-type: none"> – VPRN table lookup: per-VRF SID for the VPN-IPv4 address family – Prefix lookup in the global IPv4 routing table
End.DT6	De-encapsulate and perform an IPv6 table lookup	<ul style="list-style-type: none"> – VPRN table lookup: per-VRF SID for the VPN-IPv6 address family – Prefix lookup in the global IPv6 routing table
End.DT46	De-encapsulate and perform IPv4 and IPv6 table lookups	<ul style="list-style-type: none"> – VPRN table lookup: both IPv4 and IPv6 - equivalent to per-VRF label – VPN-IPv4 and VPN-IPv6 routes are advertised with a single label in the same VRF

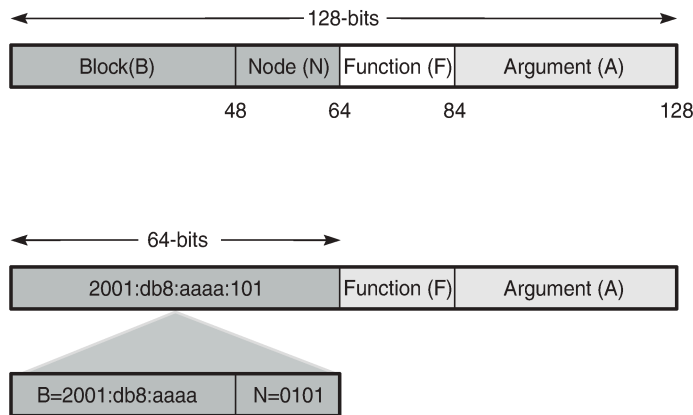
- The argument, which is not a configurable field in SR OS Release 21.10.R1, is set to all zeros.

[Figure 349: SRv6 SID encoding example](#) shows an example of an SRv6 SID with the following:

- B = 48 bits
- N = 16 bits
- L = B + N = 48 bits + 16 bits = 64 bits

- F = 20 bits
- The remaining 44 bits (A) are set to zero.

Figure 349: SRv6 SID encoding example



37193

The /64 locator part for a set of routers in a routing domain consists of:

- a common 48-bit block, for example, 2001:db8:aaaa::/48
- a unique 16-bit node identifier allocated in the range from 0000 to ffff

Some examples:

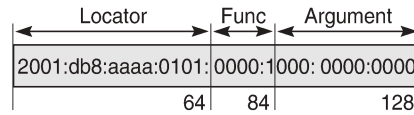
- locator for PE-1 = 2001:db8:aaaa:101::/64
- locator for PE-2 = 2001:db8:aaaa:102::/64
- locator for PE-3 = 2001:db8:aaaa:103::/64

The local router installs the locator in its IPv6 route table and FIB. The locator prefix is advertised in IS-IS in the SRv6 locator sub-TLV. Each remote router populates its route table and FIB with the locator prefixes, including the tunneled next-hop to the originating router.

The function field has a configurable length, ranging from 20 to 96 bits. By default, the function field has 20 bits. The function field is used to assign End and End.X SIDs, which are used by remote routers to create repair tunnels for remote and topology-independent loopfree-alternate (RLFA and TI-LFA) backup paths.

- An End function is statically configured in SR OS:
 - By default, the number of static functions is 1.
 - For example, the End function with value 1 in the 20-bit format is represented as 00001 in hexadecimal, followed by the zeros of the argument field.
 - The End SID (node SID) for PE-1 equals 2001:db8:aaaa:101:0:1000::/128, as shown in [Figure 350: End SID for PE-1](#)

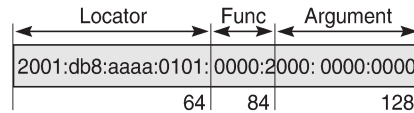
Figure 350: End SID for PE-1



37194

- The End.X function can be statically configured or automatically assigned by the system.
 - In case of static configuration, the number of static functions must be increased.
 - For the function with value 2 in a function field of 20 bits, the corresponding hexadecimal pattern is 00002, followed by the zeros of the argument field.
 - The End.X SID (adjacency SID) for PE-1 equals 2001:db8:aaaa:101:0:2000::/128, as shown in [Figure 351: End.X SID for PE-1](#).

Figure 351: End.X SID for PE-1



37195

IPv6 header and SRH

This section describes how source routing works with the insertion of an SRH.



Note:

SR OS Release 21.10 has no mechanism for computing a source-routed path for normal data traffic flow; for example, there is no equivalent to the SR-TE or SR-policy label stack for source routing. The use of the SRH is restricted to repair tunnels computed by the TI-LFA process. When a link or node failure occurs, the Point of Local Repair (PLR) inserts an appropriate SRH for SRv6 traffic that is to be routed around the failure during IGP convergence.

Different SR node types are defined: source node, transit node, and segment endpoint node. To enable source routing on the IPv6 source router, the SRH contains an ordered list of one or more SRv6 SIDs.

[Figure 352: IPv6 header defined in RFC 8200](#) shows the IPv6 header where the next header field must be coded as 43 when the IPv6 extension header, which follows the IPv6 header, is an SRH.

Figure 352: IPv6 header defined in RFC 8200

Traffic class	Flow label	
Payload length	Next header	Hop limit
Source address		
Destination address		

37196

Figure 353: Position of the SRH in the protocol stack shows that the header following the IPv6 header sits between the IPv6 header and upper layer protocols, such as TCP or UDP.

Figure 353: Position of the SRH in the protocol stack

IPv6 header
SRH
Upper layer protocol (TCP, UDP ...)

37197

Figure 354: SRH defined in RFC 8754 shows the SRH.

Figure 354: SRH defined in RFC 8754

Next header	HDR extension length	Routing type	Segments left
Last entry	Flags	Tag	
Segment list [0]			
Segment list [1]			
....			
Segment list [n]			
Optional TLVs			

37198

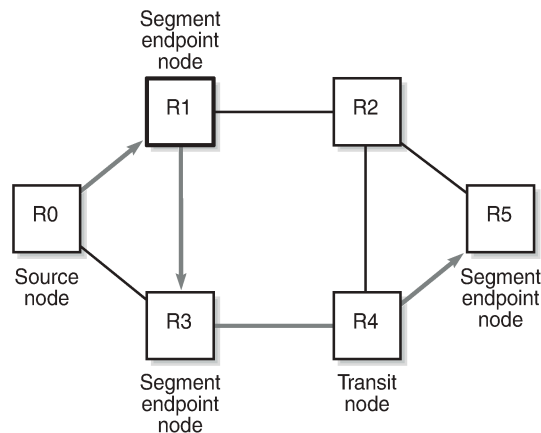
The SRH is derived from the IPv6 routing header as defined in RFC 8200. The SRH fields are:

- Next header: defines the type of header following SRH, for example, TCP or UDP.
- Routing type for SRH: 4.
- Segments left: the number of explicitly listed intermediate nodes still to be traversed before reaching the final destination.

- Last entry: contains the zero-based index of the last element of the segment list.
- Segment list [n]: a 128-bit IPv6 address representing the nth segment in the segment list. The segment list is encoded in reverse numerical order: segment list [0] is the first element in the segment list and contains the last segment of the SR path, segment list [1] contains the penultimate segment of the SR path, and so on.

Figure 355: SRv6 node types shows the SR node types: source node, transit node, and segment endpoint node for an SRv6 packet flow from R0 to R5 via hops R1, R3, and R5.

Figure 355: SRv6 node types



37199

The intermediate hops R1, R3, and R5 are programmed in the segment list of the SRH.

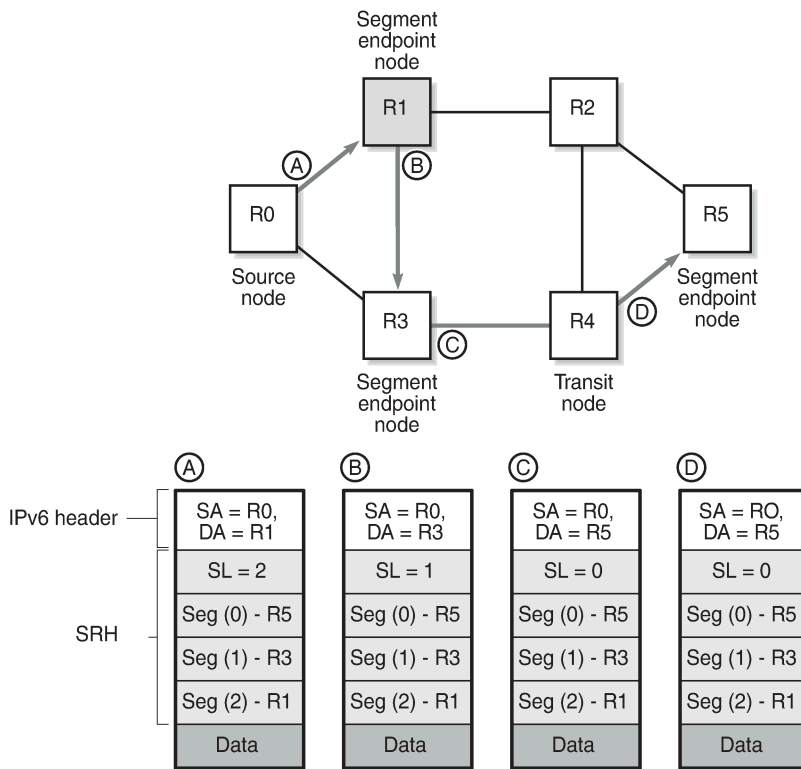
The SRv6 node types defined in RFC 8754 are:

- SR source node
 - Any node that originates an IPv6 packet with a segment (that is, an SRv6 SID) in the DA field of the IPv6 header.
 - The IPv6 packet leaving the SR source node may or may not contain an SRH. This includes either:
 - a host originating an IPv6 packet
 - an SR domain ingress router encapsulating a received packet in an outer IPv6 header, followed by an optional SRH
 - In this example, R0 acts as an SR source node and includes an SRH containing a segment list.
- SR transit node
 - Any node forwarding an IPv6 packet where the DA of the packet is not locally configured as a segment or a local interface. A transit node need not be capable of processing a segment or SRH.
 - In this example, R4 acts as an SR transit node. It forwards the SRv6 packet without processing the SRH.
- SR segment endpoint node
 - Any node receiving an IPv6 packet where the DA of that packet is locally configured as a segment or local interface.

- In this example, R1, R3, and R5 are SR segment endpoint nodes. These nodes interrogate the SRH as part of packet processing.

Figure 356: Data forwarding of SRv6 encapsulated packets using SRv6 SIDs shows the data forwarding of SRv6 encapsulated packets using SRv6 SIDs at R0 and R1.

Figure 356: Data forwarding of SRv6 encapsulated packets using SRv6 SIDs



37200

Source node R0 tunnels an SRv6 packet to destination R5, segment (0), in the SRH.

- The segment list contains SRv6 SIDs associated with each hop, such as the End SID. The first segment endpoint is the last segment in the list, segment (2) in the example. The Segments Left (SL) field is set to a value matching the highest segment list number (2).
- SRH is only used by routers where the DA is equal to a local address. The IPv6 source address is set to the local IPv6 address of R0. The IPv6 DA in the IPv6 header is set to the segment list entry indexed in the SL field; in this case, R₁.
- The packet is forwarded to R1.

At R₁, the incoming packet has the IPv6 DA matching R1.

- R1 removes the IPv6 header and processes the SRH. The SL is decremented to SL 1, which corresponds to segment (1) = R3.
- R1 adds an IPv6 header with DA equal to the SID for R3.
- R1 forwards the packet to R3.

At R₃, the incoming packet has the IPv6 DA matching R3.

- R3 removes the IPv6 header and processes the SRH. The SL is decremented to SL 0, which corresponds to segment (0) = R5.
- R3 adds an IPv6 header with DA equal to the SID for R5.
- R3 forwards the packet to R5.

At R4, the incoming packet has the IPv6 DA matching R5, so the packet is forwarded to R5 without processing the SRH header and without changing the IPv6 DA.

At R5, the incoming packet has the IPv6 DA matching R₅, so the IPv6 header is removed and the SRH header is processed. The SL value 0 cannot be decreased anymore, so R5 removes the SRH and the packet is sent for further processing, for example, to a particular VPRN.



Note:

The IPv6 SID at segment (0) may contain an opaque behavior value (function) that indicates to the destination node that further processing is required, such as a VPRN table lookup.

Data path support: forwarding path extensions

SRv6 data traffic requires additional processing at both the ingress and egress data planes. This processing is performed via an internal cross connect in the form of port cross-connect (PXC) ports. SRv6 traffic is steered from the input to a PXC port, where it is internally looped for additional processing.

The PXC port is associated with the SRv6 application using a Forwarding Path Extension (FPE). An origination (egress) FPE and termination (ingress) FPE are associated with SRv6. The additional processing in the SRv6 data path is as follows:

- Ingress PE node
 - At the origination FPE data path, L2 and L3 service packets are received and the SRv6 encapsulation header is pushed for the primary path and for the backup path based on the index passed by the service context in the internal packet header.
 - The hop-limit field in the outer IPv6 header of the SRv6 tunnel is set to 255.
 - At the termination FPE data path, a lookup is done on the DA field in the outer IPv6 header, and the packet is forwarded to one of the candidate egress network IP interfaces based on a hash of the flow label and SA/DA fields of the outer IPv6 packet header.
- Egress PE node
 - At the origination FPE data path of the incoming router interface, a longest prefix match of the DA in the outer IPv6 header is performed in the SRv6 SID FIB.
 - If there is a match against a local locator prefix, the packet is forwarded to the termination FPE for service SID termination processing.
 - The termination FPE does an Ingress Label Map (ILM) lookup on the service label and forwards the packet to the service context for further processing.
 - At the origination FPE data path, the SRH is processed. The SRv6 encapsulation is removed, and a service label is inserted into the inner packet with label value derived from the function field.
- Transit router
 - Transit routers do not require FPEs.
 - Transit routers receiving SRv6-encapsulated packets make forwarding decisions based on the IPv6 route table lookups.

SRH processing modes

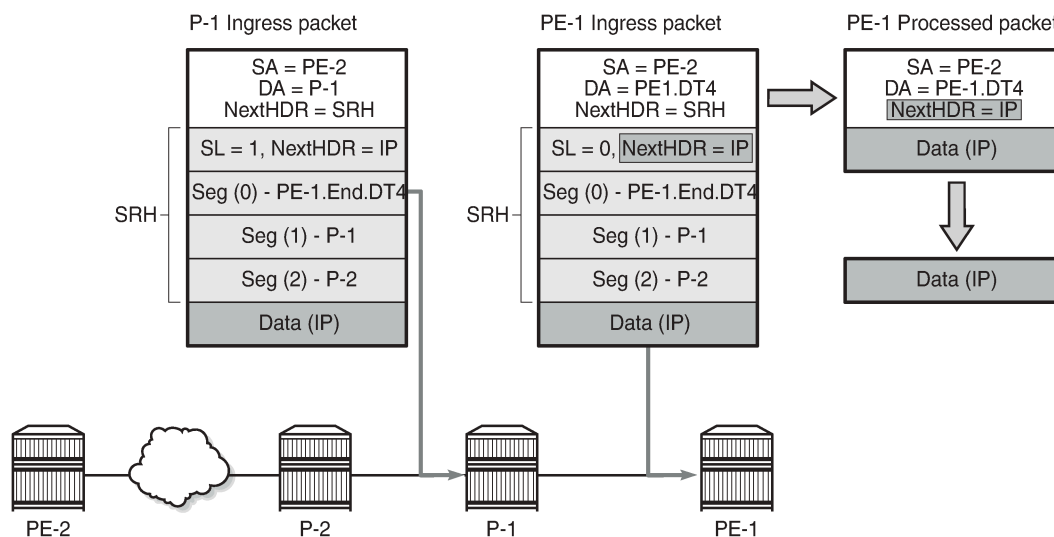
SR OS supports two SRH processing modes at the end of the SRv6 tunnel:

- Ultimate SRH Pop (USP), where the ultimate SR segment endpoint node processes and removes the SRH
- Penultimate SRH Pop (PSP), where the penultimate SR segment endpoint node processes and removes the SRH

USP mode

In the following example, source node PE-2 sends a packet to SR segment endpoint node PE-1 via intermediate hops P-2 and P-1. [Figure 357: USP mode - egress router PE-1 processes and removes SRH](#) shows how penultimate SR segment endpoint node P-1 and ultimate SR segment endpoint node PE-1 process the SRH in the packet.

Figure 357: USP mode - egress router PE-1 processes and removes SRH



37201

Source node PE-2 sends a packet with SRH with three segments in the segment list: Seg(2) for P-2, Seg(1) for P-1, and Seg(0) for destination PE-1. P-1, P-2, and PE-1 are SR segment endpoint nodes. Penultimate SR segment endpoint node P-1 decrements the value in the SL field in the SRH from 1 to 0 and copies the PE-1 SID from segment 0 into the IPv6 header DA. Ultimate SR segment endpoint node PE-1 receives the packet with the DA equal to PE-1.End.DT4 and SL 0 and processes the packet by:

- "updating the Next Header field in the IPv6 header with the Next Header field of the SRH
- "removing the SRH from the IPv6 extension header chain
- "processing the next header in the packet, which is achieved using the origination FPE data path



Note:

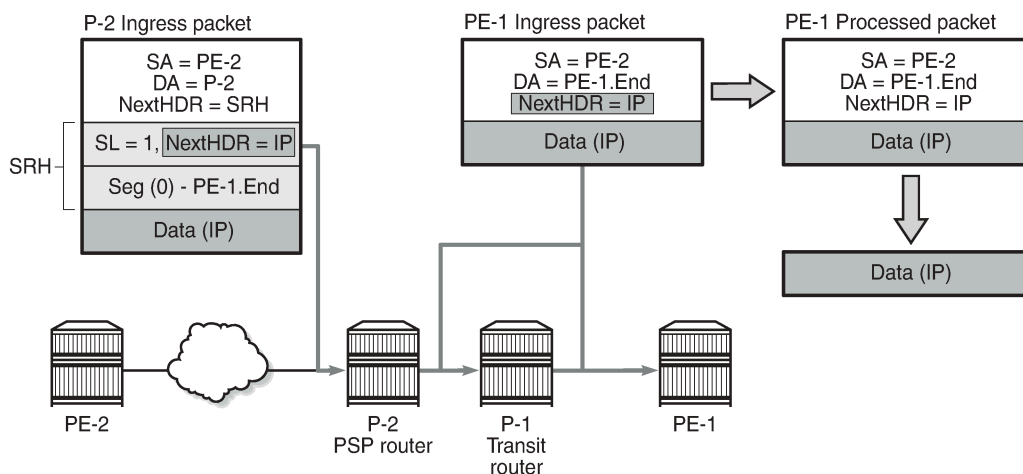
In this example, the IPv6 SID at segment (0) "PE-1.End.DT4" contains a function indicating to the destination node that a VPRN table lookup is required.

PSP mode

As stated in RFC 8986, a penultimate SR segment endpoint node is one that, as part of the SID processing, copies the last SID from the SRH into the IPv6 DA and decrements the SL value from one to zero.

Figure 358: Penultimate SRH hop P-2 processes and removes the SRH shows how penultimate SR segment endpoint node P-2 processes the packet toward PE-1. P-1 is an SR transit node in this example, so it does not process an SRH.

Figure 358: Penultimate SRH hop P-2 processes and removes the SRH



37202

The PSP operation is controlled by the SR source node. SR source node PE-2 is aware that the PE-1.End SID has SRH mode PSP. PE-2 sends a packet to PE-1 with the DA set to P-2 in the IPv6 header. The SRH contains one SID in the segment list: Seg(0) PE-1.End. The SL is set to 1.

Penultimate SR segment endpoint node P-2 processes the packet by:

- decrementing the IPv6 hop limit by 1
- decrementing the SL by 1, so SL = 0
- updating the IPv6 DA with the PE-1.End node SID from the segment list
- updating the Next Header field in the IPv6 header to the Next Header field of the SRH
- removing the SRH from the IPv6 extension header chain
- submitting the packet to the MPLS engine for transmission

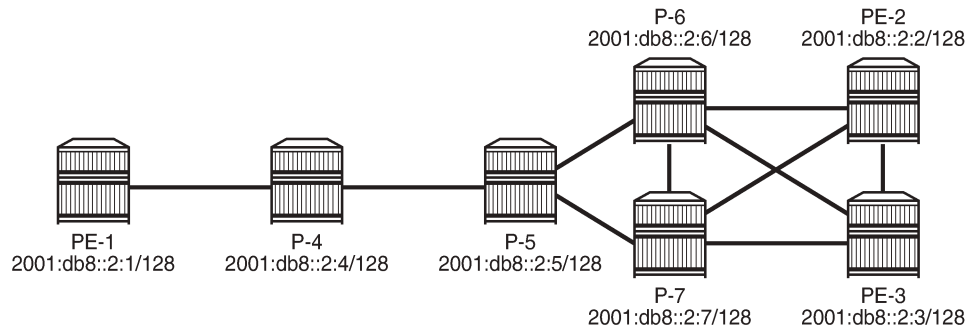
At transit node P-1, the packet is forwarded based on the RTM lookup for IPv6 DA in the IPv6 header.

At the destination node PE-1, the IPv6 header is removed and additional processing of the next header in the packet is done via an origination FPE data path.

Configuration

Figure 359: Example topology shows the example topology with seven SRv6-capable routers (with FP4).

Figure 359: Example topology



37203

PXC

SRv6 traffic is steered from input to a PXC port, where it is internally looped for additional processing. PXC can use either an internally looped physical port or an internal loopback in the FP4 MAC chip.

In case of an internally looped physical port, configure PXC on the physical port, as shown in the following example for PXC 5 on physical port 1/1/c5/1:

```
# on all SRv6-capable nodes:
configure {
  port-xc {
    port-xc {
      pxc 5 {
        admin-state enable
        port-id 1/1/c5/1
      }
    }
  }
  port pxc-5.a {
    admin-state enable
  }
  port pxc-5.b {
    admin-state enable
  }
  port 1/1/c5 {
    admin-state enable
    connector {
      breakout c4-10g
    }
  }
  port 1/1/c5/1 {
    admin-state enable
    ethernet {
      mode hybrid
      dot1x {
        tunneling true
      }
    }
  }
}
}
```

In case of internal loopbacks in the FP4 MAC chip, map PXC 1 and PXC 2 to internal loopbacks. It is possible to map PXC 1 and PXC 2 to the same loopback on the same MAC chip, but that is not configured here.

```
# on all SRv6-capable nodes:
configure {
  card 1 {
    card-type xcm-2s
    mda 1 {
      mda-type s36-100gb-qsfp28
      xconnect {
        mac 1 {
          loopback 1 {
          }
          loopback 2 {
          }
        }
      }
    }
  }
  port-xc {
    pxc 1 {
      admin-state enable
      port-id 1/1/m1/1
    }
    pxc 2 {
      admin-state enable
      port-id 1/1/m1/2          # or loopback 1/1/m1/1 (same as PXC 1)
    }
  }
  port pxc-1.a {
    admin-state enable
  }
  port pxc-1.b {
    admin-state enable
  }
  port pxc-2.a {
    admin-state enable
  }
  port pxc-2.b {
    admin-state enable
  }
  port 1/1/m1/1 {
    admin-state enable
  }
  port 1/1/m1/2 {
    admin-state enable
  }
}
```

There are several MAC chips per FP4-complex (hardware dependent). The operator configures the location of the loopback. The PXC loopback must be referenced as a port ID to enable loopback. The following **show datapath** command includes the internal loopbacks 1/1/m1/1 and 1/1/m1/2:

```
[/]
A:admin@PE-2# show datapath 1 detail

=====
Card    [X/IOM/]MDA  FP  TAP  MAC Chip Num  Connector  Port
-----
1       1             1  1    1             c1         1/1/c1/1
1       1             1  1    1             c1         1/1/c1/2
1       1             1  1    1             c1         1/1/c1/3
```

```

1      1      1  1  1      c1      1/1/c1/4
---snip---

1      1      2  1  6      c36      1/1/c36/1
1      1      2  1  6      c36      1/1/c36/2
1      1      2  1  6      c36      1/1/c36/3
1      1      2  1  6      c36      1/1/c36/4
1      1      1  1  1      N/A      1/1/m1/1
1      1      1  1  1      N/A      1/1/m1/2
=====

```

In this example, PXC loopbacks are configured on MDA 1/1, which has two MAC chips with MAC chip numbers m1 and m2. The two internal loopbacks are configured on MAC chip number m1.



Note:

Nokia recommends selecting cards and MAC chips connected to faceplate ports with lower bandwidth utilization for internal PXC.

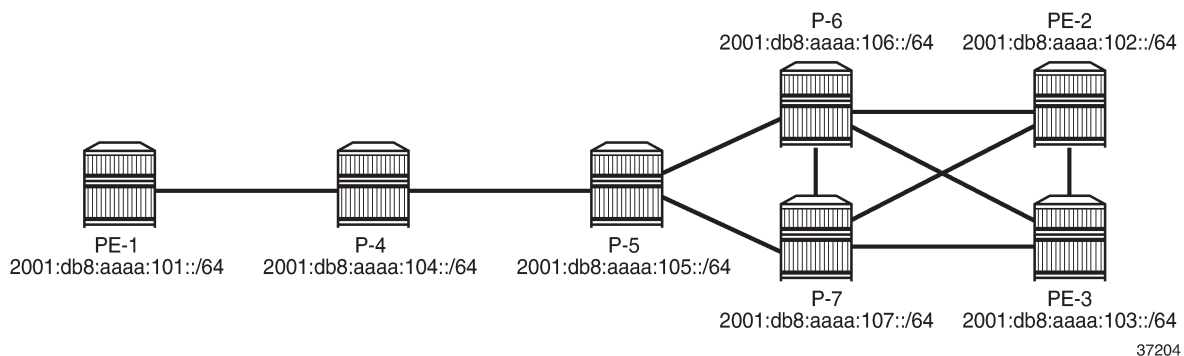
In this example, all nodes can act as SR segment endpoint nodes; there are no SR transit nodes. Perform the following steps to enable SRv6 on the nodes:

1. Allocate an address block B for all routers in a domain; for example, 2001:db8:aaaa::/48.
2. Allocate a unique node address N for each router; for example, 0101 for PE-1.
3. Configure a locator for each router in the format B:N:: and set the prefix length of the locator; for example, /64.
4. Add FPE to configure the data path.
5. Configure the End function (the SRv6 equivalent for node SID) for each router locator.
6. Configure the End.X functions (the SRv6 equivalent for adjacency SIDs) for each router associated with locator.
7. Advertise the locator in IS-IS level 1 or 2, as required.

Locator B:N::

Figure 360: SRv6 router locator prefixes shows the router locator prefixes for the seven nodes in the sample topology.

Figure 360: SRv6 router locator prefixes



Configure the SRv6 address block B and the locator prefix on the nodes. The following example shows SRv6 address block B 2001:db8:aaaa::/48 and locator prefix 2001:db8:aaaa:101::/64 in the dedicated **segment-routing-v6** context on PE-1:

```
# on PE-1:
configure {
  router "Base" {
    segment-routing {
      segment-routing-v6 {
        locator "PE-1_loc" {
          admin-state enable
          block-length 48
          prefix {
            ip-prefix 2001:db8:aaaa:101::/64
          }
        }
      }
    }
  }
}
```

The configuration on the other nodes is similar with the locator prefixes as shown in [Figure 360: SRv6 router locator prefixes](#).

FPE

SRv6 packet processing requires an ingress (termination) FPE and an egress (origination) FPE. FPE 1 is configured as **srv6>type origination**; FPE 2 as **srv6>type termination**. FPE 1 is configured as **origination-fpe** in the global **segment-routing-v6** context; FPE 2 is configured as **termination-fpe** in the **locator** context. On PE-1, the configuration is as follows:

```
# on PE-1:
configure {
  fwd-path-ext {
    fpe 1 {
      path {
        pxc 1
      }
      application {
        srv6 {
          type origination
        }
      }
    }
    fpe 2 {
      path {
        pxc 2
      }
      application {
        srv6 {
          type termination
        }
      }
    }
  }
  router "Base" {
    segment-routing {
      segment-routing-v6 {
        origination-fpe [1]
        source-address 2001:db8::2:1
        locator "PE-1_loc" {
          admin-state enable
        }
      }
    }
  }
}
```

```
        block-length 48
        termination-fpe [2]
        prefix {
            ip-prefix 2001:db8:aaaa:101::/64
        }
    }
}
```

The configuration on the other nodes is similar.

The following command for FPE 1 shows that SRv6 is enabled and operationally up and the SRv6 type is origination:

```
[/]
A:admin@PE-1# show fwd-path-ext fpe 1

=====
FPE Id: 1
=====
Description      : (Not Specified)
Multi-Path       : Disabled
Path             : pxc 1
Pw Port          : Disabled           Oper    : down
Sub Mgmt Extension : Disabled         Oper    : N/A
Vxlan Termination : Disabled         Oper    : down
Segment-Routing V6 : Enabled         Oper   : up
SRv6 Type         : origination
If-A Qos Policy  : default
If-B MTU         : 9786 bytes         Oper MTU : 1556 bytes
If-B Qos Policy  : default
=====
```

The following command for FPE 2 shows that SRv6 is enabled and operationally up and the SRv6 type is termination:

```
[/]
A:admin@PE-1# show fwd-path-ext fpe 2

=====
FPE Id: 2
=====
Description      : (Not Specified)
Multi-Path       : Disabled
Path             : pxc 2
Pw Port          : Disabled           Oper    : down
Sub Mgmt Extension : Disabled         Oper    : N/A
Vxlan Termination : Disabled         Oper    : down
Segment-Routing V6 : Enabled         Oper   : up
SRv6 Type         : termination
If-A Qos Policy  : default
If-B MTU         : 0 bytes           Oper MTU : 1556 bytes
If-B Qos Policy  : default
=====
```

The following command on PE-1 shows the associations for FPE 1. FPE 1 is an origination FPE, so it is not associated with a locator.

```
[/]
A:admin@PE-1# show fwd-path-ext associations fpe 1

=====
```

```
Segment-routing V6 associations
=====
Srv6
-----
Origination-fpe
=====
Segment-routing V6 Locator associations
=====
Locator
-----
=====
```

The following command on PE-1 shows the associations for FPE 2. FPE 2 is a termination FPE associated with locator "PE-1_loc".

```
[/]
A:admin@PE-1# show fwd-path-ext associations fpe 2

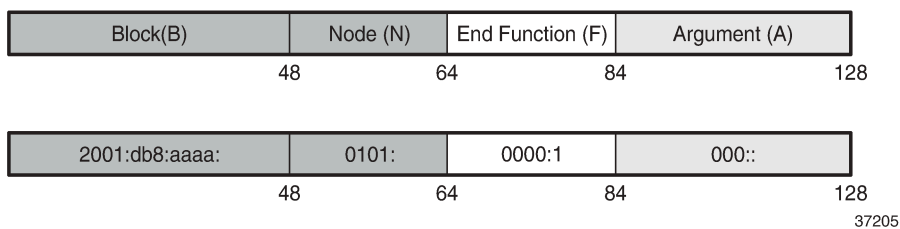
Segment-routing V6 associations
=====
Srv6
-----
Segment-routing V6 Locator associations
=====
Locator
-----
PE-1_loc
=====
```

Functions

End function

The SRv6 End function is configured in the SRv6 End SID that is the equivalent for IPv4 node SIDs. [Figure 361: SRv6 End SID on PE-1](#) shows an example with End function value 1 on PE-1:

Figure 361: SRv6 End SID on PE-1



The SRv6 End function is statically configured in the **segment-routing-v6>base-routing-instance>locator** context, as follows:

```
# on PE-1:
configure {
  router "Base" {
    segment-routing {
      segment-routing-v6 {
        locator "PE-1_loc" {
          admin-state enable
          block-length 48
          # function-length 20          # default value 20
        }
        base-routing-instance {
          locator "PE-1_loc" {
            function {
              end 1 {                  # function value = 1
                srh-mode usp          # Ultimate SRH Pop (default: PSP)
              }
            }
          }
        }
      }
    }
  }
}
```

The configuration on the other nodes is similar.

By default, the **function-length** is 20. The value **function>end 1** defines a function value of 1 inserted into the 20-bit function field. The **srh-mode** determines whether USP or PSP mode is used to process and remove the SRH. The default SRH mode is PSP.

A node can have one or two End functions. If both PSP and USP modes are used, a unique End function can be configured for each SRH mode. This requires the increase of the number static functions allowed, because the default value is 1; **static-function>max-entries** is configured for this purpose. As an example, this is configured on PE-1 only:

```
# on PE-1:
configure {
  router "Base" {
    segment-routing {
      segment-routing-v6 {
        locator "PE-1_loc" {
          static-function {
            max-entries 2              # 2 static functions (end 1, end 2)
          }
        }
        base-routing-instance {
          locator "PE-1_loc" {
            function {
              end 1 {                  # function value = 1
                srh-mode usp          # SRH mode - Ultimate SRH Pop
              }
              end 2 {                  # function value = 2
                # default SRH mode - Penultimate SRH Pop
              }
            }
          }
        }
      }
    }
  }
}
```

The following command shows the End SID values for the locators in the base routing instance on PE-1:

```
[/]
```

```
A:admin@PE-1# show router segment-routing-v6 local-sid end
=====
Segment Routing v6 Local SIDs
=====
SID                               Type      Function
Locator
Context
-----
2001:db8:aaaa:101:0:1000::       End       1
PE-1_loc
Base
2001:db8:aaaa:101:0:2000::       End       2
PE-1_loc
Base
-----
SIDs : 2
=====
```

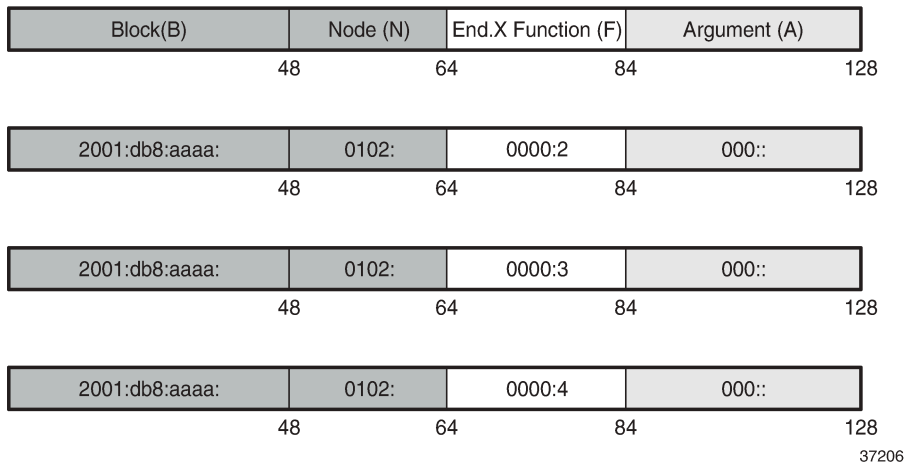
The following command on PE-1 shows the End SIDs plus SRH mode for the locators in the base routing instance:

```
[/]
A:admin@PE-1# show router segment-routing-v6 base-routing-instance end
=====
Segment Routing v6 Base Routing Instance
=====
Locator                               Type      Function      SID                               Status/InstId
SRH-mode Protection Interface
-----
PE-1_loc
End                                     1          2001:db8:aaaa:101:0:1000::       ok
USP
End                                     2          2001:db8:aaaa:101:0:2000::       ok
PSP
-----
Auto-allocated End.X:
-----
-----
Legend: * - System allocated
```

End.X function

The SRv6 End.X SID is the equivalent to IPv4 adjacency SIDs. [Figure 362: SRv6 End.X SIDs on PE-2](#) shows an example with End.X function values 2, 3, and 4 on PE-2:

Figure 362: SRv6 End.X SIDs on PE-2



End.X function SIDs can be allocated dynamically or configured as static SIDs.

For dynamically-allocated End.X function SIDs, the configuration is as follows:

```
# on PE-2:
configure {
  router "Base" {
    segment-routing {
      segment-routing-v6 {
        base-routing-instance {
          base-routing-instance {
            locator "PE-2_loc" {
              function {
                end-x-auto-allocate usp protection protected { }
              }
            }
          }
        }
      }
    }
  }
  isis 0 {
    advertise-router-capability area
    segment-routing-v6 {
      admin-state enable
      locator "PE-2_loc" {
        level-capability 2
        level 2 {
          metric 10
        }
      }
    }
  }
  ---snip---
```

PE-2 has three neighbors, so three End.X functions are automatically allocated. Each End.X SID is associated with a locator. In this example, the number of static functions is 1 and the automatically allocated End.X functions get values 2, 3, and 4. The PSP and USP protection modes specify whether the link is eligible for xLFA protection.



Note:

If TI-LFA is enabled, the protection mode must be set to Protected for the IGP to generate an End.X SID.

The following commands on PE-2 shows all local SIDs, including the End SID as well as End.X SIDs.

```
[/]
A:admin@PE-2# show router segment-routing-v6 local-sid

=====
Segment Routing v6 Local SIDs
=====
SID                                     Type      Function
Locator Context
-----
2001:db8:aaaa:102:0:1000::            End       1
PE-2_loc
Base
2001:db8:aaaa:102:0:2000::            End.X     2
PE-2_loc
None
2001:db8:aaaa:102:0:3000::            End.X     3
PE-2_loc
None
2001:db8:aaaa:102:0:4000::            End.X     4
PE-2_loc
None
-----
SIDs : 4
-----
=====
```

```
[/]
A:admin@PE-2# show router segment-routing-v6 base-routing-instance

=====
Segment Routing v6 Base Routing Instance
=====
Locator
Type      Function      SID                               Status/InstId
SRH-mode Protection Interface
-----
PE-2_loc
End       1 2001:db8:aaaa:102:0:1000::      ok
USP
-----
Auto-allocated End.X: USP Protected,
-----
End.X     *2 2001:db8:aaaa:102:0:2000::      0
USP       Protected int-PE-2-PE-3
ISIS Level: L2 Mac Address: 02:18:01:01:00:0b Nbr Sys Id: 1920.0000.2003
End.X     *3 2001:db8:aaaa:102:0:3000::      0
USP       Protected int-PE-2-P-6
ISIS Level: L2 Mac Address: 02:24:01:01:00:01 Nbr Sys Id: 1920.0000.2006
End.X     *4 2001:db8:aaaa:102:0:4000::      0
USP       Protected int-PE-2-P-7
ISIS Level: L2 Mac Address: 02:28:01:01:00:15 Nbr Sys Id: 1920.0000.2007
-----
Legend: * - System allocated
```

The End.X function can be created as a static SID, persistent through a reboot or link flap. The maximum number of static functions must be increased because additional static entries are required: one for each neighbor of PE-3, as follows:

```
# on PE-3:
configure {
  router "Base" {
    segment-routing {
      segment-routing-v6 {
        locator "PE-3_loc" {
          static-function {
            max-entries 4          # 1 End function + 3 End.X functions
          }
        }
        base-routing-instance {
          locator "PE-3_loc" {
            function {
              end 1 {
                srh-mode usp
              }
              end-x 2 {
                srh-mode usp
                interface-name "int-PE-3-PE-2"
              }
              end-x 3 {
                srh-mode usp
                interface-name "int-PE-3-P-6"
              }
              end-x 4 {
                srh-mode usp
                interface-name "int-PE-3-P-7"
              }
            }
          }
        }
      }
    }
  }
}
```

The following commands show the configured End.X SIDs on PE-3:

```
[/]
A:admin@PE-3# show router segment-routing-v6 local-sid end-x

=====
Segment Routing v6 Local SIDs
=====
```

SID	Type	Function
2001:db8:aaaa:103:0:2000:: PE-3_loc Base	End.X	2
2001:db8:aaaa:103:0:3000:: PE-3_loc Base	End.X	3
2001:db8:aaaa:103:0:4000:: PE-3_loc Base	End.X	4

```
-----
SIDs : 3
-----
```



```

=====
[/]
A:admin@PE-3# show router segment-routing-v6 base-routing-instance end-x
=====
Segment Routing v6 Base Routing Instance
=====
Locator
  Type      Function      SID      Status/InstId
  SRH-mode Protection Interface
-----
PE-3_loc
  End.X          2 2001:db8:aaaa:103:0:2000::      ok
  USP      Protected int-PE-3-PE-2
  End.X          3 2001:db8:aaaa:103:0:3000::      ok
  USP      Protected int-PE-3-P-6
  End.X          4 2001:db8:aaaa:103:0:4000::      ok
  USP      Protected int-PE-3-P-7
-----
Auto-allocated End.X:
-----
=====

```

SRv6 configuration summary example

The following summarizes the SRv6 configuration on PE-2:

```

# on PE-2:
configure {
  card 1 {
    card-type xcm-2s
    mda 1 {
      mda-type s36-100gb-qsfp28
      xconnect {
        mac 1 {
          loopback 1 {          # create internal MAC-chip loopback
          }
          loopback 2 {
          }
        }
      }
    }
  }
  port pxc-1.a {
    admin-state enable
  }
  port pxc-1.b {
    admin-state enable
  }
  port pxc-2.a {
    admin-state enable
  }
  port pxc-2.b {
    admin-state enable
  }
  port 1/1/m1/1 {
    admin-state enable          # enable internal loopback port
  }
}

```

```

port 1/1/m1/2 {
    admin-state enable                # enable internal loopback port
}
port-xc {
    pxc 1 {
        admin-state enable
        port-id 1/1/m1/1
    }
    pxc 2 {
        admin-state enable
        port-id 1/1/m1/2
    }
}
fwd-path-ext {
    fpe 1 {
        path {
            pxc 1                    # map FPE 1 to PXC 1
        }
        application {
            srv6 {
                type origination
            }
        }
    }
    fpe 2 {
        path {
            pxc 2                    # map FPE 2 to PXC 2
        }
        application {
            srv6 {
                type termination
            }
        }
    }
}
router "Base" {
    segment-routing {
        segment-routing-v6 {
            origination-fpe [1]
            source-address 2001:db8::2:2
            locator "PE-2_loc" {
                admin-state enable
                block-length 48
                function-length 20
                termination-fpe [2]
                prefix {
                    ip-prefix 2001:db8:aaaa:102::/64
                }
            }
            base-routing-instance {
                locator "PE-2_loc" {
                    function {
                        end 1 {
                            srh-mode usp
                        }
                    }
                    end-x-auto-allocate usp protection protected { }
                }
            }
        }
    }
}
isis 0 {
    admin-state enable
    advertise-passive-only true
}

```



```
IPv6 Tunnel Table (Router: Base)
=====
Destination                               Owner      Encap TunnelId  Pref
NextHop                                   Color      Metric
-----
2001:db8:aaaa:101::/64 [L]                srv6-isis SRV6  524292    0
  fe80::24:1ff:fe01:1-"int-PE-2-P-6"      50
2001:db8:aaaa:102:0:2000::/128 [L]         srv6-isis SRV6  524289    0
  fe80::24:1ff:fe01:1-"int-PE-2-P-6"      10
2001:db8:aaaa:102:0:3000::/128 [L]         srv6-isis SRV6  524290    0
  fe80::28:1ff:fe01:15-"int-PE-2-P-7"     10
2001:db8:aaaa:102:0:4000::/128 [L]         srv6-isis SRV6  524291    0
  fe80::18:1ff:fe01:b-"int-PE-2-PE-3"     10
2001:db8:aaaa:103::/64 [L]                srv6-isis SRV6  524293    0
  fe80::18:1ff:fe01:b-"int-PE-2-PE-3"     20
2001:db8:aaaa:104::/64 [L]                srv6-isis SRV6  524294    0
  fe80::24:1ff:fe01:1-"int-PE-2-P-6"      40
2001:db8:aaaa:105::/64 [L]                srv6-isis SRV6  524295    0
  fe80::24:1ff:fe01:1-"int-PE-2-P-6"      30
2001:db8:aaaa:106::/64 [L]                srv6-isis SRV6  524296    0
  fe80::24:1ff:fe01:1-"int-PE-2-P-6"      20
2001:db8:aaaa:107::/64 [L]                srv6-isis SRV6  524297    0
  fe80::28:1ff:fe01:15-"int-PE-2-P-7"     20
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====
```

The following shows the IPv6 FP-tunnel table on PE-2. For locator prefix 2001:db8:aaaa:101::/64, tunnel ID 524292 has primary next hop fe80::23:fff:fe00:0-"int-PE-2-P-6" and backup next hop fe80::27:fff:fe00:0-"int-PE-2-P-7".

```
[/]
A:admin@PE-2# show router fp-tunnel-table 1 ipv6

=====
IPv6 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                               Protocol   Tunnel-ID
Lbl/SID
NextHop                                   Intf/Tunnel
Lbl/SID (backup)
NextHop (backup)
-----
2001:db8:aaaa:101::/64                    SRV6      524292
-
  fe80::24:1ff:fe01:1-"int-PE-2-P-6"      1/1/c2/1:1000
-
  fe80::28:1ff:fe01:15-"int-PE-2-P-7" (B) 1/1/c3/1:1000
2001:db8:aaaa:103::/64                    SRV6      524293
-
  fe80::18:1ff:fe01:b-"int-PE-2-PE-3"     1/1/c1/1:1000
-
  fe80::24:1ff:fe01:1-"int-PE-2-P-6" (B) 1/1/c2/1:1000
2001:db8:aaaa:104::/64                    SRV6      524294
-
  fe80::24:1ff:fe01:1-"int-PE-2-P-6"      1/1/c2/1:1000
```

```

-
  fe80::28:1ff:fe01:15-"int-PE-2-P-7" (B)          1/1/c3/1:1000
2001:db8:aaaa:105::/64                          SRV6      524295
-
  fe80::24:1ff:fe01:1-"int-PE-2-P-6"            1/1/c2/1:1000
-
  fe80::28:1ff:fe01:15-"int-PE-2-P-7" (B)          1/1/c3/1:1000
2001:db8:aaaa:106::/64                          SRV6      524296
-
  fe80::24:1ff:fe01:1-"int-PE-2-P-6"            1/1/c2/1:1000
-
  fe80::18:1ff:fe01:b-"int-PE-2-PE-3" (B)         1/1/c1/1:1000
2001:db8:aaaa:107::/64                          SRV6      524297
-
  fe80::28:1ff:fe01:15-"int-PE-2-P-7"            1/1/c3/1:1000
-
  fe80::18:1ff:fe01:b-"int-PE-2-PE-3" (B)         1/1/c1/1:1000
2001:db8:aaaa:102:0:2000::/128                 SRV6      524289
-
  fe80::24:1ff:fe01:1-"int-PE-2-P-6"            1/1/c2/1:1000
2001:db8:aaaa:106:0:1000::
  fe80::18:1ff:fe01:b-"int-PE-2-PE-3" (B)         1/1/c1/1:1000
2001:db8:aaaa:102:0:3000::/128                 SRV6      524290
-
  fe80::28:1ff:fe01:15-"int-PE-2-P-7"            1/1/c3/1:1000
2001:db8:aaaa:107:0:1000::
  fe80::18:1ff:fe01:b-"int-PE-2-PE-3" (B)         1/1/c1/1:1000
2001:db8:aaaa:102:0:4000::/128                 SRV6      524291
-
  fe80::18:1ff:fe01:b-"int-PE-2-PE-3"            1/1/c1/1:1000
2001:db8:aaaa:103:0:1000::
  fe80::24:1ff:fe01:1-"int-PE-2-P-6" (B)         1/1/c2/1:1000
-----
Total Entries : 9
-----
=====

```

The IPv6 route table on PE-2 contains the following prefixes with shared block 2001:db8:aaaa::/48.

```

[/]
A:admin@PE-2# show router route-table ipv6 2001:db8:aaaa::/48 longer
=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age           Pref
Next Hop[Interface Name]   Metric
-----
2001:db8:aaaa:101::/64      Remote ISIS   00h13m34s  18
      2001:db8:aaaa:101::/64 (tunneled:SRV6-ISIS)
      50
2001:db8:aaaa:102::/64      Local  SRV6     00h22m44s   3
      fe80::201-"_tmnx_fpe_2.a"
      0
2001:db8:aaaa:102:0:1000::/128 Local  SRV6     00h15m41s   3
      Black Hole
      0
2001:db8:aaaa:102:0:2000::/128 Local  ISIS     00h13m35s  18
      2001:db8:aaaa:102:0:2000:: (tunneled:SRV6-ISIS)
      10
2001:db8:aaaa:102:0:3000::/128 Local  ISIS     00h13m35s  18
      2001:db8:aaaa:102:0:3000:: (tunneled:SRV6-ISIS)
      10
2001:db8:aaaa:102:0:4000::/128 Local  ISIS     00h13m35s  18
      2001:db8:aaaa:102:0:4000:: (tunneled:SRV6-ISIS)
      10
2001:db8:aaaa:103::/64      Remote ISIS   00h13m27s  18
      2001:db8:aaaa:103::/64 (tunneled:SRV6-ISIS)
      20
2001:db8:aaaa:104::/64      Remote ISIS   00h13m21s  18

```

```

2001:db8:aaaa:104::/64 (tunneled:SRV6-ISIS)          40
2001:db8:aaaa:105::/64                               Remote  ISIS    00h13m15s  18
2001:db8:aaaa:105::/64 (tunneled:SRV6-ISIS)          30
2001:db8:aaaa:106::/64                               Remote  ISIS    00h13m10s  18
2001:db8:aaaa:106::/64 (tunneled:SRV6-ISIS)          20
2001:db8:aaaa:107::/64                               Remote  ISIS    00h13m02s  18
2001:db8:aaaa:107::/64 (tunneled:SRV6-ISIS)          20
-----
No. of Routes: 11
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

The following command on PE-2 shows the corresponding FIB:

```

[/]
A:admin@PE-2# show router fib 1 ipv6 ip-prefix-prefix-length 2001:db8:aaaa::/48 longer

=====
FIB Display
=====
Prefix [Flags]                                     Protocol
NextHop
-----
2001:db8:aaaa:101::/64                             ISIS
  2001:db8:aaaa:101::/64 (Transport:SRV6:524292)
2001:db8:aaaa:102::/64                             SRV6
  fe80::201 (_tmnx_fpe_2.a)
2001:db8:aaaa:102:0:1000::/128                     SRV6
  Blackhole
2001:db8:aaaa:102:0:2000::/128                     ISIS
  2001:db8:aaaa:102:0:2000:: (Transport:SRV6:524289)
2001:db8:aaaa:102:0:3000::/128                     ISIS
  2001:db8:aaaa:102:0:3000:: (Transport:SRV6:524290)
2001:db8:aaaa:102:0:4000::/128                     ISIS
  2001:db8:aaaa:102:0:4000:: (Transport:SRV6:524291)
2001:db8:aaaa:103::/64                             ISIS
  2001:db8:aaaa:103::/64 (Transport:SRV6:524293)
2001:db8:aaaa:104::/64                             ISIS
  2001:db8:aaaa:104::/64 (Transport:SRV6:524294)
2001:db8:aaaa:105::/64                             ISIS
  2001:db8:aaaa:105::/64 (Transport:SRV6:524295)
2001:db8:aaaa:106::/64                             ISIS
  2001:db8:aaaa:106::/64 (Transport:SRV6:524296)
2001:db8:aaaa:107::/64                             ISIS
  2001:db8:aaaa:107::/64 (Transport:SRV6:524297)
-----
Total Entries : 11
=====

```

Conclusion

SRv6 offers both shortest path and source routing capabilities. SRv6 can be deployed as an IPv6 transport for implementing services across a service provider network.

Segment Routing over IPv6 for VPRN

This chapter provides information about segment routing over IPv6 for VPRN.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information and configuration in this chapter are based on SR OS Release 22.2.R1. Segment routing over IPv6 (SRv6) is supported on FP4-based equipment in SR OS Release 21.5.R2 and later.

Overview

SRv6 for VPRN allows the transport of VPRN-related IPv4 and IPv6 data across an SRv6-enabled network. To this end, VPRN-related data is sent to an ingress SRv6 router, where it is encapsulated and forwarded via an SRv6 tunnel. The SRv6 tunnel transports the encapsulated data across the SRv6-enabled network to an egress SRv6 router, where it is decapsulated and forwarded further as VPRN-related data. SRv6-tunneled data is encapsulated using an IPv6 header, where the destination address is a unique SRv6 segment identifier (SID), and is processed and forwarded in the IPv6 data plane.

An SRv6 SID is a preconfigured 128-bit routable IPv6 prefix address that is encoded in three parts: a locator, a function, and an argument. The locator is a summary IPv6 prefix for a set of SRv6 SIDs instantiated on an SRv6-capable router. It is used to route the data within the IPv6 transport network. Each participating SRv6-capable router needs its unique locator, based on a common block that all participating SRv6-capable routers share in the IPv6 address space. The function is an opaque identifier that indicates the local behavior at the endpoint of an SRv6 segment. The focus in this topic is on the SRv6 End.DT4 and the SRv6 End.DT6 functions for the VPRN, performing a prefix lookup in the VPRN service IPv4 route table (End.DT4) or in the VPRN service IPv6 route table (End.DT6). The argument is not used in SR OS 22.2.R1 and is set to all zeros.

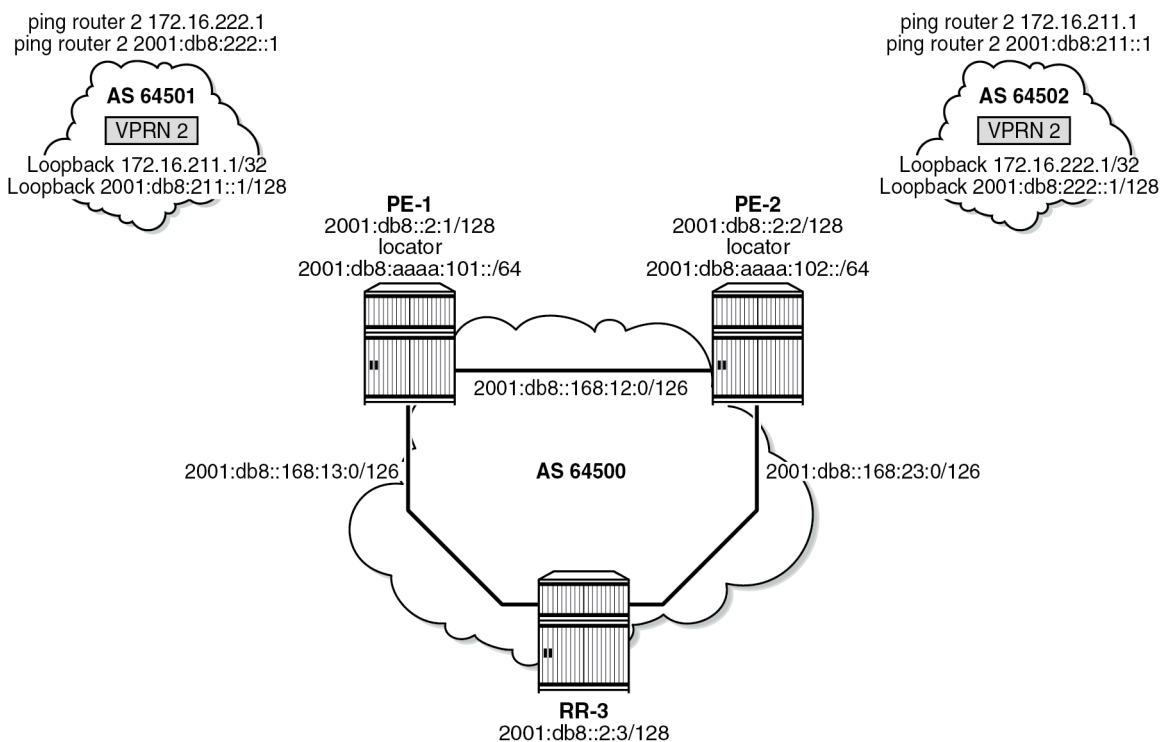
The local router installs its locator prefix in its IPv6 route table and forwarding information base (FIB), and advertises its locator prefix in IS-IS with the SRv6 locator sub-TLV. Each remote router populates its route table and FIB with the received locator prefixes, including the tunneled next hop to the originating router. Each remote router also populates its VPRN service route table with the received network prefixes, including the tunneled next hop to the VPRN of the originating router.

SRv6 data transport requires additional processing at both the ingress and egress data planes. This processing relies on forwarding path extension (FPE), as described in the [Segment Routing over IPv6](#) chapter.

Configuration

Figure 363: Example topology shows the example topology with three routers. The SRv6-enabled network that it represents comprises PE-1, PE-2, and a route reflector RR-3 in the control plane. The SRv6-enabled network has only IPv6 addresses and interfaces.

Figure 363: Example topology



37603

For the transport of IPv4 and IPv6 data from the VPRN on PE-1 to the VPRN on PE-2, PE-1 acts as the SRv6 ingress PE node, while PE-2 acts as the SRv6 egress PE node. For the transport of IPv4 and IPv6 data from the VPRN on PE-2 to the VPRN on PE-1, PE-2 acts as the SRv6 ingress PE node, while PE-1 acts as the SRv6 egress PE node. To explain SRv6 for VPRN, the topology does not need an SRv6 transit router, because SRv6 transit routers simply forward SRv6-encapsulated packets via IPv6 route table lookup without any other processing.

SRv6 and FPE are configured only on PE-1 and on PE-2. RR-3 acts as the BGP route reflector in the control plane. RR-3 does not participate in the SRv6 data transport that only exists between PE-1 and PE-2.

The **ping** and **traceroute** commands between IPv4 and IPv6 loopback addresses in the VPRNs simulate data transport.

The configuration for this example topology is symmetrical. All **configure** and **show** command output examples for PE-1 also apply to PE-2. The **configure** and **show** commands with deviating output examples for RR-3 are explicitly mentioned.

Configure the router

This configuration includes:

- ports and IPv6-only interfaces on PE-1, PE-2, and RR-3
- port cross-connect (PXC) on PE-1 and PE-2, using internal loopbacks on an FP4 MAC chip, as described in the Segment Routing over IPv6 chapter
- IS-IS
 - On PE-1, PE-2, and RR-3, include:
 - level 2 capability with wide metrics (for the 128-bit identifiers)
 - native IPv6 routing
 - On PE-1 and PE-2, as a best practice to advertise the router capability within the autonomous system (AS), also configure:
 - **traffic-engineering**
 - **traffic-engineering-options**
- BGP on PE-1, PE-2, and RR-3, with internal group “gr_v6_internal” that includes:
 - IPv4 and IPv6 families
 - **extended-nh-encoding** for IPv4
 - **advertise-ipv6-next-hops** for IPv4
 - BGP neighbor **system** IPv6 addresses
 - On PE-1 and PE-2 only: **next-hop-self**

The core network topology uses IPv6 for BGP peering (with 16 byte next hop addresses), so to advertise and receive IPv4 routes (which have 4 byte next hop addresses) with IPv6 next hop addresses, the commands **advertise-ipv6-next-hops** and **extended-nh-encoding** need to be configured at the BGP, group, or neighbor level. The **advertise-ipv6-next-hops** command instructs the system to advertise IPv4 routes with IPv6 next hop addresses. The **extended-nh-encoding** command configures BGP to advertise the capability to receive IPv4 routes with IPv6 next hop addresses.

The following example configuration applies for PE-1 and is similar for PE-2.

```
[/]  
A:admin@PE-1# configure {  
  router "Base" {  
    autonomous-system 64500  
    interface "int-PE-1-PE-2" {  
      description "interface between PE-1 and PE-2"  
      port 1/1/c1/1:1000  
      ipv6 {  
        address 2001:db8::168:12:1 {  
          prefix-length 126  
        }  
      }  
    }  
    interface "int-PE-1-RR-3" {  
      description "interface between PE-1 and RR-3"  
      port 1/1/c2/1:1000  
      ipv6 {  
        address 2001:db8::168:13:1 {  
          prefix-length 126  
        }  
      }  
    }  
  }  
}
```

```
}
interface "system" {
  description "system interface of PE-1"
  ipv6 {
    address 2001:db8::2:1 {
      prefix-length 128
    }
  }
}
bgp {
  min-route-advertisement 1
  router-id 2.2.2.1
  rapid-withdrawal true
  split-horizon true
  ebgp-default-reject-policy {
    import false
    export false
  }
  group "gr_v6_internal" {
    description "internal bgp group on PE-1"
    next-hop-self true
    type internal
    family {
      ipv4 true
      ipv6 true
    }
    extended-nh-encoding {
      ipv4 true
    }
    advertise-ipv6-next-hops {
      ipv4 true
    }
  }
  neighbor "2001:db8::2:3" { # RR-3 system address
    group "gr_v6_internal"
  }
}
isis 0 {
  admin-state enable
  advertise-router-capability as
  ipv6-routing native
  level-capability 2 # required for SRv6
  router-id 1.1.1.1
  traffic-engineering true
  area-address [49.0001]
  traffic-engineering-options {
    ipv6 true
    application-link-attributes {
    }
  }
  interface "int-PE-1-PE-2" {
    interface-type point-to-point
  }
  interface "int-PE-1-RR-3" {
    interface-type point-to-point
  }
  interface "system" {
    passive true
  }
  level 2 {
    wide-metrics-only true # required for SRv6
  }
}
exit all
```

The following example configuration applies for RR-3:

```
[/]
A:admin@RR-3# configure {
  router "Base" {
    autonomous-system 64500
    interface "int-RR-3-PE-1" {
      description "interface between RR-3 and PE-1"
      port 1/1/c1/1:1000
      ipv6 {
        address 2001:db8::168:13:2 {
          prefix-length 126
        }
      }
    }
    interface "int-RR-3-PE-2" {
      description "interface between RR-3 and PE-2"
      port 1/1/c2/1:1000
      ipv6 {
        address 2001:db8::168:23:2 {
          prefix-length 126
        }
      }
    }
    interface "system" {
      description "system interface of RR-3"
      ipv6 {
        address 2001:db8::2:3 {
          prefix-length 128
        }
      }
    }
    bgp {
      min-route-advertisement 1
      router-id 2.2.2.3
      rapid-withdrawal true
      split-horizon true
      ebgp-default-reject-policy {
        import false
        export false
      }
      group "gr_v6_internal" {
        description "internal bgp group on RR-3"
        type internal
        family {
          ipv4 true
          ipv6 true
        }
        cluster {
          cluster-id 3.3.3.3
        }
        extended-nh-encoding {
          ipv4 true
        }
        advertise-ipv6-next-hops {
          ipv4 true
        }
      }
      neighbor "2001:db8::2:1" { # PE-1 system address
        group "gr_v6_internal"
      }
      neighbor "2001:db8::2:2" { # PE-2 system address
        group "gr_v6_internal"
      }
    }
  }
}
```

```

}
isis 0 {
  admin-state enable
  ipv6-routing native
  level-capability 2    # required for SRv6
  router-id 1.1.1.3
  area-address [49.0001]
  interface "int-RR-3-PE-1" {
    interface-type point-to-point
  }
  interface "int-RR-3-PE-2" {
    interface-type point-to-point
  }
  interface "system" {
    passive true
  }
  level 2 {
    wide-metrics-only true    # required for SRv6
  }
}
exit all

```

Configure the VPRN services on PE-1 and on PE-2

This configuration includes:

- an IPv4 address and an IPv6 address for a loopback interface "lb_if_vprn"
- BGP, with external group "gr_v6_vprn" that includes the following capabilities:
 - IPv4 and IPv6 families
 - **extended-nh-encoding** for IPv4
 - **advertise-ipv6-next-hops** for IPv4
 - BGP neighbor **interface** IPv6 addresses, with BGP neighbors in a different external AS

The following example configuration applies for VPRN 2 on PE-1 and is similar for VPRN 2 on PE-2.

```

[/]
A:admin@PE-1# configure {
  service {
    vprn "VPRN_2" {
      admin-state enable
      description "VPRN 2 on PE-1"
      service-id 2
      customer "1"
      autonomous-system 64500
      bgp {
        ebgp-default-reject-policy {
          import false
          export false
        }
        group "gr_v6_vprn" {
          description "external bgp group for VPRN 2 on PE-1"
          family {
            ipv4 true
            ipv6 true
          }
          extended-nh-encoding {
            ipv4 true
          }
        }
      }
    }
  }
}

```

```

    }
    advertise-ipv6-next-hops {
        ipv4 true
    }
}
neighbor "2001:db8:101::1" {
    group "gr_v6_vprn"
    type external
    peer-as 64501
}
}
interface "lb_itf_vprn" {
    description "VPRN 2 interface on PE-1 for external subnet"
    loopback true
    ipv4 {
        primary {
            address 172.16.211.1
            prefix-length 32
        }
    }
    ipv6 {
        address 2001:db8:211::1 {
            prefix-length 128
        }
    }
}
}
exit all

```

At this point, verify that data transport is not possible between the local VPRN on PE-1 and the remote VPRN on PE-2.

```

[/]
A:admin@PE-1# ping 172.16.222.1 router-instance "VPRN_2"
PING 172.16.222.1 56 data bytes
No route to destination. Address: 172.16.222.1, Router Instance: "VPRN_2"
---snip---
---- 172.16.222.1 PING Statistics ----
5 packets transmitted, 0 packets received, 100% packet loss

[/]
A:admin@PE-1# ping 2001:db8:222::1 router-instance "VPRN_2"
PING 2001:db8:222::1 56 data bytes
No route to destination. Address: 2001:db8:222::1, Router Instance: "VPRN_2"
---snip---
---- 2001:db8:222::1 PING Statistics ----
5 packets transmitted, 0 packets received, 100% packet loss

```

The result of the verification complies with the route table for the local VPRN on PE-1 that only contains local routes for its own loopback addresses:

```

[/]
A:admin@PE-1# show router 2 route-table ipv4

=====
Route Table (Service: 2)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]
Type      Proto  Age      Pref
Active    Metric
-----
172.16.211.1/32
lb_itf_vprn
Local    Local  00h01m21s  0
         Y      0
-----

```

```

No. of Routes: 1
---snip---
=====

[/]
A:admin@PE-1# show router 2 route-table ipv6

=====
IPv6 Route Table (Service: 2)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]         Active Metric
-----
2001:db8:211::1/128              Local  Local   00h01m19s    0
  lb_itf_vprn                     Y      0
-----
No. of Routes: 1
---snip---
=====

```

Perform the same verification for data transport between the remote VPRN on PE-2 and the local VPRN on PE-1.

Configure SRv6 in the router Base context on PE-1 and on PE-2

Configure the locator in the **router Base segment-routing segment-routing-v6** context on PE-2 and similar on PE-1, with **ip-prefix** 2001:db8:aaaa:101::/64 for locator "PE-1_loc".

```

[/]
A:admin@PE-2# configure {
  router "Base" {
    segment-routing {
      segment-routing-v6 {
        locator "PE-2_loc" {
          admin-state enable
          block-length 48
          prefix {
            ip-prefix 2001:db8:aaaa:102::/64
          }
        }
      }
    }
  }
  exit all
}

```

Configure the FPEs on PE-2 and identical on PE-1.

```

[/]
A:admin@PE-2# configure {
  fwd-path-ext {
    fpe 1 {
      path {
        pxc 1
      }
      application {
        srv6 {
          type origination
        }
      }
    }
    fpe 2 {
      path {

```

```

        pxc 2
    }
    application {
        srv6 {
            type termination
        }
    }
}
exit all

```

Use FPE 1 as the SRv6 origination FPE in the **router Base segment-routing segment-routing-v6** context and FPE 2 as the SRv6 termination FPE in the **router Base segment-routing segment-routing-v6 locator** context on PE-2 and similar on PE-1, for locator “PE-1_loc”. For more information, see the [Segment Routing over IPv6](#) chapter.

```

[/]
A:admin@PE-2# configure {
    router "Base" {
        segment-routing {
            segment-routing-v6 {
                origination-fpe [1]
                locator "PE-2_loc" {
                    admin-state enable
                    termination-fpe [2]
                }
            }
        }
    }
}
exit all

```

Configure the SRv6 End function (equivalent to an IPv4 node SID) and SRv6 End.X functions (equivalent to IPv4 adjacency SIDs) in the **router Base segment-routing segment-routing-v6 base-routing-instance locator** context on PE-2 and similar on PE-1, for locator “PE-1_loc”.

```

[/]
A:admin@PE-2# configure {
    router "Base" {
        segment-routing {
            segment-routing-v6 {
                base-routing-instance {
                    locator "PE-2_loc" {
                        function {
                            end 1 {
                                srh-mode usp
                            }
                            end-x-automatically allocate psp protection unprotected { }
                        }
                    }
                }
            }
        }
    }
}
exit all

```

Advertise the locator in IS-IS while ensuring level 2 capability on PE-2 and similar on PE-1, for locator “PE-1_loc”.

```

[/]
A:admin@PE-2# configure {
    router "Base" {
        isis 0 {
            segment-routing-v6 {
                admin-state enable
                locator "PE-2_loc" {
                    level-capability 2
                }
            }
        }
    }
}

```

```
exit all
```

A summary of the locator and origination FPE configuration can be displayed with the **show router segment-routing-v6 summary** command.

Verify the SRv6 local SIDs on PE-2 and similar on PE-1. Three SRv6 local SIDs are created: one for the statically configured SRv6 End function (configured in the **router Base segment-routing segment-routing-v6 base-routing-instance locator** context) and two for the automatically allocated SRv6 End.X functions (one facing PE-1 and one facing RR-3). All three SRv6 local SIDs are concatenated with the locator. The statically configured SRv6 End function appears first with function number 1. In the example, the automatically allocated SRv6 End.X functions receive function numbers 2 and 4 respectively. RR-3 has no SRv6 configuration and does not have these SRv6 local SIDs and SRv6 functions.

```
[/]
A:admin@PE-2# show router segment-routing-v6 local-sid

=====
Segment Routing v6 Local SIDs
=====
SID                                     Type      Function
Locator
Context
-----
2001:db8:aaaa:102:0:1000::             End       1
PE-2_loc
Base
2001:db8:aaaa:102:0:2000::             End.X     2
PE-2_loc
None
2001:db8:aaaa:102:0:4000::             End.X     4
PE-2_loc
None
-----
SIDs : 3
-----
=====
```

Verify the SRv6 base routing instance details on PE-2 and similar on PE-1. The SRv6 functions for the configured locator are listed. The SRv6 End function is statically configured. There is an automatically allocated SRv6 End.X function for each IS-IS neighbor.

```
[/]
A:admin@PE-2# show router segment-routing-v6 base-routing-instance

=====
Segment Routing v6 Base Routing Instance
=====
Locator
Type      Function   SID                               Status/InstId
SRH-mode Protection Interface
-----
PE-2_loc
End      USP          1 2001:db8:aaaa:102:0:1000::      ok
-----
Auto-allocated End.X: PSP Unprotected,
-----
End.X    PSP          *2 2001:db8:aaaa:102:0:2000::      0
Unprotected int-PE-2-PE-1
ISIS Level: L2 Mac Address: 04:0a:01:01:00:01 Nbr Sys Id: 0010.0100.1001
End.X    PSP          *4 2001:db8:aaaa:102:0:4000::      0
```



```

PSP      Unprotected int-PE-2-RR-3
ISIS Level: L2 Mac Address: 04:12:01:01:00:0b Nbr Sys Id: 0010.0100.1003
-----
Legend: * - System allocated
    
```

Verify the IPv6 route table on PE-1. The IPv6 route table also has routes to the local and learned remote locators and to the local SRv6 function SIDs. The remotely configured locator prefix of PE-2 is reached via an SRv6 tunnel. The routes with protocol "SRV6" correspond with the locally configured locator prefix of PE-1 or the locally configured SRv6 End function.

```

[/]
A:admin@PE-1# show router route-table ipv6

=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]          Type   Proto   Age      Pref
Metric
-----
2001:db8::2:1/128                 Local  Local   00h10m45s  0
system                             0
2001:db8::2:2/128                 Remote  ISIS    00h00m27s  18
fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"  10
2001:db8::2:3/128                 Remote  ISIS    00h00m27s  18
fe80::612:1ff:fe01:1-"int-PE-1-RR-3"  10
2001:db8::168:12:0/126            Local  Local   00h10m44s  0
int-PE-1-PE-2                       0
2001:db8::168:13:0/126            Local  Local   00h10m44s  0
int-PE-1-RR-3                         0
2001:db8::168:23:0/126            Remote  ISIS    00h00m27s  18
fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"  20
2001:db8:aaaa:101::/64           Local  SRV6   00h01m47s  3
fe80::201-"tmnx_fpe_2.a"         0
2001:db8:aaaa:101:0:1000::/128    Local  SRV6   00h01m01s  3
Black Hole                       0
2001:db8:aaaa:101:0:2000::/128    Local  ISIS   00h00m28s  18
fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"  10
2001:db8:aaaa:101:0:4000::/128    Local  ISIS   00h00m28s  18
fe80::612:1ff:fe01:1-"int-PE-1-RR-3"  10
2001:db8:aaaa:102::/64           Remote ISIS   00h00m17s  18
2001:db8:aaaa:102::/64 (tunneled:SRV6-ISIS)  10
-----
No. of Routes: 11
---snip---
=====
    
```

Verify that the tunnel from PE-1 to the remote locator has SRv6 encapsulation and similar for the tunnel from PE-2. The tunnel table on RR-3 remains empty.

```

[/]
A:admin@PE-1# show router tunnel-table ipv6

=====
IPv6 Tunnel Table (Router: Base)
=====
Destination
Nexthop                          Owner   Encap TunnelId Pref
Color                               Metric
-----
2001:db8:aaaa:102::/64           srv6-isis SRV6 524289  0
fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"  10
-----
    
```

```
---snip---
```

Verify that the tunnel from PE-1 to the remote locator uses the “int-PE-1-PE-2” interface and similar for the tunnel from PE-2, where the tunnel to the remote locator uses the “int-PE-2-PE-1” interface. Interface “int-PE-1-PE-2” is configured on port 1/1/c1/1:1000. The FP tunnel table on RR-3 remains empty.

```
[/]
A:admin@PE-1# show router fp-tunnel-table 1 ipv6

=====
IPv6 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol      Tunnel-ID
 Lbl/SID                                     NextHop       Intf/Tunnel
 Lbl/SID (backup)                           NextHop (backup)
-----
2001:db8:aaaa:102::/64                      SRV6          524289
-
  fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"      1/1/c1/1:1000
-----
Total Entries : 1
=====
```

Verify the IS-IS data base on PE-1 with **show router isis database detail**. The output of this command provides information on each IS-IS-enabled router. Per uniquely identified IS-IS-enabled router, the SRv6 information indicates:

- the IS-IS-advertised router capabilities
- the advertised SRv6 locator TLV
- the advertised configured SRv6 End SID and automatically allocated SRv6 End.X SIDs

```
[/]
A:admin@PE-1# show router isis database detail

=====
Rtr Base ISIS Instance 0 Database (detail)
=====

Displaying Level 1 database
-----
Level (1) LSP Count : 0

Displaying Level 2 database
-----
LSP ID : PE-1.00-00                               Level : L2
Sequence : 0x6                                         Checksum : 0x97fb   Lifetime : 1189
Version : 1                                           Pkt Type : 20      Pkt Ver : 1
Attributes: L1L2                                     Max Area : 3       Alloc Len : 1492
SYS ID : 0010.0100.1001                             SysID Len : 6      Used Len : 398

TLVs :
Area Addresses:
```

```

Area Address : (3) 49.0001
Supp Protocols:
  Protocols   : IPv4
  Protocols   : IPv6
IS-Hostname   : PE-1
Router ID    :
  Router ID   : 1.1.1.1
TE Router ID v6 :
  Router ID   : 2001:db8::2:1
Router Cap    : 1.1.1.1, D:0, S:0
  TE Node Cap : B E M P
  SRv6 Cap    : 0x0000
  SR Alg      : metric based SPF
  Node MSD Cap: BMI : 0 SRH-MAX-SL : 10 SRH-MAX-END-POP : 9 SRH-MAX-H-ENCAPS : 1 SRH-MAX-END-
D : 9
I/F Addresses IPv6 :
  IPv6 Address : 2001:db8::2:1
  IPv6 Address : 2001:db8::168:12:1
  IPv6 Address : 2001:db8::168:13:1
TE IS Nbrs :
  Nbr : PE-2.00
  Default Metric : 10
  Sub TLV Len : 60
  IPv6 Addr : 2001:db8::168:12:1
  Nbr IPv6 : 2001:db8::168:12:2
  End.X-SID: 2001:db8:aaaa:101:0:2000:: flags: algo:0 weight:0 endpoint:End.X-PSP
TE IS Nbrs :
  Nbr : RR-3.00
  Default Metric : 10
  Sub TLV Len : 42
  IPv6 Addr : 2001:db8::168:13:1
  End.X-SID: 2001:db8:aaaa:101:0:4000:: flags: algo:0 weight:0 endpoint:End.X-PSP
IPv6 Reach:
  Metric: ( I ) 0
  Prefix : 2001:db8::2:1/128
  Metric: ( I ) 10
  Prefix : 2001:db8::168:12:0/126
  Metric: ( I ) 10
  Prefix : 2001:db8::168:13:0/126
  Metric: ( I ) 0
  Prefix : 2001:db8:aaaa:101::/64
SRv6 Locator :
  MT ID : 0
  Metric: ( ) 0 Algo:0
  Prefix : 2001:db8:aaaa:101::/64
  Sub TLV :
  End-SID : 2001:db8:aaaa:101:0:1000::, flags:0x0, endpoint:End-USP
-----
LSP ID : PE-2.00-00                               Level : L2
Sequence : 0x6                                     Checksum : 0x1740   Lifetime : 1185
Version : 1                                         Pkt Type : 20     Pkt Ver : 1
Attributes: L1L2                                   Max Area : 3      Alloc Len : 398
SYS ID : 0010.0100.1002                           SysID Len : 6     Used Len : 398

TLVs :
Area Addresses:
  Area Address : (3) 49.0001
Supp Protocols:
  Protocols   : IPv4
  Protocols   : IPv6
IS-Hostname   : PE-2
Router ID    :
  Router ID   : 1.1.1.2

```

```
TE Router ID v6 :
  Router ID : 2001:db8::2:2
  Router Cap : 1.1.1.2, D:0, S:0
  TE Node Cap : B E M P
  SRv6 Cap: 0x0000
  SR Alg: metric based SPF
  Node MSD Cap: BMI : 0 SRH-MAX-SL : 10 SRH-MAX-END-POP : 9 SRH-MAX-H-ENCAPS : 1 SRH-MAX-END-
D : 9
I/F Addresses IPv6 :
  IPv6 Address : 2001:db8::2:2
  IPv6 Address : 2001:db8::168:12:2
  IPv6 Address : 2001:db8::168:23:1
TE IS Nbrs :
  Nbr : PE-1.00
  Default Metric : 10
  Sub TLV Len : 60
  IPv6 Addr : 2001:db8::168:12:2
  Nbr IPv6 : 2001:db8::168:12:1
  End.X-SID: 2001:db8:aaaa:102:0:2000:: flags: algo:0 weight:0 endpoint:End.X-PSP
TE IS Nbrs :
  Nbr : RR-3.00
  Default Metric : 10
  Sub TLV Len : 42
  IPv6 Addr : 2001:db8::168:23:1
  End.X-SID: 2001:db8:aaaa:102:0:4000:: flags: algo:0 weight:0 endpoint:End.X-PSP
IPv6 Reach:
  Metric: ( I ) 0
  Prefix : 2001:db8::2:2/128
  Metric: ( I ) 10
  Prefix : 2001:db8::168:12:0/126
  Metric: ( I ) 10
  Prefix : 2001:db8::168:23:0/126
  Metric: ( I ) 0
  Prefix : 2001:db8:aaaa:102::/64
SRv6 Locator :
  MT ID : 0
  Metric: ( ) 0 Algo:0
  Prefix : 2001:db8:aaaa:102::/64
  Sub TLV :
  End-SID : 2001:db8:aaaa:102:0:1000::, flags:0x0, endpoint:End-USP
-----
LSP ID : RR-3.00-00
Sequence : 0x3
Version : 1
Attributes: L1L2
SYS ID : 0010.0100.1003
Checksum : 0xdba6
Pkt Type : 20
Max Area : 3
SysID Len : 6
Level : L2
Lifetime : 675
Pkt Ver : 1
Alloc Len : 193
Used Len : 193
TLVs :
  Area Addresses:
  Area Address : (3) 49.0001
  Supp Protocols:
  Protocols : IPv4
  Protocols : IPv6
  IS-Hostname : RR-3
  Router ID :
  Router ID : 1.1.1.3
  I/F Addresses IPv6 :
  IPv6 Address : 2001:db8::2:3
  IPv6 Address : 2001:db8::168:13:2
  IPv6 Address : 2001:db8::168:23:2
  TE IS Nbrs :
  Nbr : PE-1.00
  Default Metric : 10
```

```

Sub TLV Len      : 0
TE IS Nbrs      :
Nbr              : PE-2.00
Default Metric   : 10
Sub TLV Len      : 0
IPv6 Reach:
Metric: ( I ) 0
Prefix          : 2001:db8::2:3/128
Metric: ( I ) 10
Prefix          : 2001:db8::168:13:0/126
Metric: ( I ) 10
Prefix          : 2001:db8::168:23:0/126

Level (2) LSP Count : 3
-----
---snip---
=====

```

Verify the IS-IS routes on PE-1 and similar on PE-2.

```

[/]
A:admin@PE-1# show router isis routes

=====
Rtr Base ISIS Instance 0 Route Table
=====
Prefix[Flags]          Metric    Lvl/Typ   Ver.  SysID/Hostname
NextHop                MT        MT      AdminTag/SID[F]
-----
2001:db8::2:1/128      0         2/Int.    2     PE-1
::                    0         0         0
2001:db8::2:2/128      10        2/Int.    10    PE-2
 fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"  0         0
2001:db8::2:3/128      10        2/Int.    10    RR-3
 fe80::612:1ff:fe01:1-"int-PE-1-RR-3"  0         0
2001:db8::168:12:0/126 10        2/Int.    4     PE-1
::                    0         0
2001:db8::168:13:0/126 10        2/Int.    4     PE-1
::                    0         0
2001:db8::168:23:0/126 20        2/Int.    10    PE-2
 fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"  0         0
2001:db8:aaaa:101::/64 0         2/Int.    12    PE-1
::                    0         0
2001:db8:aaaa:102::/64 10        2/Int.    11    PE-2
 fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"  0         0

-----
No. of Routes: 8 (8 paths)
-----
---snip---
=====

```

This output corresponds with the information in the route table and in the FIB.

The BGP groups can be verified with the **show router bgp group** command. PE-1 and PE-2 know the internal and the external BGP groups. RR-3 only knows the internal BGP group.

The BGP next hops can be verified with the following commands:

- **show router bgp next-hop ipv4**
- **show router bgp next-hop ipv6**
- **show router bgp next-hop vpn-ipv4**

- **show router bgp next-hop vpn-ipv6**

Verify on PE-1 and similar on PE-2 that the locator prefixes are locally configured and advertised. In this example, PE-1 is aware of both locators. One locator is locally configured; the other is learned from the PE-2 advertisement.

```
[/]
A:admin@PE-1# show router isis segment-routing-v6 locator

=====
Rtr Base ISIS Instance 0 SRv6 Locator Table
=====
Prefix                               AdvRtr           MT      Lvl/Typ
AttributeFlags                       Tag              Flags   Algo
-----
2001:db8:aaaa:101::/64               PE-1             0       2/Int.
-                                     0                -       0
2001:db8:aaaa:102::/64               PE-2             0       2/Int.
-                                     0                -       0
-----
No. of Locators: 2
-----
---snip---
=====
```

Verify on PE-1 and similar on PE-2 that the SRv6 End SIDs are locally configured and advertised. In this example, PE-1 is aware of both SRv6 End SIDs. One End SID is locally configured; the other is learned from the PE-2 advertisement.

```
A:admin@PE-1# show router isis segment-routing-v6 end-sid

=====
Rtr Base ISIS Instance 0 SRv6 End SID Table
=====
Prefix                               AdvRtr           MT      Lvl/Typ
Sid                                  Behavior          Flags   Algo
-----
2001:db8:aaaa:101::/64               PE-1             0       2/Int.
  2001:db8:aaaa:101:0:1000::         End USP          -       0
2001:db8:aaaa:102::/64               PE-2             0       2/Int.
  2001:db8:aaaa:102:0:1000::         End USP          -       0
-----
No. of End SIDs: 2
=====
```

Configure SRv6 for the VPRNs on PE-1 and on PE-2

On PE-1, PE-2, and RR-3, extend the BGP advertisements to include the VPN-IPv4 and VPN-IPv6 families.

```
configure {
  router "Base" {
    bgp {
      rapid-update {
        vpn-ipv4 true
        vpn-ipv6 true
      }
      group "gr_v6_internal" {
        family {
```

```

        ipv4 true
        vpn-ipv4 true
        ipv6 true
        vpn-ipv6 true
    }
    extended-nh-encoding {
        vpn-ipv4 true
        ipv4 true
    }
    advertise-ipv6-next-hops {
        vpn-ipv6 true
        vpn-ipv4 true
        ipv4 true
    }
}
exit all

```

On PE-2, create an SRv6 instance for the VPRN service. Use the locator from the **router Base segment-routing segment-routing-v6** context and configure End.DT4 and End.DT6 functions for it.

Use the created SRv6 instance in the **service vprn bgp-ipvpn segment-routing-v6** context, with the configured locator as the default locator. Ensure a unique route distinguisher. Use the unique PE-2 system IPv6 address as the source address. Perform a similar configuration on PE-1, with the PE-1 locator as the default locator, the PE-1 system IPv6 address as the source address, and a different route distinguisher.

```

[/]
A:admin@PE-2# configure {
  service {
    vprn "VPRN_2" {
      segment-routing-v6 1 {
        locator "PE-2_loc" {
          function {
            end-dt4 {
            }
            end-dt6 {
            }
          }
        }
      }
    }
  }
  bgp-ipvpn {
    segment-routing-v6 1 {
      admin-state enable
      route-distinguisher "192.0.2.2:2"
      source-address 2001:db8::2:2
      vrf-target {
        community "target:64506:2"
      }
      srv6 {
        instance 1
        default-locator "PE-2_loc"
      }
    }
  }
}
exit all

```

This configuration results in BGP update exchanges from PE-2 to PE-1, via RR-3, and similar from PE-1 to PE-2, via RR-3. PE-2 sends BGP updates to RR-3 for the VPN-IPv4 and the VPN-IPv6 families respectively. Each BGP update advertises the VPN-IPv4 or VPN-IPv6 address family, the reachable network prefixes, the AS to which they belong, and an SRv6 Services TLV. The SRv6 Services TLV indicates that resolution to an SRv6 SID is available, making use of the endpoint behavior that is configured for the VPN-IPv4 or VPN-IPv6 address family on the locator. PE-1 programs the route prefixes

with an SRv6 tunnel next hop in its VPRN service route table and in its FIB. PE-1 and PE-2 advertise only the SRv6 SIDs for the SRv6 End.DT4 and SRv6 End.DT6 functions.

When debug logging for BGP updates is configured, this configuration results in the following BGP update logs for the VPN-IPv4 address family.

Consider the example for VPN-IPv4 prefix 172.16.222.1/32. Similar BGP update logs are generated also for VPN-IPv4 prefix 172.16.211.1/32, in the other direction.

The following BGP update log is for the VPN-IPv4 address family. It is sent by PE-2 and received (via RR-3) by PE-1:

```
[/]
A:admin@PE-1# show log log-id "log_2"

---snip---
3 2022/06/21 15:07:32.057 CEST MINOR: DEBUG #2001 Base Peer 1: 2001:db8::2:3
"Peer 1: 2001:db8::2:3: UPDATE
Peer 1: 2001:db8::2:3 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 128
  Flag: 0x90 Type: 14 Len: 45 Multiprotocol Reachable NLRI:
    Address Family VPN_IPV4
    NextHop len 24 NextHop 2001:db8::2:2
    172.16.222.1/32 RD 192.0.2.2:2 Label 524288 (Raw Label 0x800001)
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 0 AS Path:
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
  Flag: 0x80 Type: 9 Len: 4 Originator ID: 2.2.2.2
  Flag: 0x80 Type: 10 Len: 4 Cluster ID:
    3.3.3.3
  Flag: 0xc0 Type: 16 Len: 8 Extended Community:
    target:64506:2
  Flag: 0xc0 Type: 40 Len: 37 Prefix-SID-attr:
    SRv6 Services TLV (37 bytes):-
      Type: SRV6 L3 Service TLV (5)
      Length: 34 bytes, Reserved: 0x0
      SRv6 Service Information Sub-TLV (33 bytes)
      Type: 1 Len: 30 Rsvd1: 0x0
      SRv6 SID: 2001:db8:aaaa:102::
      SID Flags: 0x0 Endpoint Behavior: 0x13 Rsvd2: 0x0
      SRv6 SID Sub-Sub-TLV
      Type: 1 Len: 6
      BL:48 NL:16 FL:20 AL:0 TL:20 T0:64
"
---snip---
```

Similar BGP update logs are generated for the VPN-IPv6 address family.

Consider the example for VPN-IPv6 prefix 2001:db8:222::1/128. Similar BGP update logs are generated also for VPN-IPv6 prefix 2001:db8:211::1/128, in the other direction.

The following BGP update log is for the VPN-IPv6 address family. It is sent by PE-2 and received (via RR-3) by PE-1:

```
[/]
A:admin@PE-1# show log log-id "log_2"

---snip---
4 2022/06/21 15:07:32.057 CEST MINOR: DEBUG #2001 Base Peer 1: 2001:db8::2:3
"Peer 1: 2001:db8::2:3: UPDATE
Peer 1: 2001:db8::2:3 - Received BGP UPDATE:
  Withdrawn Length = 0
```



```

Total Path Attr Length = 140
Flag: 0x90 Type: 14 Len: 57 Multiprotocol Reachable NLRI:
  Address Family VPN_IPV6
  NextHop len 24 NextHop 2001:db8::2:2
  2001:db8:222::1/128 RD 192.0.2.2 Label 524287 (Raw Label 0x7ffff1)
Flag: 0x40 Type: 1 Len: 1 Origin: 0
Flag: 0x40 Type: 2 Len: 0 AS Path:
Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
Flag: 0x80 Type: 9 Len: 4 Originator ID: 2.2.2.2
Flag: 0x80 Type: 10 Len: 4 Cluster ID:
  3.3.3.3
Flag: 0xc0 Type: 16 Len: 8 Extended Community:
  target:64506:2
Flag: 0xc0 Type: 40 Len: 37 Prefix-SID-attr:
  SRv6 Services TLV (37 bytes):-
    Type: SRV6 L3 Service TLV (5)
    Length: 34 bytes, Reserved: 0x0
    SRv6 Service Information Sub-TLV (33 bytes)
      Type: 1 Len: 30 Rsvd1: 0x0
      SRv6 SID: 2001:db8:aaaa:102::
      SID Flags: 0x0 Endpoint Behavior: 0x12 Rsvd2: 0x0
      SRv6 SID Sub-Sub-TLV
        Type: 1 Len: 6
        BL:48 NL:16 FL:20 AL:0 TL:20 TO:64
"
---snip---

```

PE-1 receives from BGP peer RR-3 (peer router id 2.2.2.3) the information for network prefix 172.16.222.1/32 that PE-2 (originator id 2.2.2.2) advertised, as displayed in the RIB In Entries section in the following example. PE-1 programs route prefix 172.16.222.1/32 in its local VPRN service route table and FIB. The presence of the SRv6 Services TLV indicates that the next hop is the VPRN SRv6 End.DT4 SID which, in turn, is resolved to the remote locator for PE-2. PE-2 expects the data with VPN label 524288. PE-2 has concatenated the hexadecimal value 0x80000 of this VPN label to the remote SRv6 SID prefix 2001:db8:aaaa:102:: to form the remote SRv6 full SID 2001:db8:aaaa:102:8000:: that PE-1 must use. PE-1 uses the path that corresponds with this information (flags field). PE-1 sends SRv6 encapsulated IPv4 data from the VPRN in an SRv6 tunnel to the remote locator prefix of PE-2 on its “int-PE-1-PE-2” interface (as is shown in the output of the **show router tunnel-table ipv6** command). PE-1 uses the VPRN 2 route table for the prefix lookup (VPRN imported field).

PE-1 advertises to BGP peer RR-3 (peer router id 2.2.2.3) the information for network prefix 172.16.211.1/32, as displayed in the RIB Out Entries section in the following example. RR-3 forwards this information to its BGP neighbors, in this case PE-2. PE-2 acts in a similar way as PE-1.

The following output shows the corresponding VPN-IPv4 BGP routes on PE-1:

```

[/]
A:admin@PE-1# show router bgp routes vpn-ipv4 hunt
=====
BGP Router ID:2.2.2.1          AS:64500          Local AS:64500
=====
---snip---
=====
BGP VPN-IPv4 Routes
=====
-----
RIB In Entries
-----
Network       : 172.16.222.1/32
NextHop       : 2001:db8::2:2
Route Dist.   : 192.0.2.2                VPN Label       : 524288
Path Id       : None

```

```

From : 2001:db8::2:3
Res. Nexthop : n/a
Local Pref. : 100
Aggregator AS : None
Atomic Aggr. : Not Atomic
AIGP Metric : None
Connector : None
Community : target:64506:2
Cluster : 3.3.3.3
Originator Id : 2.2.2.2
Fwd Class : None
Flags : Used Valid Best IGP
Route Source : Internal
AS-Path : No As-Path
Route Tag : 0
Neighbor-AS : n/a
Orig Validation: N/A
Source Class : 0
Add Paths Send : Default
Last Modified : 00h00m33s
SRv6 TLV Type : SRv6 L3 Service TLV (5)
SRv6 SubTLV : SRv6 SID Information (1)
Sid : 2001:db8:aaaa:102::
Full Sid : 2001:db8:aaaa:102:8000::
Behavior : End.DT4 (19)
SRv6 SubSubTLV : SRv6 SID Structure (1)
Loc-Block-Len : 48
Func-Len : 20
Tpose-Len : 20
VPRN Imported : 2
Interface Name : int-PE-1-PE-2
Aggregator : None
MED : None
IGP Cost : 10
Peer Router Id : 2.2.2.3
Priority : None
Dest Class : 0
Loc-Node-Len : 16
Arg-Len : 0
Tpose-offset : 64

```

RIB Out Entries

```

Network : 172.16.211.1/32
Nexthop : 2001:db8::2:1
Route Dist. : 192.0.2.1:2
Path Id : None
To : 2001:db8::2:3
Res. Nexthop : n/a
Local Pref. : 100
Aggregator AS : None
Atomic Aggr. : Not Atomic
AIGP Metric : None
Connector : None
Community : target:64506:2
Cluster : No Cluster Members
Originator Id : None
Origin : IGP
AS-Path : No As-Path
Route Tag : 0
Neighbor-AS : n/a
Orig Validation: N/A
Source Class : 0
SRv6 TLV Type : SRv6 L3 Service TLV (5)
SRv6 SubTLV : SRv6 SID Information (1)
Sid : 2001:db8:aaaa:101::
Full Sid : 2001:db8:aaaa:101:8000::
Behavior : End.DT4 (19)
SRv6 SubSubTLV : SRv6 SID Structure (1)
Loc-Block-Len : 48
Func-Len : 20
Tpose-Len : 20
VPN Label : 524288
Interface Name : NotAvailable
Aggregator : None
MED : None
IGP Cost : n/a
Peer Router Id : 2.2.2.3
Loc-Node-Len : 16
Arg-Len : 0
Tpose-offset : 64

```

```
-----
Routes : 2
=====
```

For IPv6 data transport, VPRN End.DT6 behavior is needed. The IPv6 data transport uses a different VPN label 524287, resulting in a different full SRv6 SID ending with 7fff:f000:: PE-1 sends SRv6 encapsulated IPv6 data from the VPRN in an SRv6 tunnel to the remote locator prefix of PE-2 on its "int-PE-1-PE-2" interface (as is shown in the output of the **show router tunnel-table ipv6** command).

The following output shows the corresponding VPN-IPv6 BGP routes on PE-1:

```
[/]
A:admin@PE-1# show router bgp routes vpn-ipv6 hunt
=====
BGP Router ID:2.2.2.1          AS:64500          Local AS:64500
=====
---snip---
=====
BGP VPN-IPv6 Routes
=====
-----
RIB In Entries
-----
Network          : 2001:db8:222::1/128
Nexthop          : 2001:db8::2:2
Route Dist.     : 192.0.2.2:2          VPN Label       : 524287
Path Id         : None
From           : 2001:db8::2:3
Res. Nexthop    : n/a
Local Pref.     : 100
Aggregator AS  : None
Atomic Aggr.   : Not Atomic
AIGP Metric     : None
Connector      : None
Community       : target:64500:2
Cluster        : 3.3.3.3
Originator Id  : 2.2.2.2
Fwd Class       : None
Flags          : Used Valid Best IGP
Route Source    : Internal
AS-Path        : No As-Path
Route Tag       : 0
Neighbor-AS    : n/a
Orig Validation : N/A
Source Class    : 0
Add Paths Send : Default
Last Modified  : 00h00m33s
SRv6 TLV Type  : SRv6 L3 Service TLV (5)
SRv6 SubTLV    : SRv6 SID Information (1)
Sid            : 2001:db8:aaaa:102::
Full Sid       : 2001:db8:aaaa:102:7fff:f000::
Behavior       : End.DT6 (18)
SRv6 SubSubTLV : SRv6 SID Structure (1)
Loc-Block-Len  : 48
Func-Len       : 20
Tpose-Len      : 20
VPRN Imported  : 2
Interface Name  : int-PE-1-PE-2
Aggregator     : None
MED            : None
IGP Cost       : 10
Peer Router Id : 2.2.2.3
Priority        : None
Dest Class     : 0
Loc-Node-Len   : 16
Arg-Len        : 0
Tpose-offset   : 64
-----
RIB Out Entries
-----
Network          : 2001:db8:211::1/128
Nexthop          : 2001:db8::2:1
```

```

Route Dist.      : 192.0.2.1:2          VPN Label       : 524287
Path Id         : None
To              : 2001:db8::2:3
Res. Nexthop    : n/a
Local Pref.     : 100
Aggregator AS  : None                  Interface Name  : NotAvailable
Atomic Aggr.   : Not Atomic           Aggregator     : None
AIGP Metric    : None                 MED            : None
Connector      : None                 IGP Cost      : n/a
Community      : target:64506:2
Cluster        : No Cluster Members
Originator Id  : None                  Peer Router Id : 2.2.2.3
Origin         : IGP
AS-Path        : No As-Path
Route Tag      : 0
Neighbor-AS    : n/a
Orig Validation: N/A
Source Class   : 0                     Dest Class     : 0
SRv6 TLV Type  : SRv6 L3 Service TLV (5)
SRv6 SubTLV    : SRv6 SID Information (1)
Sid            : 2001:db8:aaaa:101::
Full Sid       : 2001:db8:aaaa:101:7fff:f000::
Behavior       : End.DT6 (18)
SRv6 SubSubTLV: SRv6 SID Structure (1)
Loc-Block-Len  : 48                    Loc-Node-Len   : 16
Func-Len       : 20                    Arg-Len        : 0
Tpose-Len      : 20                    Tpose-offset   : 64
    
```

```
-----
Routes : 2
=====
```

Verify that there are additional local SRv6 SIDs for PE-1 and PE-2. These local SRv6 SIDs correspond with the additional SRv6 behavior that is configured on the locator for the data transport between the local and the remote VPRN. Because RR-3 does not have SRv6 configuration, RR-3 does not have local SRv6 SIDs.

```

[/]
A:admin@PE-2# show router segment-routing-v6 local-sid

=====
Segment Routing v6 Local SIDs
=====
SID                                     Type          Function
Locator
Context
-----snip-----
2001:db8:aaaa:102:7fff:f000::          End.DT6      524287
  PE-2_loc
  SvcId: 2 Name: VPRN_2
2001:db8:aaaa:102:8000::              End.DT4      524288
  PE-2_loc
  SvcId: 2 Name: VPRN_2
-----
SIDs : 5
=====
    
```

Verify that there is SRv6 information for the VPRN service with service id 2.

```
[/]
```

```
A:admin@PE-2# show service id 2 segment-routing-v6 detail
```

```
=====
Segment Routing v6 Instance 1 Service 2
=====
Locator
Type          Function  SID                               Status
-----
PE-2_loc
  End.DT4      *524288 2001:db8:aaaa:102:8000::         ok
  End.DT6      *524287 2001:db8:aaaa:102:7fff:f000::     ok
=====
Legend: * - System allocated
```

At this point, verify that data transport is possible between the local VPRN on PE-1 and the remote VPRN on PE-2.

```
[/]
A:admin@PE-1# ping 172.16.222.1 router-instance "VPRN_2"
PING 172.16.222.1 56 data bytes
64 bytes from 172.16.222.1: icmp_seq=1 ttl=64 time=1.53ms.
---snip---
---- 172.16.222.1 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 1.48ms, avg = 1.53ms, max = 1.59ms, stddev = 0.035ms

[/]
A:admin@PE-1# ping 2001:db8:222::1 router-instance "VPRN_2"
PING 2001:db8:222::1 56 data bytes
64 bytes from 2001:db8:222::1: icmp_seq=1 hlim=64 time=1.29ms.
---snip---
---- 2001:db8:222::1 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 1.28ms, avg = 1.43ms, max = 1.78ms, stddev = 0.181ms

[/]
A:admin@PE-1# traceroute 172.16.222.1 router-instance "VPRN_2"
traceroute to 172.16.222.1, 30 hops max, 40 byte packets
 1 172.16.222.1 (172.16.222.1) 1.61 ms 1.49 ms 1.73 ms

[/]
A:admin@PE-1# traceroute 2001:db8:222::1 router-instance "VPRN_2"
traceroute to 2001:db8:222::1, 30 hops max, 60 byte packets
 1 2001:db8:222::1 (2001:db8:222::1) 1.36 ms 1.47 ms 1.57 ms
```

The result of the verification complies with the route table for the local VPRN on PE-1, which now also contains routes for the loopback addresses in the remote VPRN on PE-2. The same is true for data transport between the remote VPRN on PE-2 and the local VPRN on PE-1.

```
A:admin@PE-1# show router 2 route-table ipv4
```

```
=====
Route Table (Service: 2)
=====
Dest Prefix[Flags]          Type  Proto  Age      Pref
Next Hop[Interface Name]   Active Metric
-----
172.16.211.1/32             Local  Local  00h08m40s  0
  lb_itf_vprn                Y      0
172.16.222.1/32           Remote BGP VPN 00h01m00s 170
  2001:db8:aaaa:102:8000:: (tunneled:SRV6)  Y      10
-----
```

```
No. of Routes: 2
---snip---
=====

[/]
A:admin@PE-1# show router 2 route-table ipv6 all

=====
IPv6 Route Table (Service: 2)
=====
Dest Prefix[Flags]                Type    Proto  Age      Pref
  Next Hop[Interface Name]        Active  Active Metric
-----
2001:db8:211::1/128              Local   Local  00h08m38s  0
  lb_itf_vprn                    Y
2001:db8:222::1/128              Remote  BGP VPN 00h01m00s 170
  2001:db8:aaaa:102:7fff:f000:: (tunneled:SRV6) Y      10
-----
No. of Routes: 2
---snip---
=====
```

Conclusion

SRv6 shortest path routing can be used as an IPv6 transport for implementing VPRN services across an IPv6 service provider network.

Segment Routing with IS-IS Control Plane

This chapter provides information about Segment Routing (SR) with Intermediate System to Intermediate System (IS-IS) control plane.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

Segment routing is supported in SR OS Release 13.0, and later. This chapter was initially written for SR OS Release 13.0.R3, but the MD-CLI in the current edition corresponds to SR OS Release 21.2.R1.

Overview

Segment Routing (SR) is a technology for IP/Multi-Protocol Label Switching (MPLS) networks that enables source routing. With source routing, operators can specify a forwarding path, from ingress to egress, that is independent of the shortest path determined by the Interior Gateway Protocol (IGP).

The main benefit of segment routing compared to other source routing protocols (such as ReSource reservation Protocol with Traffic Engineering (RSVP-TE)) is that, from a control plane perspective, no signaling protocol is required. Segment routing provides a path or tunnel, encoded as a sequential list of sub-paths or segments that are advertised within the segment routing domain, using extensions to well-known link state routing protocols, such as IS-IS or Open Shortest Path First (OSPF).

Implementation

A segment routing tunnel can contain a single segment that represents the destination node, or it can contain a list of segments that the tunnel must traverse. The tunnel can be established over an IPv4/IPv6 MPLS or IPv6 data plane, encoded as a stack of MPLS labels or as a number of IPv6 addresses contained in an IPv6 extension header.

Network elements are modeled as segments. For each segment, IGP advertises an identifier referred to as a segment ID (SID).

The two segment types are:

- **Prefix segment** — Globally unique and allocated from a Segment Routing Global Block (SRGB), typically multi-hop and signaled by the IGP. It is the Equal Cost Multi-Path ECMP-aware shortest path IGP route to a related prefix. A typical example of a prefix segment is a node SID. Within the SR OS implementation, the node SID is either the system address or another interface address in the Global Routing Table (GRT) of type loopback. Node SIDs are advertised in IS-IS using a prefix SID sub-TLV (Type Length Value).

- **Adjacency segment** — Locally unique and allocated from the (local) dynamic label space, so that other routers in the SR domain can use the same label space. Adjacency segments are signaled by the IGP. Within the SR OS implementation, adjacency SIDs are automatically assigned and advertised when the SR context within the IGP instance is enabled. Adjacency SIDs are advertised in IS-IS using an adjacency SID sub-TLV.

To make prefix segments globally unique within the segment routing domain, an indexing mechanism is required, because production networks consist of multiple vendors and multiple products. As a result, it is often difficult to agree on a common SRGB for the prefix SIDs.

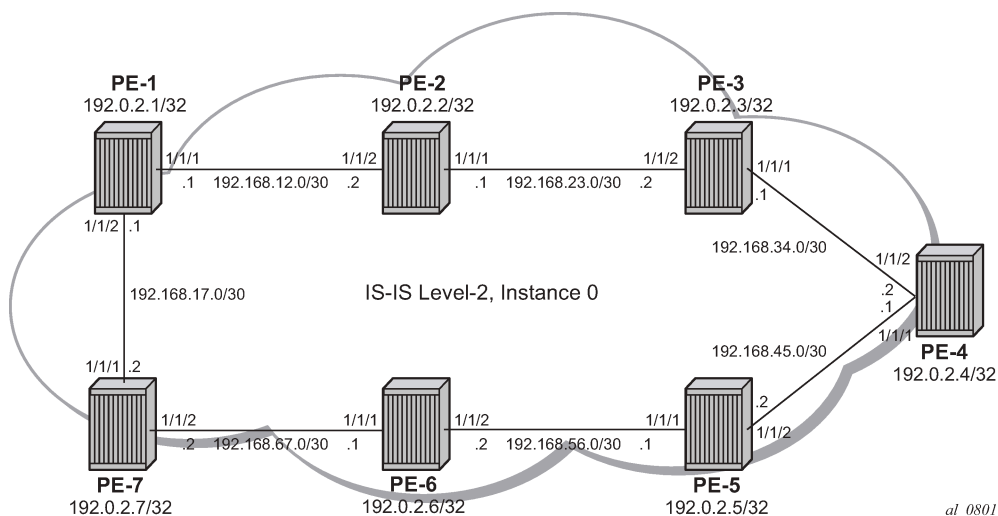
All routers within the SR domain are expected to configure and advertise the same Prefix SID index range for an IGP instance. The label value used by each router to represent a prefix can be local to that router by the use of an offset label, referred to as a start label:

$$\text{Local label (for a prefix)} = (\text{local}) \text{ start label} + \{\text{Prefix SID index}\}$$

Within the SR OS implementation, prefix Loop-Free Alternate (LFA) is supported for segment routing to improve the Fast ReRoute (FRR) coverage. Remote LFA (RLFA) is also supported. With RLFA, segment routing shortest path tunnels are used as a virtual LFA or repair tunnel toward the PQ node.

The following example uses IS-IS as an IGP protocol, with an MPLS data plane and services enabled using LFA and RLFA. [Figure 364: Example topology](#) shows the example topology with seven PEs.

Figure 364: Example topology



Configuration

1. Configure router interfaces and IS-IS according to [Figure 364: Example topology](#).
 - The system and IP interface addresses are configured according to [Figure 364: Example topology](#).
 - IS-IS level 2 is selected as the IGP to distribute routing information between all PEs. All IS-IS interfaces are of type point-to-point to avoid running the Designated Router/Backup Designated Router (DR/BDR) election process.
2. Configure segment routing.

Before enabling segment routing on a router, define a dedicated SRGB. This SRGB is required on each individual router part of the SR domain and is used to allocate the Prefix SIDs.

By default, an SRGB is not instantiated and, when configured by the operator, it is taken from the system dynamic label range. By default, the following label ranges are available:

```
[/]
A:admin@PE-1# show router mpls-labels label-range
```

```
=====
Label Ranges
=====
```

Label Type	Start Label	End Label	Aging	Available	Total
Static	32	18431	-	18400	18400
Dynamic	18432	524287	0	505856	505856
Seg-Route	0	0	-	0	0

```
=====
```

For simplicity, the same SRGB is used in this example for all SR domain routers. Within the command, a start value and end value define the size of the SRGB. The following command configures an SRGB of 100 MPLS labels, from label 20000 to label 20099:

```
# on PE-1, PE-2, PE-3, PE-4, PE-5, PE-6, PE-7:
configure {
  router "Base" {
    mpls-labels {
      sr-labels {
        start 20000
        end 20099
      }
    }
  }
}
```

```
[/]
A:admin@PE-1# show router mpls-labels label-range
```

```
=====
Label Ranges
=====
```

Label Type	Start Label	End Label	Aging	Available	Total
Static	32	18431	-	18400	18400
Dynamic	18432	524287	0	505756	505856
Seg-Route	20000	20099	-	0	100

```
=====
```

This command is repeated for all other nodes. The allocated MPLS labels are only for the prefix SIDs. The adjacency SIDs, which are only locally unique, are taken from the dynamic range; in this example, between 18432 and 524287.

a. Enable router capability in the IGP instance.

It is mandatory to enable the router-capability parameter inside the IS-IS instance, to advertise SR support among the IS-IS adjacencies. By configuring this command within the IGP instance, the SR capability sub-TLV is propagated and is used to indicate the index range and the start label. The SR algorithm sub-TLV is also used to advertise the algorithm used for path calculations. Only Shortest Path First (SPF) (value 0) is defined. This is configured as follows:

```
# on PE-1, PE-2, PE-3, PE-4, PE-5, PE-6, PE-7:
```

```
configure {
  router "Base" {
    isis 0 {
      advertise-router-capability area
    }
  }
}
```

The flooding parameter is a mandatory parameter in this command. The keyword `area` or `as` indicates that the router capabilities label switched path (LSP) should be advertised throughout the same level or throughout the whole Autonomous System (AS). In the preceding example, all routers belong to the same level, so the `area` argument is sufficient. When the SR context within the IGP instance is enabled, both IS-IS sub-TLVs are flooded.

b. Define the Prefix SID index range.

The SR OS implementation for SR provides two mutually exclusive modes of operation to define the prefix SID index range: global mode and per-instance mode. Per-instance mode is useful in a seamless MPLS environment when multiple IGP instances are used. The main difference between the modes is the way that the start label and index range are calculated.

A comparison of the modes is shown in following table:

Table 23: Mode comparison

Global	Per instance
Applicable for all IGP instances on that node	Applicable for one dedicated IGP instance
Start label is first label of SRGB	Start label is configurable (but part of SRGB range); use of non-overlapping sub-ranges of SRGB
Prefix SID index range is "size" of SRGB	Prefix SID index-range is configurable
If SRGB needs to change, disable SR and delete prefix-SID-ranges in all IGP instances	If prefix SID index and/or label range needs to change, disable SR in that specific IGP instance
SW checks whether any allocated SID index/label goes out of range.	
SW checks also for overlaps of the resulting net label value range across IGP instances.	

For simplicity, global mode is used for this example, as follows:

```
# on PE-1, PE-2, PE-3, PE-4, PE-5, PE-6, PE-7:
configure {
  router "Base" {
    isis 0 {
      segment-routing {
        prefix-sid-range {
          global
        }
      }
    }
  }
}
```

c. Assign a prefix SID index or label to the prefix representing a node.

To be able to set up SR shortest path tunnels to all routers of the SR domain, each router needs to be uniquely defined within the SR domain. Therefore, the system address or other loopback

interface in the GRT will be assigned an **ipv4-node-sid index** or **label** value that is unique within the SR domain. The prefix SID index is assigned as follows:

```
# on PE-1:
configure {
  router "Base" {
    isis 0 {
      interface "system" {
        ipv4-node-sid {
          index 1
        }
      }
    }
  }
}
```

```
# on PE-2:
configure {
  router "Base" {
    isis 0 {
      interface "system" {
        ipv4-node-sid {
          index 2
        }
      }
    }
  }
}
```

```
# on PE-3:
configure {
  router "Base" {
    isis 0 {
      interface "system" {
        ipv4-node-sid {
          index 3
        }
      }
    }
  }
}
```

```
# on PE-4:
configure {
  router "Base" {
    isis 0 {
      interface "system" {
        ipv4-node-sid {
          index 4
        }
      }
    }
  }
}
```

```
# on PE-5:
configure {
  router "Base" {
    isis 0 {
      interface "system" {
        ipv4-node-sid {
          index 5
        }
      }
    }
  }
}
```

```
# on PE-6:
configure {
  router "Base" {
    isis 0 {
      interface "system" {
        ipv4-node-sid {
```

```

        }
    }
}
index 6
}
}

# on PE-7:
configure {
    router "Base" {
        isis 0 {
            interface "system" {
                ipv4-node-sid {
                    index 7
                }
            }
        }
    }
}

```

Because the SRGB is the same on all nodes, each node in the network can be reached using the same MPLS label. For example, the node SID for PE-5 on all nodes has a start label (first label of the SRGB (= 20000) + ipv4-node-sid index on node PE-5 (= 5)) of 20005.

When there is one consistent SRGB for the SR domain, the SR OS allows the use of absolute MPLS label values instead of index values. For example, on PE-1, an operator can use an explicit MPLS label value, as follows:

```

# on PE-1:
configure {
    router "Base" {
        isis 0 {
            interface "system" {
                ipv4-node-sid {
                    label 20001
                }
            }
        }
    }
}

```

Internally, this explicit value is translated into an index value (index-value 1) before advertising it toward its neighbors, taking into account the prefix SID index-range mode (global or per-instance) and the SRGB.

- d. Enable SR context within the IGP instance, as follows:

```

# on PE-1, PE-2, PE-3, PE-4, PE-5, PE-6, PE-7:
configure {
    router "Base" {
        isis 0 {
            segment-routing {
                admin-state enable
            }
        }
    }
}

```

After enabling the SR context within an IGP instance, the SR capability sub-TLV, and the SR algorithm sub-TLV between all routers within the SR domain, are flooded. The following show command displays the SR related router capability information on PE-1:

```

[/]
A:admin@PE-1# show router isis capabilities level 2

=====
Rtr Base ISIS Instance 0 Capabilities
=====
Displaying Level 2 capabilities

```

```

-----
LSP ID      : PE-1.00-00
  Router Cap : 192.0.2.1, D:0, S:0
    TE Node Cap : B E M P
    SR Cap: IPv4 MPLS-IPv6
      SRGB Base:20000, Range:100
    SR Alg: metric based SPF
    Node MSD Cap: BMI : 12 ERLD : 15
LSP ID      : PE-2.00-00
  Router Cap : 192.0.2.2, D:0, S:0
    TE Node Cap : B E M P
    SR Cap: IPv4 MPLS-IPv6
      SRGB Base:20000, Range:100
    SR Alg: metric based SPF
    Node MSD Cap: BMI : 12 ERLD : 15
LSP ID      : PE-3.00-00
  Router Cap : 192.0.2.3, D:0, S:0
    TE Node Cap : B E M P
    SR Cap: IPv4 MPLS-IPv6
      SRGB Base:20000, Range:100
    SR Alg: metric based SPF
    Node MSD Cap: BMI : 12 ERLD : 15
LSP ID      : PE-4.00-00
  Router Cap : 192.0.2.4, D:0, S:0
    TE Node Cap : B E M P
    SR Cap: IPv4 MPLS-IPv6
      SRGB Base:20000, Range:100
    SR Alg: metric based SPF
    Node MSD Cap: BMI : 12 ERLD : 15
LSP ID      : PE-5.00-00
  Router Cap : 192.0.2.5, D:0, S:0
    TE Node Cap : B E M P
    SR Cap: IPv4 MPLS-IPv6
      SRGB Base:20000, Range:100
    SR Alg: metric based SPF
    Node MSD Cap: BMI : 12 ERLD : 15
LSP ID      : PE-6.00-00
  Router Cap : 192.0.2.6, D:0, S:0
    TE Node Cap : B E M P
    SR Cap: IPv4 MPLS-IPv6
      SRGB Base:20000, Range:100
    SR Alg: metric based SPF
    Node MSD Cap: BMI : 12 ERLD : 15
LSP ID      : PE-7.00-00
  Router Cap : 192.0.2.7, D:0, S:0
    TE Node Cap : B E M P
    SR Cap: IPv4 MPLS-IPv6
      SRGB Base:20000, Range:100
    SR Alg: metric based SPF
    Node MSD Cap: BMI : 12 ERLD : 15
Level (2) Capability Count : 7
=====

```

A similar output occurs for each router in the SR domain.

After enabling the SR context within the IGP instance, the assigned index for each locally configured prefix SID is advertised. After the advertisement of prefix SIDs, MPLS data plane Ingress Label Mapping (ILM) is programmed with a pop operation. In this context, a show command can be used to display the prefix SIDs, in order, within the SR domain. As an example, on PE-1, this becomes:

```

[/]
A:admin@PE-1# show router isis prefix-sids

```

```

=====
Rtr Base ISIS Instance 0 Prefix/SID Table
=====
Prefix                               SID      Lvl/Typ  SRMS   AdvRtr
MT                                     Flags
-----
192.0.2.1/32                         1        2/Int.   N      PE-1
                                           0      NnP
192.0.2.2/32                         2        2/Int.   N      PE-2
                                           0      NnP
192.0.2.3/32                         3        2/Int.   N      PE-3
                                           0      NnP
192.0.2.4/32                         4        2/Int.   N      PE-4
                                           0      NnP
192.0.2.5/32                         5        2/Int.   N      PE-5
                                           0      NnP
192.0.2.6/32                         6        2/Int.   N      PE-6
                                           0      NnP
192.0.2.7/32                         7        2/Int.   N      PE-7
                                           0      NnP
-----
No. of Prefix/SIDs: 7 (7 unique)
-----
SRMS : Y/N = prefix SID advertised by SR Mapping Server (Y) or not (N)
      S    = SRMS prefix SID is selected to be programmed
Flags: R    = Re-advertisement
      N    = Node-SID      nP = no penultimate hop POP
      E    = Explicit-Null
      V    = Prefix-SID carries a value
      L    = value/index has local significance
=====

```

By default, the SR OS implementation sets the node SID (or **N**-flag) and no Penultimate hop PoP (or **nP**-flag) inside the prefix SID TLV. Another useful flag that can be set is the re-advertisement (or **R**-flag). The R-flag is set when a prefix SID is propagated between levels or areas, or redistribution is in place (from another protocol).

Prefix SID information can also be viewed within the IGP database attached to (extended) IP prefix reachability TLVs. For example, on PE-1, as follows:

```

[/]
A:admin@PE-1# show router isis database PE-1.00-00 detail level 2

=====
Rtr Base ISIS Instance 0 Database (detail)
=====

Displaying Level 2 database
-----
LSP ID       : PE-1.00-00                Level      : L2
Sequence    : 0x6                       Checksum   : 0xc640  Lifetime  : 971
Version     : 1                          Pkt Type  : 20     Pkt Ver   : 1
Attributes: L1L2                         Max Area  : 3      Alloc Len : 1492
SYS ID      : 1920.0000.2001             SysID Len : 6      Used Len  : 254

TLVs :
  Supp Protocols:
    Protocols   : IPv4
  IS-Hostname   : PE-1
  Router ID     :
    Router ID   : 192.0.2.1
  Router Cap   : 192.0.2.1, D:0, S:0
  TE Node Cap  : B E M P

```

```

SR Cap: IPv4 MPLS-IPv6
  SRGB Base:20000, Range:100
SR Alg: metric based SPF
Node MSD Cap: BMI : 12 ERLD : 15
---snip---
Internal Reach:
---snip---
Default Metric: (I) 0
Delay Metric : (I) 0
Expense Metric: (I) 0
Error Metric : (I) 0
IP Address   : 192.0.2.1
IP Mask      : 255.255.255.255
I/F Addresses :
I/F Address  : 192.0.2.1
---snip---
TE IP Reach  :
---snip---
Default Metric : 0
Control Info:  S, prefLen 32
Prefix       : 192.0.2.1
Sub TLV      :
Prefix-SID Index:1, Algo:0, Flags:NnP

Level (2) LSP Count : 1
-----
---snip---
Prefix-SID Flags : R = Re-advertisement Flag
                  N = Node-SID Flag
                  nP = no penultimate hop POP
                  E = Explicit-Null Flag
                  V = Prefix-SID carries a value
                  L = value/index has local significance
---snip---
=====

```

After enabling the SR context within the IGP instance, adjacency SIDs are also automatically assigned and advertised for each formed adjacency over an IP interface. From a data plane perspective, one local adjacency SID consumes one ILM entry, programming a pop operation.

Similar to prefix SIDs, adjacency SID information can be viewed within the IGP database attached to IS neighbor TLVs, as follows:

```

[/]A:admin@PE-1# show router isis database PE-1.00-00 detail level 2

=====
Rtr Base ISIS Instance 0 Database (detail)
=====

Displaying Level 2 database
-----
LSP ID   : PE-1.00-00          Level    : L2
Sequence : 0x6                Checksum  : 0xc640  Lifetime : 971
Version  : 1                  Pkt Type  : 20    Pkt Ver  : 1
Attributes: L1L2             Max Area  : 3      Alloc Len : 1492
SYS ID   : 1920.0000.2001     SysID Len : 6      Used Len  : 254

TLVs :
  Supp Protocols:
    Protocols    : IPv4
  IS-Hostname   : PE-1
  Router ID     :
  Router ID     : 192.0.2.1

```

```

---snip---
IS Neighbors :
  Virtual Flag : 0
  Default Metric: (I) 10
  Delay Metric : (I) 0
  Expense Metric: (I) 0
  Error Metric : (I) 0
  Neighbor      : PE-2.00
IS Neighbors :
  Virtual Flag : 0
  Default Metric: (I) 10
  Delay Metric : (I) 0
  Expense Metric: (I) 0
  Error Metric : (I) 0
  Neighbor      : PE-7.00
Internal Reach:
  Default Metric: (I) 10
  Delay Metric : (I) 0
  Expense Metric: (I) 0
  Error Metric : (I) 0
  IP Address   : 192.168.12.0
  IP Mask      : 255.255.255.252
  Default Metric: (I) 10
  Delay Metric : (I) 0
  Expense Metric: (I) 0
  Error Metric : (I) 0
  IP Address   : 192.168.17.0
  IP Mask      : 255.255.255.252
---snip---
I/F Addresses :
---snip---
  I/F Address   : 192.168.12.1
  I/F Address   : 192.168.17.1
TE IS Nbrs :
  Nbr      : PE-2.00
  Default Metric : 10
  Sub TLV Len  : 19
  IF Addr   : 192.168.12.1
  Nbr IP    : 192.168.12.2
  Adj-SID: Flags:v4VL Weight:0 Label:524287
TE IS Nbrs : Nbr : PE-7.00
  Default Metric : 10
  Sub TLV Len  : 19
  IF Addr   : 192.168.17.1
  Nbr IP    : 192.168.17.2
  Adj-SID: Flags:v4VL Weight:0 Label:524286
TE IP Reach :
  Default Metric : 10
  Control Info:  , prefLen 30
  Prefix      : 192.168.12.0
  Default Metric : 10
  Control Info:  , prefLen 30
  Prefix      : 192.168.17.0
---snip---

Level (2) LSP Count : 1
-----
--snip--
Adj-SID Flags : v4/v6 = IPv4 or IPv6 Address-Family
               B = Backup Flag
               V = Adj-SID carries a value
               L = value/index has local significance
               S = Set of Adjacencies
               P = Persistently allocated

```


---snip---

By default, the SR OS implementation sets the value (**V**-flag), meaning that the adjacency SID carries a value (as opposed to an index). Also, the local **L**-flag is set by default, meaning that the adjacency SID has only local significance. The **v4**-flag set to 0 means that the adjacency SID references to an adjacency with outgoing IPv4 encapsulation.

Another way to display adjacency SID information is using the **show router isis adjacency detail** command.

```
[/]
A:admin@PE-1# show router isis adjacency "int-PE-1-PE-2" detail

=====
Rtr Base ISIS Instance 0 Adjacency (detail)
=====
Hostname       : PE-2
SystemID       : 1920.0000.2002                SNPA           : 04:14:01:01:00:02
Interface      : int-PE-1-PE-2                Up Time        : 0d 00:12:53
State          : Up                          Priority        : 0
Nbr Sys Typ    : L2                          L. Circ Typ    : L2
Hold Time     : 19                          Max Hold       : 27
Adj Level      : L2                          MT Enabled     : No
Topology       : Unicast

IPv6 Neighbor  : ::
IPv4 Neighbor  : 192.168.12.2
IPv4 Adj SID  : Label 524287
Restart Support : Disabled
Restart Status  : Not currently being helped
Restart Supressed : Disabled
Number of Restarts: 0
Last Restart at   : Never

=====
```

```
[/]
A:admin@PE-1# show router isis adjacency "int-PE-1-PE-7" detail

=====
Rtr Base ISIS Instance 0 Adjacency (detail)
=====
Hostname       : PE-7
SystemID       : 1920.0000.2007                SNPA           : 04:27:01:01:00:01
Interface      : int-PE-1-PE-7                Up Time        : 0d 00:10:41
State          : Up                          Priority        : 0
Nbr Sys Typ    : L2                          L. Circ Typ    : L2
Hold Time     : 25                          Max Hold       : 27
Adj Level      : L2                          MT Enabled     : No
Topology       : Unicast

IPv6 Neighbor  : ::
IPv4 Neighbor  : 192.168.17.2
IPv4 Adj SID  : Label 524286
Restart Support : Disabled
Restart Status  : Not currently being helped
Restart Supressed : Disabled
Number of Restarts: 0
Last Restart at   : Never
```

Finally, when enabling the SR context within the IGP instance, the SR module resolves received prefixes with prefix SID sub-TLVs present. As a result, MPLS data plane resources are consumed. The ILM is programmed with a swap operation and the label-to-next-hop-label-forwarding-entry (LTN) with a push operation, both pointing to the primary and/or LFA next-hop label forwarding entry (NHLFE). Also, an SR tunnel is added in the Tunnel Table Manager (TTM). As a result, an SR shortest path tunnel is set up to each other router that is part of the SR domain. Now, SR shortest path tunnels can be used for all users of TTM.

Example 1: VPRN service with LFA and RLFA enabled

In the network topology of [Figure 364: Example topology](#), no LDP and RSVP-TE signaling protocols are enabled. Each router of the SR domain has a full mesh of SR shortest path tunnels to the other routers, and no LDP and RSVP-TE LSPs are present. For example, on PE-1, the TTM looks as follows:

```
[/]
A:admin@PE-1# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner      Encap TunnelId Pref  Nexthop      Metric
  Color
-----
192.0.2.2/32         isis (0)  MPLS  524291   11   192.168.12.2  10
192.0.2.3/32         isis (0)  MPLS  524292   11   192.168.12.2  20
192.0.2.4/32         isis (0)  MPLS  524293   11   192.168.12.2  30
192.0.2.5/32         isis (0)  MPLS  524296   11   192.168.17.2  30
192.0.2.6/32         isis (0)  MPLS  524294   11   192.168.17.2  20
192.0.2.7/32         isis (0)  MPLS  524295   11   192.168.17.2  10
192.168.12.2/32      isis (0)  MPLS  524289   11   192.168.12.2   0
192.168.17.2/32      isis (0)  MPLS  524290   11   192.168.17.2   0
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====
```

The objective is to configure a VPRN between PE-1 and PE-7, using SR shortest path tunnels as transport tunnel. The configuration is as follows:

```
# on PE-1:
configure {
  service {
    vprn "VPRN100" {
      admin-state enable
      service-id 100
      customer "1"
      autonomous-system 64496
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "64496:10001"
          vrf-target {
            community "target:64496:100"
          }
        }
        auto-bind-tunnel {
          resolution any
        }
      }
    }
  }
}
```

```

    }
  }
}
interface "loopback" {
  loopback true
  ipv4 {
    primary {
      address 192.0.1.1
      prefix-length 32
    }
  }
}
}

```

```

# on PE-7:
configure {
  service {
    vprn "VPRN100" {
      admin-state enable
      service-id 100
      customer "1"
      autonomous-system 64496
      bgp-ipvpn {
        mpls {
          admin-state enable
          route-distinguisher "64496:10007"
          vrf-target {
            community "target:64496:100"
          }
          auto-bind-tunnel {
            resolution any
          }
        }
      }
    }
    interface "loopback" {
      loopback true
      ipv4 {
        primary {
          address 192.0.1.7
          prefix-length 32
        }
      }
    }
  }
}

```

Within the VPRN service configuration, a loopback interface is created on both PEs to verify the transport mechanism. Tunnel information displaying the MPLS label value is retrieved using the **show router fp-tunnel-table <slot number>** command, as follows:

```

[~/]A:admin@PE-1# show router fp-tunnel-table 1
=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol                                Tunnel-ID
  Lbl
  NextHop
  Lbl      (backup)                        Intf/Tunnel
  NextHop  (backup)
-----
192.0.2.2/32                               SR-ISIS-0                               524291

```

```

20002
  192.168.12.2
192.0.2.3/32 SR-ISIS-0 1/1/1 524292
20003
  192.168.12.2
192.0.2.4/32 SR-ISIS-0 1/1/1 524293
20004
  192.168.12.2
192.0.2.5/32 SR-ISIS-0 1/1/1 524296
20005
  192.168.17.2
192.0.2.6/32 SR-ISIS-0 1/1/2 524294
20006
  192.168.17.2
192.0.2.7/32 SR-ISIS-0 1/1/2 524295
20007
  192.168.17.2
192.168.12.2/32 SR 1/1/2 524289
  3
  192.168.12.2
192.168.17.2/32 SR 1/1/1 524290
  3
  192.168.17.2 1/1/2
-----
Total Entries : 8
-----
=====

```

This means that, when traffic arrives on PE-1, the MPLS label 20007 is pushed to reach destination PE-7. Because, in this example, the prefix SID index range global mode is used, the value 20007 comes from the start label on PE-7 (first label of the SRGB, which is 20000, plus the configured index value of node SID PE-7 (7)), so 20007.

Enabling prefix LFA within the IS-IS context on PE-1 will enable LFA/FRR protection. Next-hop LFA protection is present for node PE-4, node PE-5, and the link between PE-4 and PE-5, as follows:

```

# on PE-1:
configure {
  router "Base" {
    isis 0 {
      loopfree-alternate {
      }
    }
  }
}

```

```

[/]
A:admin@PE-1# show router isis lfa-coverage
=====
Rtr Base ISIS Instance 0 LFA Coverage
=====
Topology      Level  Node          IPv4           IPv6
-----
IPV4 Unicast  L1     0/0(0%)       3/11(27%)     0/0(0%)
IPV6 Unicast  L1     0/0(0%)       0/0(0%)       0/0(0%)
IPV4 Multicast L1     0/0(0%)       0/0(0%)       0/0(0%)
IPV6 Multicast L1     0/0(0%)       0/0(0%)       0/0(0%)
IPV4 Unicast  L2     2/6(33%)     3/11(27%)     0/0(0%)
IPV6 Unicast  L2     0/0(0%)       0/0(0%)       0/0(0%)
IPV4 Multicast L2     0/0(0%)       0/0(0%)       0/0(0%)
IPV6 Multicast L2     0/0(0%)       0/0(0%)       0/0(0%)

```

```
[/]A:admin@PE-1# show router route-table alternative
```

```
Route Table (Router: Base)
```

Dest	Prefix[Flags] Next Hop[Interface Name] Alt-NextHop	Type	Proto	Age	Metric Alt- Metric	Pref
192.0.2.1/32	system	Local	Local	00h54m06s	0	0
192.0.2.2/32	192.168.12.2	Remote	ISIS	00h44m31s	10	18
192.0.2.3/32	192.168.12.2	Remote	ISIS	00h44m15s	20	18
192.0.2.4/32	192.168.12.2 192.168.17.2 (LFA)	Remote	ISIS	00h44m14s	30 40	18
192.0.2.5/32	192.168.17.2 192.168.12.2 (LFA)	Remote	ISIS	00h44m08s	30 40	18
192.0.2.6/32	192.168.17.2	Remote	ISIS	00h44m08s	20	18
192.0.2.7/32	192.168.17.2	Remote	ISIS	00h44m08s	10	18
192.168.12.0/30	int-PE-1-PE-2	Local	Local	00h54m06s	0	0
192.168.17.0/30	int-PE-1-PE-7	Local	Local	00h54m06s	0	0
192.168.23.0/30	192.168.12.2	Remote	ISIS	00h53m35s	20	18
192.168.34.0/30	192.168.12.2	Remote	ISIS	00h53m05s	30	18
192.168.45.0/30	192.168.12.2 192.168.17.2 (LFA)	Remote	ISIS	00h52m35s	40 50	18
192.168.56.0/30	192.168.17.2	Remote	ISIS	00h51m24s	30	18
192.168.67.0/30	192.168.17.2	Remote	ISIS	00h51m24s	20	18

No. of Routes: 14

Flags: n = Number of times nexthop is repeated
Backup = BGP backup route
LFA = Loop-Free Alternate nexthop
S = Sticky ECMP requested

```
[/]A:admin@PE-1# show router fp-tunnel-table 1
```

```
IPv4 Tunnel Table Display
```

Legend: label stack is ordered from bottom-most to top-most

B - FRR Backup

Destination Lbl	Protocol	Tunnel-ID
NextHop		Intf/Tunnel

Lbl	(backup) NextHop (backup)		
192.0.2.2/32		SR-ISIS-0	524291
20002	192.168.12.2		1/1/1
192.0.2.3/32		SR-ISIS-0	524292
20003	192.168.12.2		1/1/1
192.0.2.4/32		SR-ISIS-0	524293
20004	192.168.12.2		1/1/1
20004	192.168.17.2(B)		1/1/2
192.0.2.5/32		SR-ISIS-0	524296
20005	192.168.17.2		1/1/2
20005	192.168.12.2(B)		1/1/1
192.0.2.6/32		SR-ISIS-0	524294
20006	192.168.17.2		1/1/2
192.0.2.7/32		SR-ISIS-0	524295
20007	192.168.17.2		1/1/2
192.168.12.2/32		SR	524289
3	192.168.12.2		1/1/1
192.168.17.2/32		SR	524290
3	192.168.17.2		1/1/2

Total Entries : 8			

=====			

[/]`A:admin@PE-1# show router tunnel-table detail`

=====
Tunnel Table (Router: Base)
=====

```

Destination      : 192.0.2.2/32
NextHop          : 192.168.12.2
Tunnel Flags     : entropy-label-capable
Age              : 00h01m02s
CBF Classes      : (Not Specified)
Owner            : isis (0)           Encap           : MPLS
Tunnel ID        : 524291             Preference      : 11
Tunnel Label     : 20002              Tunnel Metric   : 10
Tunnel MTU       : 1560              Max Label Stack : 1
-----
Destination      : 192.0.2.3/32
NextHop          : 192.168.12.2
Tunnel Flags     : entropy-label-capable
Age              : 00h01m02s
CBF Classes      : (Not Specified)
Owner            : isis (0)           Encap           : MPLS
Tunnel ID        : 524292             Preference      : 11
Tunnel Label     : 20003              Tunnel Metric   : 20
Tunnel MTU       : 1560              Max Label Stack : 1
-----
Destination      : 192.0.2.4/32 [L]
NextHop          : 192.168.12.2

```

```
Tunnel Flags : has-lfa entropy-label-capable
Age          : 00h01m01s
CBF Classes  : (Not Specified)
Owner       : isis (0)           Encap          : MPLS
Tunnel ID   : 524293           Preference     : 11
Tunnel Label : 20004           Tunnel Metric  : 30
Tunnel MTU  : 1560            Max Label Stack : 1
-----
Destination  : 192.0.2.5/32 [L]
NextHop      : 192.168.17.2
Tunnel Flags : has-lfa entropy-label-capable
Age          : 00h01m01s
CBF Classes  : (Not Specified)
Owner       : isis (0)           Encap          : MPLS
Tunnel ID   : 524296           Preference     : 11
Tunnel Label : 20005           Tunnel Metric  : 30
Tunnel MTU  : 1560            Max Label Stack : 1
-----
Destination  : 192.0.2.6/32
NextHop      : 192.168.17.2
Tunnel Flags : entropy-label-capable
Age          : 00h01m02s
CBF Classes  : (Not Specified)
Owner       : isis (0)           Encap          : MPLS
Tunnel ID   : 524294           Preference     : 11
Tunnel Label : 20006           Tunnel Metric  : 20
Tunnel MTU  : 1560            Max Label Stack : 1
-----
Destination  : 192.0.2.7/32
NextHop      : 192.168.17.2
Tunnel Flags : entropy-label-capable
Age          : 00h01m02s
CBF Classes  : (Not Specified)
Owner       : isis (0)           Encap          : MPLS
Tunnel ID   : 524295           Preference     : 11
Tunnel Label : 20007           Tunnel Metric  : 10
Tunnel MTU  : 1560            Max Label Stack : 1
-----
Destination  : 192.168.12.2/32
NextHop      : 192.168.12.2
Tunnel Flags : is-adjacency-tunnel
Age          : 00h01m02s
CBF Classes  : (Not Specified)
Owner       : isis (0)           Encap          : MPLS
Tunnel ID   : 524289           Preference     : 11
Tunnel Label : 3               Tunnel Metric  : 0
Tunnel MTU  : 1560            Max Label Stack : 1
-----
Destination  : 192.168.17.2/32
NextHop      : 192.168.17.2
Tunnel Flags : is-adjacency-tunnel
Age          : 00h01m02s
CBF Classes  : (Not Specified)
Owner       : isis (0)           Encap          : MPLS
Tunnel ID   : 524290           Preference     : 11
Tunnel Label : 3               Tunnel Metric  : 0
Tunnel MTU  : 1560            Max Label Stack : 1
-----
Number of tunnel-table entries : 8
Number of tunnel-table entries with LFA : 2
=====
```

When a failure occurs on the primary SR path (only applicable for prefix PE-4/PE-5 and the link between PE-4 and PE-5), the traffic takes the LFA backup SR path to the destination using the same MPLS label value.

To extend the LFA/FRR coverage, for example, to find an LFA protection for node PE-7, which is one of the VPRN service endpoints, RLFA can be enabled. RLFA creates a virtual LFA by using a repair tunnel to carry packets to a point in the network from where they will not be looped back to the source, but forwarded (SPF-based) toward the destination prefix.

The RLFA implementation uses the PQ algorithm. The node where RLFA is configured (PE-1 in this example) computes an extended P-space and a Q-space. The intersection of both spaces is called the PQ-node. This PQ node is the destination node of the repair tunnel using an SR shortest path tunnel. To compute both spaces, SPF is used.

In this example, IS-IS is used as the IGP, using a default metric value of 10 for all links. With the assumption that the link between PE-1 and PE-7 is broken, the calculation of both the extended P-space and the Q-space at PE-1 is as follows:

- extended P-space — An SPF computed from node PE-1 and rooted at PE-2. It is used to calculate the set of routers that are reachable without any path transiting the protected link between PE-1 and PE-7. The following nodes belong to the extended P-space: PE-2, PE-3, PE-4, and PE-5.
- Q-space — A reverse SPF computed from PE-1 and rooted from PE-7 (acting as destination proxy). It is used to calculate the set of routers that can reach PE-7 without transiting the protected link between PE-1 and PE-7. The nodes PE-4, PE-5, and PE-6 belong to the Q-space.

Possible PQ-nodes are PE-4 or PE-5, because they are in the intersection of both spaces.

RLFA is configured as follows:

```
# on PE-1:
configure {
  router "Base" {
    isis 0 {
      loopfree-alternate {
        remote-lfa
      }
    }
  }
}
```

The nodes PE-2, PE-3, PE-6, and PE-7 now have RLFA protection, whereas PE-4 and PE-5 have LFA protection.

```
[/]A:admin@PE-1# show router fp-tunnel-table 1
```

```
=====
IPv4 Tunnel Table Display
```

```
Legend: label stack is ordered from bottom-most to top-most
B - FRR Backup
```

```
=====
Destination                                Protocol                                Tunnel-ID
 Lbl                                         NextHop                                Intf/Tunnel
 Lbl      (backup)                          NextHop  (backup)
-----
192.0.2.2/32                                SR-ISIS-0                              524291
 20002
 192.168.12.2                               1/1/1
 20002/20005
```



```

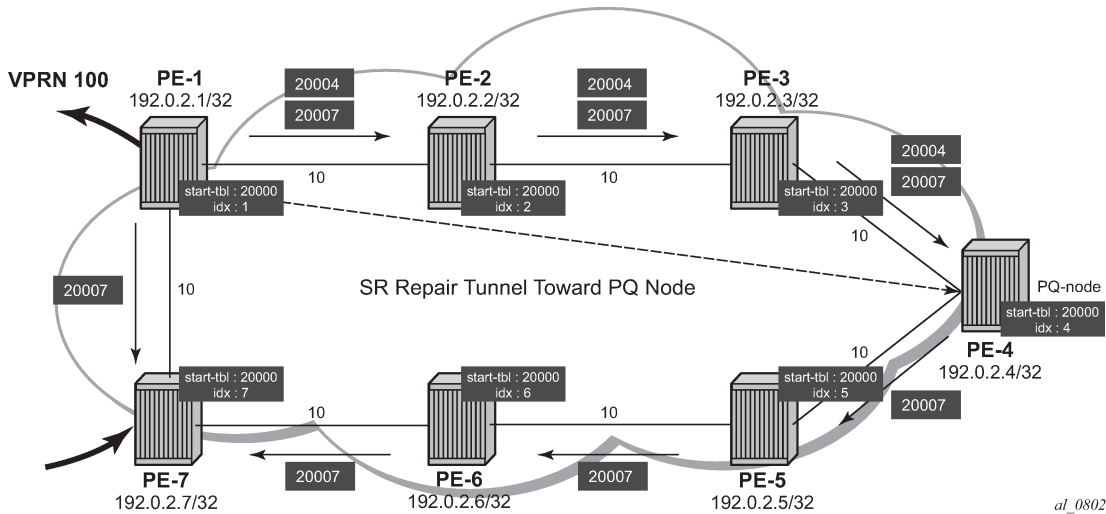
192.168.17.2(B)
192.0.2.3/32 SR-ISIS-0 1/1/2 524292
20003
192.168.12.2 1/1/1
20003/20005
192.168.17.2(B) 1/1/2
192.0.2.4/32 SR-ISIS-0 524293
20004
192.168.12.2 1/1/1
20004
192.168.17.2(B) 1/1/2
192.0.2.5/32 SR-ISIS-0 524296
20005
192.168.17.2 1/1/2
20005
192.168.12.2(B) 1/1/1
192.0.2.6/32 SR-ISIS-0 524294
20006
192.168.17.2 1/1/2
20006/20004
192.168.12.2(B) 1/1/1
192.0.2.7/32 SR-ISIS-0 524295
20007
192.168.17.2 1/1/2
20007/20004
192.168.12.2(B) 1/1/1
192.168.12.2/32 SR 524289
3
192.168.12.2 1/1/1
20002/20005
192.168.17.2(B) 1/1/2
192.168.17.2/32 SR 524290
3
192.168.17.2 1/1/2
20007/20004
192.168.12.2(B) 1/1/1
-----
Total Entries : 8
-----
=====

```

The main difference between normal prefix LFA and RLFA is that for RLFA a two-MPLS label stack is pushed by the head-end node (PE-1). The top label is the SR-label to reach the PQ node (for example, 20004 for PE-4) and the bottom label is the SR-label to reach the destination node (for example, 20007 for PE-7). The notation inside the show command is bottom-label/top-label.

[Figure 365: RLFA traffic path during protection](#) illustrates the RLFA traffic path protecting the link between PE-1 and PE-7:

Figure 365: RLFA traffic path during protection



Inside the TTM, a tunnel-flag, **has-lfa**, is set for all destination nodes that have LFA protection available. The last two tunnels are adjacency tunnels and have in addition the flag **is-adjacency-tunnel**.

```
[/]A:admin@PE-1# show router tunnel-table detail
```

```
=====
Tunnel Table (Router: Base)
=====
```

```
Destination      : 192.0.2.2/32 [L]
NextHop          : 192.168.12.2
Tunnel Flags     : has-lfa entropy-label-capable
Age              : 00h00m41s
CBF Classes      : (Not Specified)
Owner            : isis (0)           Encap           : MPLS
Tunnel ID        : 524291             Preference      : 11
Tunnel Label     : 20002              Tunnel Metric   : 10
Tunnel MTU       : 1556              Max Label Stack: 2
```

```
-----
Destination      : 192.0.2.3/32 [L]
NextHop          : 192.168.12.2
Tunnel Flags     : has-lfa entropy-label-capable
Age              : 00h00m41s
CBF Classes      : (Not Specified)
Owner            : isis (0)           Encap           : MPLS
Tunnel ID        : 524292             Preference      : 11
Tunnel Label     : 20003              Tunnel Metric   : 20
Tunnel MTU       : 1556              Max Label Stack: 2
```

```
-----
Destination      : 192.0.2.4/32 [L]
NextHop          : 192.168.12.2
Tunnel Flags     : has-lfa entropy-label-capable
Age              : 00h00m42s
CBF Classes      : (Not Specified)
Owner            : isis (0)           Encap           : MPLS
Tunnel ID        : 524293             Preference      : 11
Tunnel Label     : 20004              Tunnel Metric   : 30
Tunnel MTU       : 1556              Max Label Stack: 2
```

```
-----
Destination      : 192.0.2.5/32 [L]
```

```

NextHop      : 192.168.17.2
Tunnel Flags : has-lfa entropy-label-capable
Age          : 00h00m42s
CBF Classes  : (Not Specified)
Owner       : isis (0)           Encap          : MPLS
Tunnel ID   : 524296           Preference     : 11
Tunnel Label : 20005           Tunnel Metric  : 30
Tunnel MTU  : 1556           Max Label Stack : 2
-----
Destination  : 192.0.2.6/32 [L]
NextHop      : 192.168.17.2
Tunnel Flags : has-lfa entropy-label-capable
Age          : 00h00m41s
CBF Classes  : (Not Specified)
Owner       : isis (0)           Encap          : MPLS
Tunnel ID   : 524294           Preference     : 11
Tunnel Label : 20006           Tunnel Metric  : 20
Tunnel MTU  : 1556           Max Label Stack : 2
-----
Destination  : 192.0.2.7/32 [L]
NextHop      : 192.168.17.2
Tunnel Flags : has-lfa entropy-label-capable
Age          : 00h00m41s
CBF Classes  : (Not Specified)
Owner       : isis (0)           Encap          : MPLS
Tunnel ID   : 524295           Preference     : 11
Tunnel Label : 20007           Tunnel Metric  : 10
Tunnel MTU  : 1556           Max Label Stack : 2
-----
Destination  : 192.168.12.2/32 [L]
NextHop      : 192.168.12.2
Tunnel Flags : has-lfa is-adjacency-tunnel
Age          : 00h00m41s
CBF Classes  : (Not Specified)
Owner       : isis (0)           Encap          : MPLS
Tunnel ID   : 524289           Preference     : 11
Tunnel Label : 3               Tunnel Metric  : 0
Tunnel MTU  : 1556           Max Label Stack : 2
-----
Destination  : 192.168.17.2/32 [L]
NextHop      : 192.168.17.2
Tunnel Flags : has-lfa is-adjacency-tunnel
Age          : 00h00m41s
CBF Classes  : (Not Specified)
Owner       : isis (0)           Encap          : MPLS
Tunnel ID   : 524290           Preference     : 11
Tunnel Label : 3               Tunnel Metric  : 0
Tunnel MTU  : 1556           Max Label Stack : 2
-----
Number of tunnel-table entries      : 8
Number of tunnel-table entries with LFA : 8
=====

```

Verification of the loopback address configured within the VPRN service context on PE-7 (using loopback address 192.0.1.7/32) shows that an SR shortest path tunnel is used as the transport mechanism:

```

[/]A:admin@PE-1# show router 100 route-table 192.0.1.7/32 extensive

=====
Route Table (Service: 100)
=====
Dest Prefix      : 192.0.1.7/32

```

```

Protocol          : BGP_VPN
Age               : 00h00m42s
Preference       : 170
Indirect Next-Hop : 192.0.2.7
Label            : 524285
QoS              : Priority=n/c, FC=n/c
Source-Class     : 0
Dest-Class      : 0
ECMP-Weight      : N/A
Resolving Next-Hop : 192.0.2.7 (SR-ISIS tunnel:524295)
Label           : 524285
Metric          : 10
ECMP-Weight     : N/A

```

```
-----
No. of Destinations: 1
=====
```

Example 2: TTM preference with VPRN service

The following example is a variant on the previous example. The difference in this example is that, in addition to SR, LDP and RSVP-TE are also enabled between PE-1 and PE-7. A single RSVP LSP is configured originating at PE-1 and terminating at PE-7.

The objective of this example is to show the difference in protocol preference within TTM and how to influence the default behavior. This can be useful in case of migration scenarios from a non-SR environment toward a hybrid environment having LDP/RSVP and SR enabled.

In the following example, LFA/RLFA is no longer configured on the PE-1 node:

```

# on PE-1:
configure exclusive
  router "Base" {
    isis 0 {
      delete loopfree-alternate
    }
  }
commit

```

```

# on PE-1:
configure {
  router "Base" {
    mpls {
      admin-state enable
      interface "int-PE-1-PE-7" {
      }
      path "dyn" {
        admin-state enable
      }
      lsp "LSP-PE-1-PE-7" {
        admin-state enable
        type p2p-rsvp
        to 192.0.2.7
        primary "dyn"
      }
    }
  }
  rsvp {
    admin-state enable
    interface "int-PE-1-PE-7" {
    }
  }
  ldp {

```

```

        interface-parameters {
            interface "int-PE-1-PE-7" {
                ipv4 {
                }
            }
        }
    }
}

```

```

# on PE-7:
configure {
    router "Base" {
        mpls {
            admin-state enable
            interface "int-PE-7-PE-1" {
            }
        }
        rsvp {
            admin-state enable
            interface "int-PE-7-PE-1" {
            }
        }
        ldp {
            interface-parameters {
                interface "int-PE-7-PE-1" {
                    ipv4 {
                    }
                }
            }
        }
    }
}

```

By enabling LDP and RSVP between PE-1 and PE-7, the TTM on both nodes changed. With the VPRN service between PE-1 and PE-7 of example 1, only those two specific service endpoints are displayed:

```
[/]A:admin@PE-1# show router tunnel-table 192.0.2.7
```

```
=====
IPv4 Tunnel Table (Router: Base)
=====
```

Destination Color	Owner	Encap	TunnelId	Pref	Nexthop	Metric
192.0.2.7/32	rsvp	MPLS	1	7	192.168.17.2	10
192.0.2.7/32	ldp	MPLS	65537	9	192.168.17.2	10
192.0.2.7/32	isis (0)	MPLS	524295	11	192.168.17.2	10

```
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
=====
```

```
[/]A:admin@PE-7# show router tunnel-table 192.0.2.1
```

```
=====
IPv4 Tunnel Table (Router: Base)
=====
```

Destination Color	Owner	Encap	TunnelId	Pref	Nexthop	Metric
192.0.2.1/32	ldp	MPLS	65537	9	192.168.17.1	10

```

192.0.2.1/32      isis (0)  MPLS  524292   11    192.168.17.1  10
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====

```

On node PE-1, an RSVP LSP, an LDP LSP, and an SR shortest path tunnel (using IS-IS) are present. Because the VPRN service has **auto-bind-tunnel resolution any** enabled, the protocol type with the highest TTM preference (meaning the lowest absolute preference value in TTM) is taken; in this case, the RSVP LSP. This can be verified for the configured loopback address within the VPRN service context, as follows:

```

[~/]A:admin@PE-1# show router 100 route-table 192.0.1.7/32 extensive

=====
Route Table (Service: 100)
=====
Dest Prefix      : 192.0.1.7/32
Protocol         : BGP_VPN
Age              : 00h02m22s
Preference      : 170
Indirect Next-Hop : 192.0.2.7
Label           : 524285
QoS              : Priority=n/c, FC=n/c
Source-Class    : 0
Dest-Class      : 0
ECMP-Weight     : N/A
Resolving Next-Hop : 192.0.2.7 (RSVP tunnel:1)
Label          : 524285
Metric          : 10
ECMP-Weight     : N/A
-----
No. of Destinations: 1
=====

```

On node PE-7, only an LDP LSP and an SR shortest path tunnel (using IS-IS) are present. Because the VPRN service has **auto-bind-tunnel resolution any** enabled, the protocol type with highest TTM preference (meaning the lowest absolute preference value in TTM) is taken; in this case, the LDP LSP. This can be verified for the configured loopback address within the VPRN service context, as follows:

```

[~/]A:admin@PE-7# show router 100 route-table 192.0.1.1/32 extensive

=====
Route Table (Service: 100)
=====
Dest Prefix      : 192.0.1.1/32
Protocol         : BGP_VPN
Age              : 00h02m53s
Preference      : 170
Indirect Next-Hop : 192.0.2.1
Label           : 524285
QoS              : Priority=n/c, FC=n/c
Source-Class    : 0
Dest-Class      : 0
ECMP-Weight     : N/A
Resolving Next-Hop : 192.0.2.1 (LDP tunnel)
Label          : 524285
Metric          : 10
ECMP-Weight     : N/A

```

```
-----
No. of Destinations: 1
=====
```

Some configuration changes are possible to change this default behavior:

- It is possible to change the **auto-bind-tunnel resolution any** command into **auto-bind-tunnel resolution filter**. Because this is a service-specific parameter, the operator has the choice to only configure this on one specific service endpoint. From a migration point of view, a smooth and easy SR migration is possible, not affecting any other deployed services on this node.
- It is possible to change the SR tunnel-table protocol preference on a node. From a migration point of view, this affects all services initiating on this node.

Using the current example, PE-1 implements the auto-bind-tunnel change (option 1), while PE-7 implements the TTM preference change (option 2).

A resolution-filter context within VPRN 100 on node PE-1 must be created. The example uses a **resolution-filter** context, which uses a filter to only allow SR shortest path tunnels (IS-IS based). The **auto-bind-tunnel resolution any** command is changed into **resolution filter** on PE-1, as follows:

```
# on PE-1:
configure {
  service {
    vprn "VPRN100" {
      bgp-ipvpn {
        mpls {
          auto-bind-tunnel {
            resolution filter
            resolution-filter {
              sr-isis true
            }
          }
        }
      }
    }
  }
}
```

As a result, the RSVP LSP is no longer used. Instead, the SR shortest path tunnel is used for the traffic from PE-1 to PE-7:

```
[/]/A:admin@PE-1# show router 100 route-table 192.0.1.7/32 extensive
```

```
=====
Route Table (Service: 100)
=====
```

```
Dest Prefix      : 192.0.1.7/32
Protocol         : BGP_VPN
Age              : 00h00m14s
Preference      : 170
Indirect Next-Hop : 192.0.2.7
Label           : 524285
QoS              : Priority=n/c, FC=n/c
Source-Class    : 0
Dest-Class      : 0
ECMP-Weight     : N/A
Resolving Next-Hop : 192.0.2.7 (SR-ISIS tunnel:524295)
Label           : 524285
Metric          : 10
ECMP-Weight     : N/A
```

```
-----
No. of Destinations: 1
```

The VPRN service on node PE-7 is still using the LDP LSP as transport mechanism to reach node PE-1 at this point. Because the previous change is only done within VPRN 100 on PE-1, only the direction from PE-1 to PE-7 is affected.

Another way to influence the default TTM preference is shown as follows on the PE-7 node. Using the default behavior, the LDP LSP is used, because of the preference value of 9. If the SR tunnel table preference value is lowered to a value smaller than LDP, for instance 4, the SR shortest path tunnels originating on this node will always have preference compared to LDP LSP. On PE-7, the SR tunnel table preference is configured with a value of 4, as follows:

```
# on PE-7:
configure {
  router "Base" {
    isis 0 {
      segment-routing {
        tunnel-table-pref 4
      }
    }
  }
}
```

```
[/]A:admin@PE-7# show router tunnel-table 192.0.2.1
```

```
=====  
IPv4 Tunnel Table (Router: Base)  
=====
```

Destination Color	Owner	Encap	TunnelId	Pref	Nexthop	Metric
192.0.2.1/32	isis (0)	MPLS	524292	4	192.168.17.1	10
192.0.2.1/32	ldp	MPLS	65537	9	192.168.17.1	10

```
-----  
Flags: B = BGP or MPLS backup hop available  
L = Loop-Free Alternate (LFA) hop available  
E = Inactive best-external BGP route  
k = RIB-API or Forwarding Policy backup hop  
=====
```

As a result, the LDP LSP is no longer used and the SR shortest path tunnel is the preferred transport tunnel:

```
[/]A:admin@PE-7# show router 100 route-table 192.0.1.1/32 extensive
```

```
=====  
Route Table (Service: 100)  
=====
```

```
Dest Prefix      : 192.0.1.1/32  
Protocol        : BGP_VPN  
Age             : 00h00m33s  
Preference      : 170  
Indirect Next-Hop : 192.0.2.1  
Label           : 524285  
QoS             : Priority=n/c, FC=n/c  
Source-Class    : 0  
Dest-Class      : 0  
ECMP-Weight     : N/A  
Resolving Next-Hop : 192.0.2.1 (SR-ISIS tunnel:524292)  
Label           : 524285  
Metric          : 10
```



```

ECMP-Weight      : N/A
-----
No. of Destinations: 1
=====

```

At this point, within the VPRN service, the SR shortest path tunnels are used bidirectionally between PE-1 and PE-7.

If, for example, an operator configures explicit SDP binding within the same VPRN service on both endpoints, the explicit SDPs will always have preference. In this example, manual SDPs are configured on nodes PE-1 and PE-7, both using LDP, as follows:

```

# on PE-1:
configure {
  service
    sdp 17 {
      admin-state enable
      delivery-type mpls
      ldp true
      far-end {
        ip-address 192.0.2.7
      }
    }
  vprn "VPRN100" {
    spoke-sdp 17:100 {
    }
  }
}

```

```

# on PE-7:
configure {
  service {
    sdp 71 {
      admin-state enable
      delivery-type mpls
      ldp true
      far-end {
        ip-address 192.0.2.1
      }
    }
  }
  vprn "VPRN100" {
    spoke-sdp 71:100 {
    }
  }
}

```

As a result, SR shortest path tunnels are no longer used, but rather LDP-based SDPs are used instead:

```

[/]A:admin@PE-1# show router 100 route-table 192.0.1.7/32 extensive

=====
Route Table (Service: 100)
=====
Dest Prefix      : 192.0.1.7/32
Protocol         : BGP_VPN
Age              : 00h00m40s
Preference      : 170
Indirect Next-Hop : 192.0.2.7
Label           : 524285
QoS              : Priority=n/c, FC=n/c
Source-Class     : 0

```

```
Dest-Class      : 0
ECMP-Weight    : N/A
Resolving Next-Hop : 192.0.2.7 (SDP tunnel:17)
  Label        : 524285
  Metric       : 0
  ECMP-Weight  : N/A
-----
No. of Destinations: 1
=====
```

```
[/]A:admin@PE-7# show router 100 route-table 192.0.1.1/32 extensive
=====
Route Table (Service: 100)
=====
Dest Prefix      : 192.0.1.1/32
Protocol        : BGP_VPN
Age             : 00h00m52s
Preference      : 170
Indirect Next-Hop : 192.0.2.1
  Label         : 524285
  QoS           : Priority=n/c, FC=n/c
  Source-Class  : 0
  Dest-Class    : 0
  ECMP-Weight   : N/A
Resolving Next-Hop : 192.0.2.1 (SDP tunnel:71)
  Label        : 524285
  Metric       : 0
  ECMP-Weight  : N/A
-----
No. of Destinations: 1
=====
```

Conclusion

Segment Routing is a technique using extensions of the existing link state protocols, and using existing MPLS or IPv6 infrastructure as the data plane. It is a source routing technique similar to RSVP-TE, but without the need to run an extra signaling protocol. SR also avoids other scaling restrictions of associated RSVP-TE, such as midpoint state. SR is simple to control and operate because the intelligence and state are part of the packet, not held by the network. Other benefits are that SR can be introduced in an incremental way using different migration scenarios to assure a smooth transition.

SR-TE LSP Path Computation Using Local CSPF

This chapter describes the SR-TE LSP path computation using local CSPF.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

This chapter was initially written for SR OS Release 19.10.R1, but the MD-CLI in the current edition corresponds to SR OS Release 21.2.R1. Local CSPF can be used in IPv4 SR-TE LSP primary and secondary path computation in SR OS Release 19.7.R1, and later.

Overview

Segment Routing with Traffic Engineering Label Switched Paths (SR-TE LSPs) can be computed using:

- hop-to-label (IP-to-label) translation (default; **path-computation-method** not specified)
- Path Computation Element (PCE) path computation (**path-computation-method pce**)
- local Constrained Shortest Path First (CSPF) (**path-computation-method local-cspf**)

Hop-to-label path computation

SR-TE LSP path computation uses hop-to-label translation as the default computation method. The PCC interrogates the TE database, and translates any hop configured in the applied path statement to a Node SID (N-SID) or Adjacency SID (A-SID), to produce a list of segment IDs. Strict hops are mapped to adjacency SIDs; loose hops are mapped to node SIDs. The destination address in the LSP configuration implies a final loose hop.

PCE path computation

SR-TE LSP path computation can also be performed using an external PCE controller. In this case, the PCC maintains a Path Computation Element Protocol (PCEP) session with the PCE and the path computation is done as follows:

- the PCC sends a PCReq requesting a path
- the PCE replies with a PCReply including a path (if available). This path contains a segment list.
- Optionally, the PCC sends a path status report (PCRpt) to the PCE. However, the PCC may also delegate the control of the path to the PCE.

PCE path computation is supported for SR-TE LSPs, but not for SR-TE LSP templates. You cannot have PCE path computation for SR-TE LSPs that use LSP templates **one-hop-p2p-srte** or **mesh-p2p-srte** auto-LSPs. PCE path computation is not further treated in this chapter.

Local CSPF path computation

SR-TE LSP path computation using local CSPF can be used in single-area OSPFv2 or single-level IS-IS IGP instances. More complex LSP path computations, or when the network is expanded into multiple IGP areas or instances, require an external PCE.

One of the major changes to the SR-TE algorithm from RSVP-TE CSPF is that SR-TE does not require each router to be TE enabled: the links do not have to be TE links. Provided that the routers at each end of the link are SR enabled, local CSPF will calculate an end-to-end path.

Full CSPF path computation on the head-end router (PCC) results in a full explicit path to the destination. The PCC calculates an end-to-end path and the following applies:

- The computed path is a full explicit TE path.
- Each link is represented by an adjacency SID or adjacency set SID.
- CSPF returns a label stack list of adjacency SIDs or adjacency set SIDs.

Like RSVP-TE LSPs, an SR-TE LSP can be resigned when a timer expires or when an operator issues a `tools` command.

Paths computed by local CSPF contain an adjacency SID for each link in the path and the stack may contain numerous labels. If the **max-sr-labels** value may be exceeded or the maximum segment depth of a downstream router may be less than the calculated LSP label stack size, the label stack can be reduced. The label reduction capability can replace a series of adjacency SIDs with a node SID. For loose-hop path computation, node SIDs can be used or a combination of node and adjacency SIDs.

Local CSPF is supported on both primary and secondary standby paths of an SR-TE LSP. Local CSPF path calculation can be used for RSVP-TE LSP as well as for SR-TE LSP templates.

Local CSPF path computation and SR protected interfaces

When SR is enabled and IGP adjacency is established over a link, the router advertises an adjacency SID in the adjacency SID sub-TLV. When Loop-Free Alternate (LFA), Remote LFA (RLFA), or Topology-Independent LFA (TI-LFA) is enabled, protected adjacencies have the backup flag (B-flag) set in the adjacency SID sub-TLV. Each adjacency is available for SID protection when LFA, RLFA, or TI-LFA is enabled. It is possible to remove this on a specific link (**sid-protection false**).

Adjacency sets are specified in an adjacency-set sub-TLV as a single object. Adjacency sets never have the B-flag set and are always unprotected. However, each individual link in the adjacency set is protected. For more information about adjacency sets, see the [Parallel Adjacency Sets in Segment Routing](#) chapter.

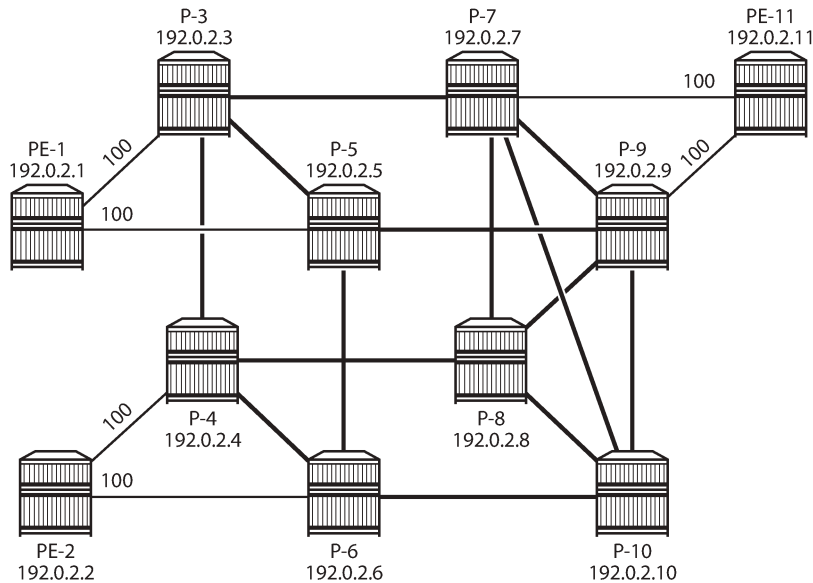
Local CSPF path calculation can set up a path that:

- only includes protected adjacencies (**local-sr-protection mandatory**)
- only includes unprotected adjacencies (**local-sr-protection none**)
- can include both protected and unprotected adjacencies (**local-sr-protection preferred** (default))

Configuration

Figure 366: Example topology shows the example topology.

Figure 366: Example topology



35620

The initial configuration on each of the nodes includes:

- Cards, MDAs, ports
- Router interfaces
- IS-IS enabled on all router interfaces (alternatively, OSPFv2 can be used as IGP)
 - The interfaces in the core (between P-3, P-4, P-5, P-6, P-7, P-8, P-9, and P-10) have metric 10.
 - The access interfaces to and from PE-1, PE-2, and PE-11 have metric 100.
 - TE is enabled on the head-end routers.
- MPLS is enabled on the head-end routers.

For an in-depth description of the configuration of SR-ISIS, see the [Segment Routing with IS-IS Control Plane](#) chapter. On PE-2, the IS-IS configuration is as follows:

```
# on PE-2:
configure {
  router "Base" {
    mpls-labels {
      sr-labels {
        start 32000
        end 32999
      }
    }
  }
  isis 0 {
    admin-state enable
    advertise-router-capability area
    traffic-engineering true
    area-address [49.0001]
  }
}
```

```
segment-routing {
  admin-state enable
  prefix-sid-range {
    global
  }
}
interface "int-PE-2-P-4" {
  interface-type point-to-point
  level 1 {
    metric 100
  }
  level 2 {
    metric 100
  }
}
interface "int-PE-2-P-6" {
  interface-type point-to-point
  level 1 {
    metric 100
  }
  level 2 {
    metric 100
  }
}
interface "system" {
  ipv4-node-sid {
    index 2
  }
}
level 1 {
  wide-metrics-only true
}
level 2 {
  wide-metrics-only true
}
}
```

With this configuration, the node SID on PE-2 is $32000 + \text{index } 2 = 32002$. The configuration is similar on the other nodes.

On PE-2, the following SR-TE LSPs are configured toward PE-11:

- SR-TE LSP with empty path and:
 - hop-to-label path computation
 - local CSPF path computation without label stack reduction
 - local CSPF path computation with label stack reduction
- SR-TE LSP with path with two strict hops—P-4 and P-3—and:
 - hop-to-label path computation
 - local CSPF path computation without label stack reduction
 - local CSPF path computation with label stack reduction
- SR-TE LSP with path with two loose hops—P-3 and P-9—and:
 - hop-to-label path computation
 - local CSPF path computation without label stack reduction
 - local CSPF path computation with label stack reduction

SR-TE LSPs using empty path

The configuration of SR-TE LSPs is described in chapter [Segment Routing – Traffic Engineered Tunnels](#) chapter. On PE-2, the following SR-TE LSPs toward PE-11 are configured with an empty path. The path computation method is hop-to-label for the first SR-TE LSP and local CSPF for the second SR-TE LSP.

```
# on PE-2:
configure {
  router "Base" {
    mpls {
      admin-state enable
      path "empty_path" {
        admin-state enable
      }
      lsp "LSP-PE-2-PE-11_empty_path_hop-to-label" {
        admin-state enable
        type p2p-sr-te
        to 192.0.2.11
        max-sr-labels {
          additional-frr-labels 2
        }
        primary "empty_path" {
        }
      }
      lsp "LSP-PE-2-PE-11_empty_path_local-cspf" {
        admin-state enable
        type p2p-sr-te
        to 192.0.2.11
        path-computation-method local-cspf
        max-sr-labels {
          additional-frr-labels 2
        }
        primary "empty_path" {
        }
      }
    }
  }
}
```

With hop-to-label path computation, the destination 192.0.2.11 is an implied loose hop that will be mapped to the node SID 32011 of the destination PE-11, as follows:

```
[/]
A:admin@PE-2# show router mpls sr-te-lsp "LSP-PE-2-PE-11_empty_path_hop-to-label" path detail

=====
MPLS SR-TE LSP LSP-PE-2-PE-11_empty_path_hop-to-label
Path (Detail)
=====
Legend :
  S      - Strict
  A-SID  - Adjacency SID
  +      - Inherited
  L      - Loose
  N-SID  - Node SID
=====
-----
LSP SR-TE LSP-PE-2-PE-11_empty_path_hop-to-label
Path empty_path
-----
LSP Name      : LSP-PE-2-PE-11_empty_path_hop-to-label
Path LSP ID   : 24576
From          : 192.0.2.2
To            : 192.0.2.11
Admin State   : Up
Oper State    : Up
Path Name     : empty_path
```

```

Path Type      : Primary
Path Admin    : Up
Path Up Time  : 0d 00:03:36
Retry Limit   : 0
Retry Attempt : 0
Path Oper     : Up
Path Down Time : 0d 00:00:00
Retry Timer   : 30 sec
Next Retry In : 0 sec

PathCompMethod : none
MetricType    : igp
LocalSrProt   : preferred
LabelStackRed : Disabled
OperPathCompMethod: none
Oper MetricType : igp
Oper LocalSrProt : N/A
Oper LabelStackRed: N/A

Bandwidth     : No Reservation
Hop Limit     : 255
Setup Priority : 7
Hold Priority  : 0
Inter-area    : N/A
Oper Bandwidth : 0 Mbps
Oper HopLimit : 255
Oper SetupPriority: 7
Oper HoldPriority : 0

PCE Updt ID   : 0
PCE Upd Fail Code: noError
PCE Updt State : None

PCE Report    : Disabled+
PCE Control   : Disabled
Oper PCE Report : Disabled
Oper PCE Control : Disabled

Include Groups :
None
Exclude Groups :
None
Last Resignal : n/a
Oper IncludeGroups:
None
Oper ExcludeGroups:
None

IGP/TE Metric : 16777215
Oper MTU      : 8978
Oper Metric   : 16777215
Failure Code  : noError
Failure Node  : n/a
Path Trans    : 1
Explicit Hops :
No Hops Specified
Actual Hops   :
192.0.2.11(192.0.2.11) (N-SID) Record Label : 32011

BFD Configuration and State
Template      : None
Enable       : False
WaitForUpTimer : 4 sec
WaitForUpTmLeft : 0
StartFail Rsn : N/A
Ping Interval : N/A
State        : notApplicable
OperWaitForUpTimer: 0 sec

```

With local CSPF path computation, the SR-TE path contains contiguous strict hops with A-SIDs. Several ECMP paths are available and, in this case, the path goes from PE-2 via P-6, P-10, and P-7 to the destination PE-11, as follows:

```

[/]
A:admin@PE-2# show router mpls sr-te-lsp "LSP-PE-2-PE-11_empty_path_local-cspf" path detail

=====
MPLS SR-TE LSP LSP-PE-2-PE-11_empty_path_local-cspf
Path (Detail)
=====
Legend :
  S      - Strict
  A-SID  - Adjacency SID
  +      - Inherited
  L      - Loose
  N-SID  - Node SID

```



```

=====
-----
LSP SR-TE LSP-PE-2-PE-11_empty_path_local-cspf
Path empty_path
-----
LSP Name      : LSP-PE-2-PE-11_empty_path_local-cspf
Path LSP ID   : 16384
From          : 192.0.2.2
To            : 192.0.2.11
Admin State   : Up                               Oper State    : Up
Path Name     : empty_path
Path Type     : Primary
Path Admin    : Up                               Path Oper     : Up
Path Up Time  : 0d 00:03:36                       Path Down Time : 0d 00:00:00
Retry Limit   : 0                                 Retry Timer    : 30 sec
Retry Attempt : 0                                 Next Retry In  : 0 sec

PathCompMethod : local-cspf                    OperPathCompMethod: local-cspf
MetricType     : igp                               Oper MetricType : igp
LocalSrProt    : preferred                         Oper LocalSrProt : preferred
LabelStackRed  : Disabled                         Oper LabelStackRed: Disabled

Bandwidth      : No Reservation                    Oper Bandwidth  : 0 Mbps
Hop Limit      : 255                              Oper HopLimit   : 255
Setup Priority  : 7                               Oper SetupPriority: 7
Hold Priority   : 0                               Oper HoldPriority : 0
Inter-area     : N/A

---snip---

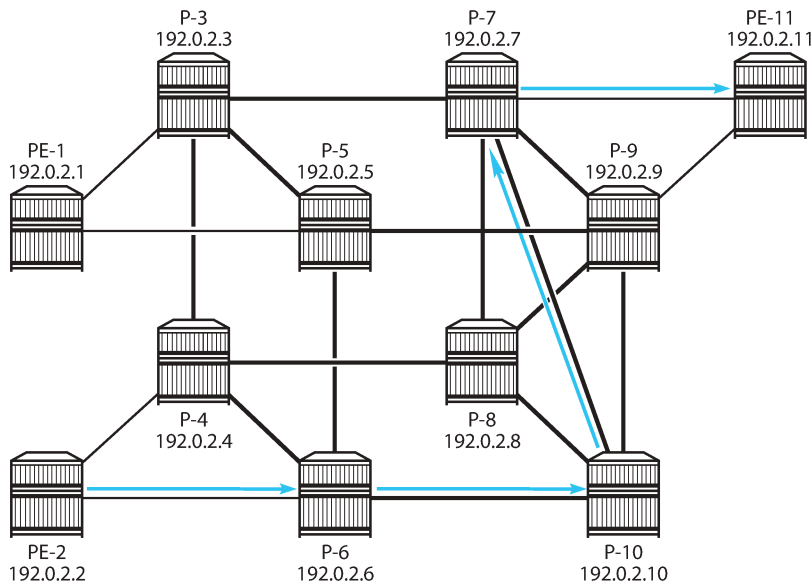
IGP/TE Metric  : 220                              Oper Metric     : 220
Oper MTU       : 8966                             Path Trans      : 1
Failure Code   : noError
Failure Node   : n/a
Explicit Hops  :
  No Hops Specified
Actual Hops    :
  192.168.26.2(192.0.2.6) (A-SID)                 Record Label    : 524286
-> 192.168.106.2(192.0.2.10) (A-SID)              Record Label    : 524284
-> 192.168.107.1(192.0.2.7) (A-SID)              Record Label    : 524284
-> 192.168.117.2(192.0.2.11) (A-SID)             Record Label    : 524283

---snip---
=====

```

The path goes from PE-2 via P-6, P-10, and P-7 to the destination PE-11, as shown in [Figure 367: Empty path from PE-2 to PE-11](#).

Figure 367: Empty path from PE-2 to PE-11



35621

The label stack of Adjacency SIDs (A-SIDs) can be reduced to a smaller number of Node SIDs (N-SIDs), or a combination of N-SIDs and A-SIDs, by enabling **label-stack-reduction**, as follows:

```
# on PE-2:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-2-PE-11_empty_path_local-cspf" {
        admin-state enable
        type p2p-sr-te
        to 192.0.2.11
        path-computation-method local-cspf
        label-stack-reduction true
        max-sr-labels {
          additional-frr-labels 2
        }
        primary "empty_path" {
        }
      }
    }
  }
}
```

Label stack reduction reduces the label stack to one or more node SIDs in segments, with each segment delimited by configured path hops. The path computed to the node SID must satisfy any required path constraints. In this example, the label stack is reduced to the N-SID of the destination PE-11, as follows:

```
[/]
A:admin@PE-2# show router mpls sr-te-lsp "LSP-PE-2-PE-11_empty_path_local-cspf" path detail

=====
MPLS SR-TE LSP LSP-PE-2-PE-11_empty_path_local-cspf
Path (Detail)
=====
Legend :
  S   - Strict
  A-SID - Adjacency SID
  L   - Loose
  N-SID - Node SID
```

```

+      - Inherited
=====
-----
LSP SR-TE LSP-PE-2-PE-11_empty_path_local-cspf
Path empty_path
-----
LSP Name      : LSP-PE-2-PE-11_empty_path_local-cspf
Path LSP ID   : 16384
From          : 192.0.2.2
To            : 192.0.2.11
Admin State   : Up                               Oper State    : Up
Path Name     : empty_path
Path Type     : Primary
Path Admin    : Up                               Path Oper     : Up
Path Up Time  : 0d 00:02:09                       Path Down Time : 0d 00:00:00
Retry Limit   : 0                                 Retry Timer    : 30 sec
Retry Attempt : 0                                 Next Retry In  : 0 sec

PathCompMethod : local-cspf                    OperPathCompMethod: local-cspf
MetricType     : igp                             Oper MetricType : igp
LocalSrProt    : preferred                       Oper LocalSrProt : preferred
LabelStackRed : Enabled                        Oper LabelStackRed: Enabled

Bandwidth      : No Reservation                   Oper Bandwidth  : 0 Mbps
Hop Limit      : 255                             Oper HopLimit   : 255
Setup Priority  : 7                               Oper SetupPriority: 7
Hold Priority   : 0                               Oper HoldPriority: 0
Inter-area     : N/A

---snip---

IGP/TE Metric  : 220                             Oper Metric     : 220
Oper MTU       : 8978                             Path Trans      : 3
Failure Code   : noError
Failure Node   : n/a
Explicit Hops  :
  No Hops Specified
Actual Hops    :
  192.0.2.11(192.0.2.11) (N-SID)                Record Label      : 32011

---snip---
=====

```

SR-TE LSPs using path with strict hops

In the following example, the SR-TE LSP path includes strict hops—that must be contiguous hops from the head-end router—and an implicit loose hop to the destination 192.0.2.11. The SR-TE LSPs on PE-2 are configured as follows:

```

# on PE-2:
configure {
  router "Base" {
    mpls {
      path "path_via-P-4-P-3_S" {
        admin-state enable
        hop 10 {
          ip-address 192.0.2.4
          type strict
        }
        hop 20 {

```

```

        ip-address 192.0.2.3
        type strict
    }
}
lsp "LSP-PE-2-PE-11_strict-hops_hop-to-label" {
    admin-state enable
    type p2p-sr-te
    to 192.0.2.11
    max-sr-labels {
        additional-frr-labels 2
    }
    primary "path_via-P-4-P-3_S" {
    }
}
lsp "LSP-PE-2-PE-11_strict-hops_local-cspf" {
    admin-state enable
    type p2p-sr-te
    to 192.0.2.11
    path-computation-method local-cspf
    max-sr-labels {
        additional-frr-labels 2
    }
    primary "path_via-P-4-P-3_S" {
    }
}
}

```

With hop-to-label path computation, strict hops are translated into adjacency SIDs, whereas loose hops are translated into node SIDs. In this example, the path has an A-SID to P-4 and an A-SID to P-3 followed by an N-SID to the destination PE-11, as follows:

```

[/]
A:admin@PE-2# show router mpls sr-te-lsp "LSP-PE-2-PE-11_strict-hops_hop-to-label" path detail
=====
MPLS SR-TE LSP LSP-PE-2-PE-11_strict-hops_hop-to-label
Path (Detail)
=====
Legend :
  S      - Strict
  A-SID  - Adjacency SID
  +      - Inherited
  L      - Loose
  N-SID  - Node SID
=====
-----
LSP SR-TE LSP-PE-2-PE-11_strict-hops_hop-to-label
Path path_via-P-4-P-3_S
-----
LSP Name      : LSP-PE-2-PE-11_strict-hops_hop-to-label
Path LSP ID   : 26624
From          : 192.0.2.2
To            : 192.0.2.11
Admin State   : Up
Oper State    : Up
Path Name     : path_via-P-4-P-3_S
Path Type     : Primary
Path Admin    : Up
Path Oper     : Up
Path Up Time  : 0d 00:03:36
Path Down Time : 0d 00:00:00
Retry Limit   : 0
Retry Timer   : 30 sec
Retry Attempt : 0
Next Retry In : 0 sec

PathCompMethod : none
MetricType     : igp
LocalSrProt    : preferred
LabelStackRed  : Disabled

OperPathCompMethod: none
Oper MetricType : igp
Oper LocalSrProt : N/A
Oper LabelStackRed: N/A

```

```

Bandwidth      : No Reservation      Oper Bandwidth   : 0 Mbps
Hop Limit      : 255                  Oper HopLimit    : 255
Setup Priority  : 7                    Oper SetupPriority: 7
Hold Priority   : 0                    Oper HoldPriority : 0
Inter-area     : N/A

---snip---

IGP/TE Metric  : 16777215             Oper Metric      : 16777215
Oper MTU       : 8970                 Path Trans       : 1
Failure Code   : noError
Failure Node   : n/a
Explicit Hops  :
                192.0.2.4(S)
                -> 192.0.2.3(S)
Actual Hops    :
                192.168.24.2(192.0.2.4) (A-SID)      Record Label    : 524287
                -> 192.168.34.1(192.0.2.3) (A-SID)      Record Label    : 524286
                -> 192.0.2.11(192.0.2.11) (N-SID)      Record Label    : 32011

---snip---

=====

```

With local CSPF path computation, the path is a sequence of A-SIDs, as follows:

```

[/]
A:admin@PE-2# show router mpls sr-te-lsp "LSP-PE-2-PE-11_strict-hops_local-cspf" path detail

=====
MPLS SR-TE LSP LSP-PE-2-PE-11_strict-hops_local-cspf
Path (Detail)
=====
Legend :
  S      - Strict
  A-SID  - Adjacency SID
  +      - Inherited
  L      - Loose
  N-SID  - Node SID

=====
-----
LSP SR-TE LSP-PE-2-PE-11_strict-hops_local-cspf
Path path_via-P-4-P-3_S
-----
LSP Name      : LSP-PE-2-PE-11_strict-hops_local-cspf
Path LSP ID   : 52224
From          : 192.0.2.2
To            : 192.0.2.11
Admin State   : Up
Oper State    : Up
Path Name     : path_via-P-4-P-3_S
Path Type     : Primary
Path Admin    : Up
Path Oper     : Up
Path Up Time  : 0d 00:03:36
Path Down Time : 0d 00:00:00
Retry Limit   : 0
Retry Timer   : 30 sec
Retry Attempt : 0
Next Retry In : 0 sec

PathCompMethod : local-cspf      OperPathCompMethod: local-cspf
MetricType     : igp             Oper MetricType    : igp
LocalSrProt    : preferred       Oper LocalSrProt   : preferred
LabelStackRed  : Disabled        Oper LabelStackRed : Disabled

Bandwidth      : No Reservation      Oper Bandwidth   : 0 Mbps
Hop Limit      : 255                  Oper HopLimit    : 255
Setup Priority  : 7                    Oper SetupPriority: 7
Hold Priority   : 0                    Oper HoldPriority : 0

```

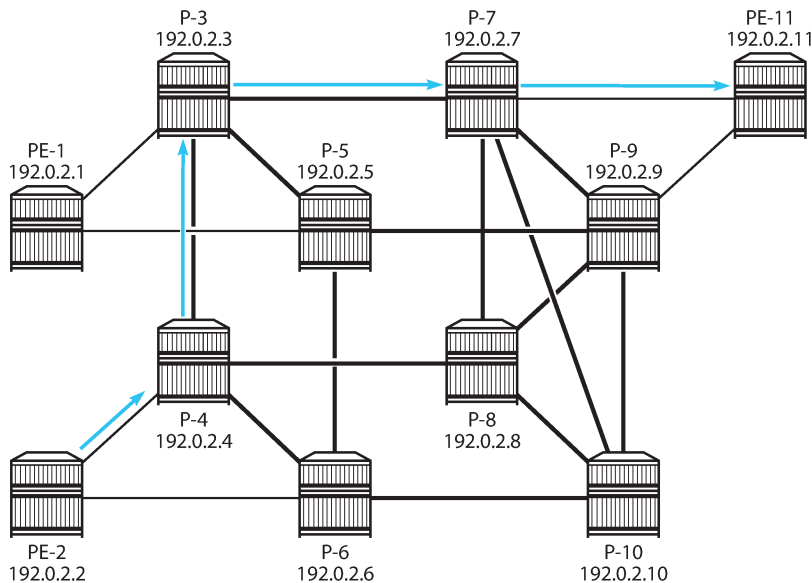
```

Inter-area      : N/A
---snip---
IGP/TE Metric   : 220                Oper Metric     : 220
Oper MTU        : 8966                Path Trans      : 1
Failure Code    : noError
Failure Node    : n/a
Explicit Hops   :
                192.0.2.4(S)
                -> 192.0.2.3(S)
Actual Hops     :
  192.168.24.2(192.0.2.4) (A-SID)      Record Label    : 524287
-> 192.168.34.1(192.0.2.3) (A-SID)      Record Label    : 524286
-> 192.168.37.2(192.0.2.7) (A-SID)      Record Label    : 524284
-> 192.168.117.2(192.0.2.11) (A-SID)     Record Label    : 524283
---snip---
=====

```

The path from PE-2 to PE-11 must go via P-4 and P-3. The loose hop from P-3 to the destination PE-11 is translated into an A-SID to P-7 followed by an A-SID to PE-11, as shown in [Figure 368: Path from PE-2 to PE-11 via strict hops P-4 and P-3](#).

Figure 368: Path from PE-2 to PE-11 via strict hops P-4 and P-3



35622

With label stack reduction, the configuration of the SR-TE LSP is as follows:

```

# on PE-2:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-2-PE-11_strict-hops_local-cspf" {
        admin-state enable
        type p2p-sr-te
        to 192.0.2.11
      }
    }
  }
}

```

```

    path-computation-method local-cspf
    label-stack-reduction true
    max-sr-labels {
        additional-frr-labels 2
    }
    primary "path_via-P-4-P-3_S" {
    }
}

```

Label stack reduction will reduce the label stack to one or more node SIDs in segments, with each segment delimited by configured path hops. The computed path to the node SID must satisfy any required path constraints. The calculated path from PE-2 to PE-11 via the strict hops P-4 and P-3 shows a series of N-SIDs, as follows:

```

[/]
A:admin@PE-2# show router mpls sr-te-lsp "LSP-PE-2-PE-11_strict-hops_local-cspf" path detail
=====
MPLS SR-TE LSP LSP-PE-2-PE-11_strict-hops_local-cspf
Path (Detail)
=====
Legend :
  S      - Strict          L      - Loose
  A-SID  - Adjacency SID  N-SID - Node SID
  +      - Inherited
=====
-----
LSP SR-TE LSP-PE-2-PE-11_strict-hops_local-cspf
Path path_via-P-4-P-3_S
-----
LSP Name      : LSP-PE-2-PE-11_strict-hops_local-cspf
Path LSP ID   : 52224
From          : 192.0.2.2
To            : 192.0.2.11
Admin State   : Up                Oper State    : Up
Path Name     : path_via-P-4-P-3_S
Path Type     : Primary
Path Admin    : Up                Path Oper     : Up
Path Up Time  : 0d 00:02:09       Path Down Time : 0d 00:00:00
Retry Limit   : 0                 Retry Timer    : 30 sec
Retry Attempt : 0                 Next Retry In  : 0 sec

PathCompMethod : local-cspf      OperPathCompMethod: local-cspf
MetricType    : igp              Oper MetricType  : igp
LocalSrProt   : preferred        Oper LocalSrProt: preferred
LabelStackRed: Enabled          Oper LabelStackRed: Enabled

Bandwidth     : No Reservation     Oper Bandwidth  : 0 Mbps
Hop Limit     : 255                Oper HopLimit   : 255
Setup Priority : 7                  Oper SetupPriority: 7
Hold Priority  : 0                  Oper HoldPriority: 0
Inter-area    : N/A

---snip---

IGP/TE Metric : 220                Oper Metric     : 220
Oper MTU      : 8970                Path Trans      : 3
Failure Code  : noError
Failure Node  : n/a
Explicit Hops :
              : 192.0.2.4(S)
              : -> 192.0.2.3(S)
Actual Hops   :

```

```

192.0.2.4(192.0.2.4) (N-SID)           Record Label      : 32004
-> 192.0.2.3(192.0.2.3) (N-SID)      Record Label      : 32003
-> 192.0.2.11(192.0.2.11) (N-SID)    Record Label      : 32011

---snip---

=====

```

SR-TE LSPs using path with loose hops

The following SR-TE LSPs on PE-2 toward PE-11 use a path with loose hops P-3 and P-9:

```

# on PE-2:
configure {
  router "Base" {
    mpls {
      path "path_via-P-3-P-9_L" {
        admin-state enable
        hop 10 {
          ip-address 192.0.2.3
          type loose
        }
        hop 20 {
          ip-address 192.0.2.9
          type loose
        }
      }
      lsp "LSP-PE-2-PE-11_loose-hops_hop-to-label" {
        admin-state enable
        type p2p-sr-te
        to 192.0.2.11
        max-sr-labels {
          additional-frr-labels 2
        }
        primary "path_via-P-3-P-9_L" {
        }
      }
      lsp "LSP-PE-2-PE-11_loose-hops_local-cspf" {
        admin-state enable
        type p2p-sr-te
        to 192.0.2.11
        path-computation-method local-cspf
        max-sr-labels {
          additional-frr-labels 2
        }
        primary "path_via-P-3-P-9_L" {
        }
      }
    }
  }
}

```

With hop-to-label path calculation, loose hops are translated into N-SIDs. In this example, the actual hops are the N-SIDs of P-3, P-9, and PE-11, as follows:

```

[/]
A:admin@PE-2# show router mpls sr-te-lsp "LSP-PE-2-PE-11_loose-hops_hop-to-label" path detail

=====
MPLS SR-TE LSP LSP-PE-2-PE-11_loose-hops_hop-to-label
Path (Detail)
=====
Legend :

```



```

S      - Strict
A-SID  - Adjacency SID
+      - Inherited

L      - Loose
N-SID  - Node SID
=====
-----
LSP SR-TE LSP-PE-2-PE-11_loose-hops_hop-to-label
Path path_via-P-3-P-9_L
-----
LSP Name      : LSP-PE-2-PE-11_loose-hops_hop-to-label
Path LSP ID   : 41472
From          : 192.0.2.2
To            : 192.0.2.11
Admin State   : Up
Oper State    : Up
Path Name     : path_via-P-3-P-9_L
Path Type     : Primary
Path Admin    : Up
Path Oper     : Up
Path Up Time  : 0d 00:03:36
Path Down Time : 0d 00:00:00
Retry Limit   : 0
Retry Timer   : 30 sec
Retry Attempt : 0
Next Retry In : 0 sec

PathCompMethod : none
MetricType     : igp
LocalSrProt    : preferred
LabelStackRed  : Disabled

OperPathCompMethod: none
Oper MetricType : igp
Oper LocalSrProt : N/A
Oper LabelStackRed: N/A

Bandwidth      : No Reservation
Hop Limit      : 255
Setup Priority  : 7
Hold Priority   : 0
Inter-area     : N/A

Oper Bandwidth : 0 Mbps
Oper HopLimit  : 255
Oper SetupPriority: 7
Oper HoldPriority : 0

---snip---

IGP/TE Metric : 16777215
Oper MTU       : 8970
Oper Metric    : 16777215
Path Trans     : 1
Failure Code   : noError
Failure Node   : n/a
Explicit Hops  :
                192.0.2.3(L)
                -> 192.0.2.9(L)
Actual Hops    :
                192.0.2.3(192.0.2.3) (N-SID)
                -> 192.0.2.9(192.0.2.9) (N-SID)
                -> 192.0.2.11(192.0.2.11) (N-SID)
                Record Label : 32003
                Record Label : 32009
                Record Label : 32011

---snip---
=====

```

With local CSPF path calculation, the actual hops are the A-SIDs toward P-4, P-3, P-7, P-9, and PE-11, as follows:

```

[/]
A:admin@PE-2# show router mpls sr-te-lsp "LSP-PE-2-PE-11_loose-hops_local-cspf" path detail
=====
MPLS SR-TE LSP LSP-PE-2-PE-11_loose-hops_local-cspf
Path (Detail)
=====
Legend :
S      - Strict
A-SID  - Adjacency SID
+      - Inherited

L      - Loose
N-SID  - Node SID

```

```

=====
-----
LSP SR-TE LSP-PE-2-PE-11_loose-hops_local-cspf
Path path_via-P-3-P-9_L
-----
LSP Name      : LSP-PE-2-PE-11_loose-hops_local-cspf
Path LSP ID   : 59392
From          : 192.0.2.2
To            : 192.0.2.11
Admin State   : Up                               Oper State    : Up
Path Name     : path_via-P-3-P-9_L
Path Type     : Primary
Path Admin    : Up                               Path Oper     : Up
Path Up Time  : 0d 00:03:36                       Path Down Time : 0d 00:00:00
Retry Limit   : 0                                 Retry Timer    : 30 sec
Retry Attempt : 0                                 Next Retry In  : 0 sec

PathCompMethod : local-cspf                    OperPathCompMethod: local-cspf
MetricType     : igp                               Oper MetricType : igp
LocalSrProt    : preferred                         Oper LocalSrProt : preferred
LabelStackRed  : Disabled                         Oper LabelStackRed: Disabled

Bandwidth      : No Reservation                    Oper Bandwidth  : 0 Mbps
Hop Limit      : 255                              Oper HopLimit   : 255
Setup Priority  : 7                               Oper SetupPriority: 7
Hold Priority   : 0                               Oper HoldPriority : 0
Inter-area     : N/A

---snip---

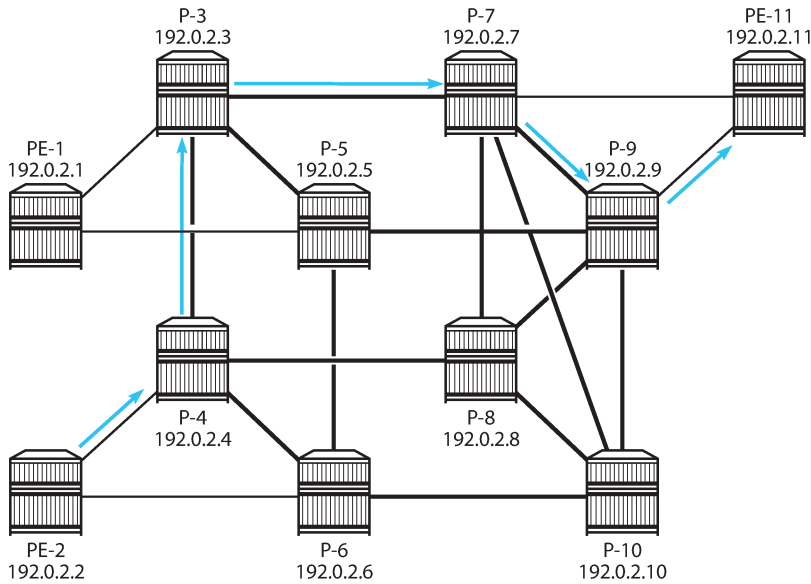
IGP/TE Metric  : 230                              Oper Metric     : 230
Oper MTU       : 8962                              Path Trans      : 1
Failure Code   : noError
Failure Node   : n/a
Explicit Hops  :
                192.0.2.3(L)
                -> 192.0.2.9(L)
Actual Hops    :
    192.168.24.2(192.0.2.4) (A-SID)                Record Label   : 524287
-> 192.168.34.1(192.0.2.3) (A-SID)                Record Label   : 524286
-> 192.168.37.2(192.0.2.7) (A-SID)                Record Label   : 524284
-> 192.168.79.2(192.0.2.9) (A-SID)                Record Label   : 524285
-> 192.168.119.2(192.0.2.11) (A-SID)              Record Label   : 524283

---snip---
=====

```

Figure 369: Path from PE-2 to PE-11 via loose hops P-3 and P-9 shows the path from PE-2 to PE-11 via loose hops P-3 and P-9.

Figure 369: Path from PE-2 to PE-11 via loose hops P-3 and P-9



35623

Label stack reduction is configured as follows:

```
# on PE-2:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-2-PE-11_loose-hops_local-cspf" {
        admin-state enable
        type p2p-sr-te
        to 192.0.2.11
        path-computation-method local-cspf
        label-stack-reduction true
        max-sr-labels {
          additional-frr-labels 2
        }
        primary "path_via-P-3-P-9_L" {
        }
      }
    }
  }
}
```

With label stack reduction, the actual hops in the path are the following:

```
[/]
A:admin@PE-2# show router mpls sr-te-lsp "LSP-PE-2-PE-11_loose-hops_local-cspf" path detail

=====
MPLS SR-TE LSP LSP-PE-2-PE-11_loose-hops_local-cspf
Path (Detail)
=====
Legend :
  S      - Strict
  A-SID  - Adjacency SID
  +      - Inherited
  L      - Loose
  N-SID  - Node SID
=====
-----
LSP SR-TE LSP-PE-2-PE-11_loose-hops_local-cspf
```

```

Path path_via-P-3-P-9_L
-----
LSP Name      : LSP-PE-2-PE-11_loose-hops_local-cspf
Path LSP ID   : 59392
From          : 192.0.2.2
To            : 192.0.2.11
Admin State   : Up                               Oper State    : Up
Path Name     : path_via-P-3-P-9_L
Path Type     : Primary
Path Admin    : Up                               Path Oper     : Up
Path Up Time  : 0d 00:02:09                       Path Down Time : 0d 00:00:00
Retry Limit   : 0                                 Retry Timer    : 30 sec
Retry Attempt : 0                                 Next Retry In  : 0 sec

PathCompMethod : local-cspf                    OperPathCompMethod: local-cspf
MetricType    : igp                               Oper MetricType : igp
LocalSrProt   : preferred                         Oper LocalSrProt : preferred
LabelStackRed : Enabled                       Oper LabelStackRed: Enabled

Bandwidth     : No Reservation                    Oper Bandwidth  : 0 Mbps
Hop Limit     : 255                               Oper HopLimit   : 255
Setup Priority : 7                               Oper SetupPriority: 7
Hold Priority  : 0                               Oper HoldPriority: 0
Inter-area    : N/A

---snip---

IGP/TE Metric : 230                               Oper Metric     : 230
Oper MTU      : 8970                               Path Trans      : 3
Failure Code  : noError
Failure Node  : n/a
Explicit Hops :
                192.0.2.3(L)
                -> 192.0.2.9(L)
Actual Hops   :
    192.0.2.3(192.0.2.3) (N-SID)                    Record Label   : 32003
-> 192.0.2.9(192.0.2.9) (N-SID)                    Record Label   : 32009
-> 192.0.2.11(192.0.2.11) (N-SID)                  Record Label   : 32011

---snip---
=====

```

Tunnel tables

The following command on PE-2 lists the SR-TE tunnels. By default, SR-TE tunnels have preference 8. The first three SR-TE LSP tunnels used local CSPF path computation without label stack reduction, while the latter three used hop-to-label path computation. For all SR-TE LSPs with next-hop 192.0.168.24.2, the first hop is mapped to an adjacency SID. All paths computed with local CSPF without label stack reduction only have adjacency SIDs or adjacency set SIDs. For hop-to-label path computation, only the strict hops are translated into adjacency SIDs.

```

[/]
A:admin@PE-2# show router tunnel-table protocol sr-te

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId Pref  Nexthop      Metric
Color

```

```
-----
192.0.2.11/32      sr-te    MPLS  655363  8    192.168.26.2  220
192.0.2.11/32      sr-te    MPLS  655367  8    192.168.24.2   220
192.0.2.11/32      sr-te    MPLS  655365  8    192.168.24.2   230
192.0.2.11/32      sr-te    MPLS  655362  8    192.0.2.11     16777215
192.0.2.11/32      sr-te    MPLS  655364  8    192.0.2.3      16777215
192.0.2.11/32      sr-te    MPLS  655366  8    192.168.24.2   16777215
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
```

For the hop-to-label computed paths, the value of the metric is set to 16777215 (infinity – 1), because CSPF is not used and the head-end router is unaware of the full topology between head- and tail-end router. For the paths computed with local CSPF, the IGP metrics are added; for example, for the first tunnel: 100 (PE-2 to P-6) + 10 (P-6 to P-10) + 10 (P-10 to P-7) + 100 (P-7 to PE-11) = 220.

When label stack reduction is configured, the next hops may be slightly different when the first hop—after label stack reduction—is mapped to a node SID, as follows:

```
[/]
A:admin@PE-2# show router tunnel-table protocol sr-te

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
192.0.2.11/32    sr-te     MPLS  655363   8    192.0.2.11   220
192.0.2.11/32    sr-te     MPLS  655367   8    192.0.2.4   220
192.0.2.11/32    sr-te     MPLS  655365   8    192.0.2.3   230
192.0.2.11/32    sr-te     MPLS  655362   8    192.0.2.11   16777215
192.0.2.11/32    sr-te     MPLS  655364   8    192.0.2.3    16777215
192.0.2.11/32    sr-te     MPLS  655366   8    192.168.24.2 16777215
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      E = Inactive best-external BGP route
      k = RIB-API or Forwarding Policy backup hop
=====
```

The following command on PE-2 shows the FP tunnel table for SR-TE LSP tunnels without label stack reduction. The first three SR-TE LSP tunnels (with tunnel IDs 655362, 655363, and 655364) have hop-to-label path computation and the latter three have local CSPF path computation. For hop-to-label path computation, A-SIDs are used for strict hops and N-SIDs are used for loose hops. For local CSPF path computation without label stack reduction, only A-SIDs and adjacency set SIDs are used.

```
[/]
A:admin@PE-2# show router fp-tunnel-table 1

=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination      Protocol      Tunnel-ID
-----
```

Lbl NextHop Lbl (backup) NextHop (backup)		Intf/Tunnel
192.0.2.1/32 32001 192.168.24.2	SR-ISIS-0	524290 1/1/1:1000
192.0.2.3/32 32003 192.168.24.2	SR-ISIS-0	524291 1/1/1:1000
192.0.2.4/32 32004 192.168.24.2	SR-ISIS-0	524292 1/1/1:1000
192.0.2.5/32 32005 192.168.26.2	SR-ISIS-0	524293 1/1/2:1000
192.0.2.6/32 32006 192.168.26.2	SR-ISIS-0	524295 1/1/2:1000
192.0.2.7/32 32007 192.168.24.2	SR-ISIS-0	524296 1/1/1:1000
192.0.2.8/32 32008 192.168.24.2	SR-ISIS-0	524297 1/1/1:1000
192.0.2.9/32 32009 192.168.24.2	SR-ISIS-0	524298 1/1/1:1000
192.0.2.10/32 32010 192.168.26.2	SR-ISIS-0	524299 1/1/2:1000
192.0.2.11/32 32011 192.168.24.2	SR-ISIS-0	524300
192.0.2.11/32 3 192.0.2.11	SR-TE	655362 SR
192.0.2.11/32 524283/524284/524284 192.168.26.2	SR-TE	655363 SR
192.0.2.11/32 32011/32009 192.0.2.3	SR-TE	655364 SR
192.0.2.11/32 524283/524285/524284/524286 192.168.24.2	SR-TE	655365 SR
192.0.2.11/32 32011/524286 192.168.24.2	SR-TE	655366 SR
192.0.2.11/32 524283/524284/524286 192.168.24.2	SR-TE	655367 SR
192.0.2.12/32 32012 192.168.24.2	SR-ISIS-0	524301 1/1/1:1000
192.168.24.2/32 3 192.168.24.2	SR	524289 1/1/1:1000
192.168.26.2/32 3 192.168.26.2	SR	524294 1/1/1:1000
		1/1/2:1000

Total Entries : 19		

The following command on PE-2 shows only the FP tunnel table for local CSPF path computed SR-TE LSP tunnels with label stack reduction, where N-SIDs replace sequences of A-SIDs.

```
[/]
A:admin@PE-2# show router fp-tunnel-table 1
=====
IPv4 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol      Tunnel-ID
Lbl                                         Intf/Tunnel
  NextHop
  Lbl      (backup)
  NextHop  (backup)
-----
---snip---
192.0.2.11/32                               SR-ISIS-0    524300
 32011
  192.168.24.2                               1/1/1:1000
192.0.2.11/32                               SR-TE        655362
 3
  192.0.2.11                               SR
192.0.2.11/32                             SR-TE      655363
 3
  192.0.2.11                               SR
192.0.2.11/32                               SR-TE        655364
 32011/32009
  192.0.2.3                               SR
192.0.2.11/32                             SR-TE      655365
32011/32009
  192.0.2.3                               SR
192.0.2.11/32                               SR-TE        655366
 32011/524286
  192.168.24.2                               SR
192.0.2.11/32                             SR-TE      655367
32011/32003
  192.0.2.4                               SR
---snip---
-----
Total Entries : 19
=====
```

Resignaling an SR-TE LSP

Point-to-point SR-TE LSPs have a resignal timer to match that of RSVP. It must be set to allow manual and automatic resignaling for optimization of SR-TE LSPs. The following command can be used to set a resignal timer in minutes for all originating SR-TE LSPs:

```
[ex:/configure router "Base" mpls sr-te-resignal]
A:admin@PE-2# resignal-timer ?
```

```
resignal-timer <number>
<number> - <30..10080> - minutes

Resignal timer for SR-TE LSPs
```

The following tools command can be launched for manually-triggered re-optimization of LSPs:

```
[/]
A:admin@PE-2# tools perform router mpls resignal

resignal p2mp-lsp <string> p2mp-instance <string>
resignal p2mp-delay <number>
resignal lsp <string> path <string>
resignal delay <number>
resignal sr-te-lsp <string> path <string>
resignal sr-te-delay <number>

delay - <number> - <0..30>
lsp - string '<1..64 characters>'
p2mp-delay - <number> - <0..60>
p2mp-instance - [Max 32 chars]
p2mp-lsp - string '<1..64 characters>'
path - string '<0..255 characters>'
sr-te-delay - <number> - <0..30>
sr-te-lsp - string '<1..64 characters>'
```

A manual re-optimization for a specific path in a specific SR-TE LSP can be forced, as follows:

```
[/]
A:admin@PE-2# tools perform router mpls resignal sr-te-lsp "LSP-PE-2-PE-11_loose-hops_local-
cspf" path "path_via-P-3-P-9_L"
```

The **sr-te-delay** parameter overrides the global resignal timer value for all SR-TE LSPs. When this timer expires, the procedures of the timer-based resignal are applied to all SR-TE LSPs and the SR-TE resignal time is then reset to its configured value in the MPLS configuration.

The following command forces a re-optimization of the SR-TE LSPs after an **sr-te-delay** of 3 minutes, but this CLI delay will only be in effect when the **sr-te-resignal-timer** is configured in the **mpls** context. If not, the following error is raised:

```
[/]
A:admin@PE-2# tools perform router mpls resignal sr-te-delay 3
WARNING: CLI #2006: Warning while processing command - WARNING: CLI Delay will not be in
effect, configure resignal-timer under config>router>mpls>sr-te-resignal.
```

The SR-TE resignal timer is configured in the **mpls** context with a value of 60 minutes, as follows:

```
# on PE-2:
configure {
  router "Base" {
    mpls {
      sr-te-resignal {
        resignal-timer 60
      }
    }
  }
}
```

With the SR-TE resignal timer configured, the tools command can be launched to override this SR-TE resignal timer to a value of 3 minutes, as follows:

```
[/]
```



```
A:admin@PE-2# tools perform router mpls resignal sr-te-delay 3
```

Local CSPF and SR protected adjacencies

The following command enables TI-LFA with link protection on all nodes:

```
# on all nodes:
configure {
  router "Base" {
    isis 0 {
      loopfree-alternate {
        ti-lfa {
        }
      }
    }
  }
}
```

As a result of this, each adjacency is available for SID protection. For example, on P-4, all adjacency SID sub-TLVs have the B-flag set, as follows:

```
[/]
A:admin@P-4# show router isis database P-4 detail level 2 | match "Adj-SID" pre-lines 1
  Nbr IP      : 192.168.24.1
  Adj-SID: Flags:v4BVL Weight:0 Label:524287
  Nbr IP      : 192.168.34.1
  Adj-SID: Flags:v4BVL Weight:0 Label:524286
  Nbr IP      : 192.168.46.2
  Adj-SID: Flags:v4BVL Weight:0 Label:524285
  Nbr IP      : 192.168.48.2
  Adj-SID: Flags:v4BVL Weight:0 Label:524284
  E = Entropy Label Capability (ELC) Flag
Adj-SID Flags : v4/v6 = IPv4 or IPv6 Address-Family
  B = Backup Flag
  V = Adj-SID carries a value
```

The following command removes SID protection from the interface toward P-8:

```
# on P-4:
configure {
  router "Base" {
    isis 0 {
      interface "int-P-4-P-8" {
        sid-protection false
      }
    }
  }
}
```

The adjacency SID sub-TLV for this link does not have the B-flag set, as follows:

```
[/]
A:admin@P-4# show router isis database P-4 detail level 2

=====
Rtr Base ISIS Instance 0 Database (detail)
=====

Displaying Level 2 database
-----
LSP ID      : P-4.00-00
Sequence    : 0x19
Version     : 1
Attributes: L1L2
Checksum    : 0x559a
Pkt Type    : 20
Max Area    : 3
Level       : L2
Lifetime    : 1188
Pkt Ver     : 1
Alloc Len   : 1492
```

```

SYS ID      : 1920.0000.2004          SysID Len : 6          Used Len  : 645

TLVs :
  Area Addresses:
    Area Address : (3) 49.0001
  Supp Protocols:
    Protocols    : IPv4
  IS-Hostname   : P-4
  Router ID     :
    Router ID    : 192.0.2.4
  Router Cap    : 192.0.2.4, D:0, S:0
    TE Node Cap  : B E M P
    SR Cap       : IPv4 MPLS-IPv6
    SRGB Base    : 32000, Range:1000
    SR Alg       : metric based SPF
    Node MSD Cap: BMI : 12 ERLD : 15
  I/F Addresses :
    I/F Address  : 192.168.46.1
    I/F Address  : 192.168.48.1
    I/F Address  : 192.168.24.2
    I/F Address  : 192.168.34.2
    I/F Address  : 192.0.2.4
  ---snip---
  TE IS Nbrs   :
    Nbr         : P-8.00
    Default Metric : 10
    Sub TLV Len  : 19
    IF Addr     : 192.168.48.1
    Nbr IP      : 192.168.48.2
    Adj-SID: Flags:v4VL Weight:0 Label:524284
  ---snip---

Level (2) LSP Count : 1
-----
---snip---
Adj-SID Flags      : v4/v6 = IPv4 or IPv6 Address-Family
                    B = Backup Flag
                    V = Adj-SID carries a value
                    L = value/index has local significance
                    S = Set of Adjacencies
                    P = Persistently allocated
  ---snip---
=====

```

Local CSPF will, by default, compute an end-to-end path by selecting protected adjacencies, which have the B-flag set, as previously described. If no such path is available, the local CSPF may select an unprotected adjacency with the assumption that all other path constraints are met.

The following tools command calculates the path from P-4 to P-8 without establishing it. With the **preferred** option, protected adjacencies are preferred over unprotected adjacency when both exist for a TE link. In this example, the shortest path contains the direct link to P-8, which does not have a backup:

```

[/]
A:admin@P-4# tools perform router mpls sr-te-cspf to 192.0.2.8 path-computation-method local-
cspf local-sr-protection preferred
Req CSPF TE path
  From: this node To: 192.0.2.8
CSPF TE Path
To: 192.0.2.8
[1] Source Add 192.0.2.4      Cost 10
  Hop 1 -> Label 524284 NH 192.168.48.1 --> 192.168.48.2 (192.0.2.8) Cost 10 Color 0x0

```

The following tools command calculates the path from P-4 to P-8 with the restriction that each link in the path is unprotected. The shortest path is the unprotected link to P-8:

```
[/]
A:admin@P-4# tools perform router mpls sr-te-cspf to 192.0.2.8 path-computation-method local-
cspf local-sr-protection none
Req CSPF TE path
  From: this node To: 192.0.2.8
CSPF TE Path
  To: 192.0.2.8
  [1] Source Add 192.0.2.4    Cost 10
      Hop 1 -> Label 524284 NH 192.168.48.1 --> 192.168.48.2 (192.0.2.8) Cost 10 Color 0x0
```

The following tools command calculates the path from P-4 to P-8 but all adjacencies must be protected, so the unprotected direct link to P-8 is excluded and the path goes via P-6 and P-10 to P-8:

```
[/]
A:admin@P-4# tools perform router mpls sr-te-cspf to 192.0.2.8 path-computation-method local-
cspf local-sr-protection mandatory
Req CSPF TE path
  From: this node To: 192.0.2.8
CSPF TE Path
  To: 192.0.2.8
  [1] Source Add 192.0.2.4    Cost 30
      Hop 1 -> Label 524285 NH 192.168.46.1 --> 192.168.46.2 (192.0.2.6) Cost 10 Color 0x0
      Hop 2 -> Label 524284 NH 192.168.106.1 --> 192.168.106.2 (192.0.2.10) Cost 10 Color 0x0
      Hop 3 -> Label 524287 NH 192.168.108.2 --> 192.168.108.1 (192.0.2.8) Cost 10 Color 0x0
```

On PE-2, the following SR-TE LSPs with local CSPF path computation using a loose path are configured toward PE-11:

- an SR-TE LSP with local SR protection preferred (= default setting)
- an SR-TE LSP with mandatory local SR protection, where all adjacencies must have a backup
- an SR-TE LSP without local SR protection (none), where all adjacencies are unprotected

```
# on PE-2:
configure {
  router "Base" {
    mpls {
      lsp "LSP-PE-2-PE-11_empty_path_local-cspf_protection-preferred" {
        admin-state enable
        type p2p-sr-te
        to 192.0.2.11
        path-computation-method local-cspf
        local-sr-protection preferred          # default
        max-sr-labels {
          additional-frr-labels 2
        }
        primary "empty_path" {
        }
      }
      lsp "LSP-PE-2-PE-11_empty_path_local-cspf_protection-mandatory" {
        admin-state enable
        type p2p-sr-te
        to 192.0.2.11
        path-computation-method local-cspf
        local-sr-protection mandatory        # all links in E2E path protected
        max-sr-labels {
          additional-frr-labels 2
        }
      }
    }
  }
}
```

```

        primary "empty_path" {
        }
    }
    lsp "LSP-PE-2-PE-11_empty_path_local-cspf_protection-none" {
    admin-state enable
    type p2p-sr-te
    to 192.0.2.11
    path-computation-method local-cspf
    local-sr-protection none          # all links in E2E path unprotected
    max-sr-labels {
        additional-frr-labels 2
    }
    primary "empty_path" {
    }
    }
}

```

For test purposes, the metric on the unprotected interface between P-4 and P-8 is lowered to 5, so the shortest path to PE-11 will include the unprotected interface when allowed:

```

# on P-4:
configure {
    router "Base" {
        isis 0 {
            interface "int-P-4-P-8" {
                interface-type point-to-point
                sid-protection false
                level 1 {
                    metric 5
                }
                level 2 {
                    metric 5
                }
            }
        }
    }
}

```

For SR-TE LSP "LSP-PE-2-PE-11_empty_path_local-cspf_protection-preferred", a path can be established from PE-2 via P-4, P-8, and P-7 to PE-11, but for the same metric, P-7 can be replaced by P-9. The direct link between P-4 and P-8 is unprotected, while all other adjacencies in the path have a backup.

```

[/]
A:admin@PE-2# show router mpls sr-te-lsp "LSP-PE-2-PE-11_empty_path_local-cspf_protection-
preferred" path detail

=====
MPLS SR-TE LSP LSP-PE-2-PE-11_empty_path_local-cspf_protection-preferred
Path (Detail)
=====
Legend :
  S      - Strict          L      - Loose
  A-SID  - Adjacency SID  N-SID - Node SID
  +      - Inherited

-----
LSP SR-TE LSP-PE-2-PE-11_empty_path_local-cspf_protection-preferred
Path empty_path
-----
LSP Name      : LSP-PE-2-PE-11_empty_path_local-cspf_protection-preferred
Path LSP ID   : 37888
From          : 192.0.2.2
To            : 192.0.2.11
Admin State   : Up                Oper State    : Up
Path Name     : empty_path
Path Type     : Primary

```

```

Path Admin      : Up
Path Up Time    : 0d 00:02:21
Retry Limit     : 0
Retry Attempt   : 0

PathCompMethod : local-cspf
MetricType     : igp
LocalSrProt    : preferred
LabelStackRed  : Disabled

Bandwidth      : No Reservation
Hop Limit      : 255
Setup Priority  : 7
Hold Priority   : 0
Inter-area     : N/A

---snip---

IGP/TE Metric  : 215
Oper MTU       : 8958
Failure Code   : noError
Failure Node   : n/a
Explicit Hops  :
  No Hops Specified
Actual Hops    :
  192.168.24.2(192.0.2.4) (A-SID)
  -> 192.168.48.2(192.0.2.8) (A-SID)
  -> 192.168.78.1(192.0.2.7) (A-SID)
  -> 192.168.117.2(192.0.2.11) (A-SID)

Record Label   : 524287
Record Label  : 524284
Record Label   : 524287
Record Label   : 524283

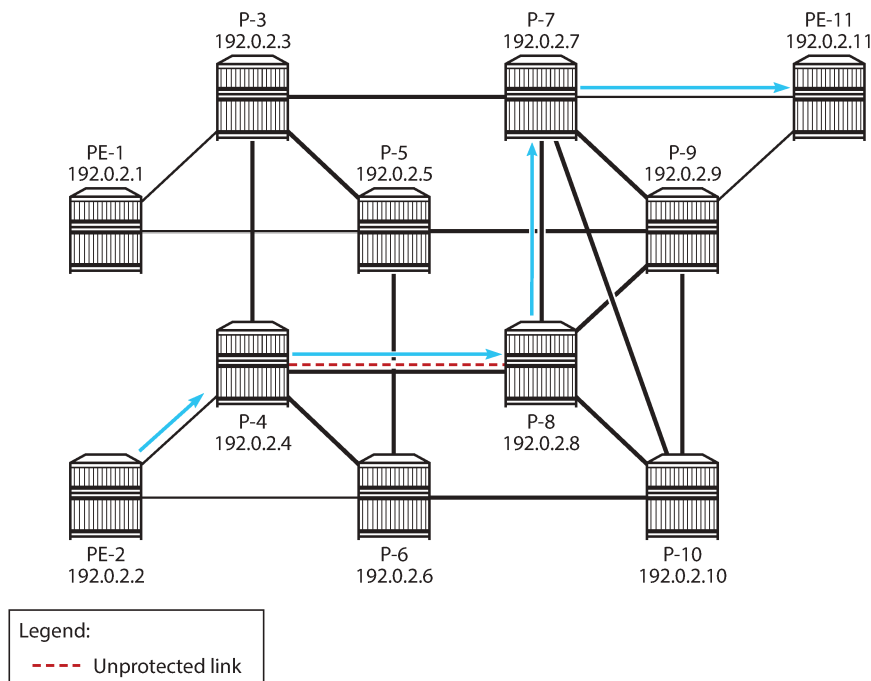
---snip---

=====

```

Figure 370: Loose path from PE-2 to PE-11 including unprotected link shows the established path from PE-2 to PE-11, which uses protected and unprotected links.

Figure 370: Loose path from PE-2 to PE-11 including unprotected link



35624

For SR-TE LSP "LSP-PE-2-PE-11_empty_path_local-cspf_protection-mandatory", the path must exclude the unprotected link from P-4 to P-8. The path goes from PE-2 via P-6, P-10, and P-9 to PE-11, but for the same metric, other paths are possible.

```
[/]
A:admin@PE-2# show router mpls sr-te-lsp "LSP-PE-2-PE-11_empty_path_local-cspf_protection-
mandatory" path detail

=====
MPLS SR-TE LSP LSP-PE-2-PE-11_empty_path_local-cspf_protection-mandatory
Path (Detail)
=====
Legend :
S      - Strict          L      - Loose
A-SID  - Adjacency SID  N-SID  - Node SID
+      - Inherited

-----
LSP SR-TE LSP-PE-2-PE-11_empty_path_local-cspf_protection-mandatory
Path empty_path
-----
LSP Name      : LSP-PE-2-PE-11_empty_path_local-cspf_protection-mandatory
Path LSP ID   : 4096
From          : 192.0.2.2
To           : 192.0.2.11
Admin State   : Up                Oper State    : Up
Path Name     : empty_path
Path Type     : Primary
Path Admin    : Up                Path Oper     : Up
Path Up Time  : 0d 00:02:21       Path Down Time : 0d 00:00:00
Retry Limit   : 0                 Retry Timer    : 30 sec
```

```

Retry Attempt      : 0                      Next Retry In     : 0 sec
PathCompMethod    : local-cspf              OperPathCompMethod: local-cspf
MetricType        : igp                    Oper MetricType   : igp
LocalSrProt      : mandatory              Oper LocalSrProt : mandatory
LabelStackRed     : Disabled                Oper LabelStackRed: N/A

Bandwidth         : No Reservation           Oper Bandwidth    : 0 Mbps
Hop Limit         : 255                     Oper HopLimit     : 255
Setup Priority    : 7                       Oper SetupPriority: 7
Hold Priority     : 0                       Oper HoldPriority : 0
Inter-area       : N/A

---snip---

IGP/TE Metric     : 220                     Oper Metric       : 220
Oper MTU          : 8958                    Path Trans       : 1
Failure Code      : noError
Failure Node      : n/a
Explicit Hops     :
    No Hops Specified
Actual Hops       :
    192.168.26.2(192.0.2.6) (A-SID)          Record Label     : 524286
-> 192.168.106.2(192.0.2.10) (A-SID)         Record Label     : 524284
-> 192.168.109.1(192.0.2.9) (A-SID)          Record Label     : 524285
-> 192.168.119.2(192.0.2.11) (A-SID)         Record Label     : 524283

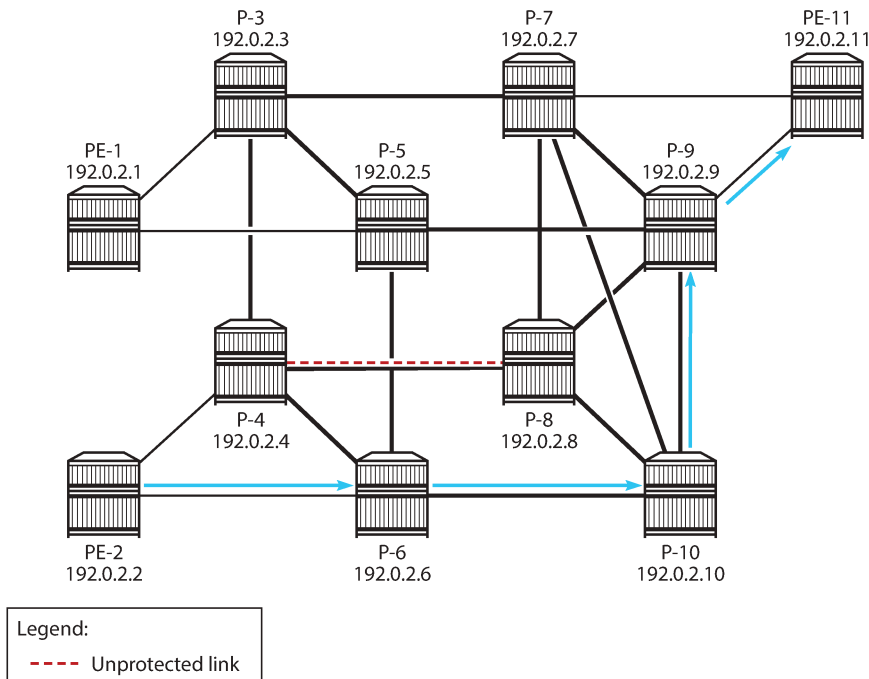
---snip---

=====

```

Figure 371: Loose path from PE-2 to PE-11 including only protected links shows the loose path that excludes the unprotected link between P-4 and P-8.

Figure 371: Loose path from PE-2 to PE-11 including only protected links



35625

For SR-TE LSP "LSP-PE-2-PE-11_empty_path_local-cspf_protection-none", the path should only use unprotected links, but only the adjacency between P-4 and P-8 is unprotected, so CSPF cannot find an end-to-end path. The path and the LSP remain operationally down with failure code noCspfRouteToDestination, as follows:

```
[/]
A:admin@PE-2# show router mpls sr-te-lsp "LSP-PE-2-PE-11_empty_path_local-cspf_protection-none"
path detail

=====
MPLS SR-TE LSP LSP-PE-2-PE-11_empty_path_local-cspf_protection-none
Path (Detail)
=====
Legend :
  S      - Strict          L      - Loose
  A-SID  - Adjacency SID   N-SID  - Node SID
  +      - Inherited
=====
LSP SR-TE LSP-PE-2-PE-11_empty_path_local-cspf_protection-none
Path empty_path
-----
LSP Name      : LSP-PE-2-PE-11_empty_path_local-cspf_protection-none
Path LSP ID   : 50176
From          : 192.0.2.2
To            : 192.0.2.11
Admin State   : Up          Oper State    : Down
Path Name     : empty_path
Path Type     : Primary
Path Admin    : Up          Path Oper     : Down
Path Up Time  : 0d 00:00:00 Path Down Time: 0d 00:02:21
```



```
Retry Limit      : 0                Retry Timer      : 30 sec
Retry Attempt    : 5                Next Retry In   : 19 sec

PathCompMethod  : local-cspf        OperPathCompMethod: N/A
MetricType      : igp              Oper MetricType  : N/A
LocalSrProt     : none             Oper LocalSrProt : N/A
LabelStackRed   : Disabled         Oper LabelStackRed: N/A

Bandwidth       : No Reservation    Oper Bandwidth   : N/A
Hop Limit       : 255              Oper HopLimit    : N/A
Setup Priority   : 7               Oper SetupPriority: N/A
Hold Priority    : 0               Oper HoldPriority : N/A
Inter-area      : N/A

---snip---

IGP/TE Metric   : N/A              Oper Metric      : N/A
Oper MTU        : N/A              Path Trans       : 0
Failure Code   : noCspfRouteToDestination
Failure Node    : 192.0.2.2
Explicit Hops   :
  No Hops Specified
Actual Hops     :
  No Hops Specified

---snip---
```

Conclusion

Within a single-level IS-IS network or a single-area OSPF network, SR-TE LSP path calculation using local CSPF on the head-end router results in an end-to-end path using IPv4 adjacencies. The local CSPF path computation method can also be used for RSVP-TE LSPs.

SRv6 Encapsulation in the Base Routing Instance

This chapter provides information about SRv6 encapsulation in the base routing instance.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information and configuration in this chapter are based on SR OS Release 22.2.R1. Segment Routing over IPv6 (SRv6) is supported on FP4-based equipment in SR OS Release 21.5.R2 and later.

Overview

SRv6 encapsulation in the base routing instance allows the transport of native IPv4 and IPv6 data across an SRv6-enabled network. To this end, native IPv4 and IPv6 data is sent to an ingress SRv6 router, where it is encapsulated and forwarded via an SRv6 tunnel. The SRv6 tunnel transports the encapsulated data across the SRv6-enabled network to an egress SRv6 router, where it is decapsulated and forwarded further as native IPv4 and IPv6 data. SRv6-tunneled data is encapsulated using an IPv6 header, where the destination address is a unique SRv6 segment identifier (SID), and is processed and forwarded in the IPv6 data plane.

An SRv6 SID is a preconfigured 128-bit routable IPv6 prefix address that is encoded in three parts: a locator, a function, and an argument. The locator is a summary IPv6 prefix for a set of SRv6 SIDs instantiated on an SRv6-capable router. It is used to route the data within the IPv6 transport network. Each participating SRv6-capable router needs its unique locator, based on a common block that all participating SRv6-capable routers share in the IPv6 address space. The function is an opaque identifier that indicates the local behavior at the endpoint of an SRv6 segment. The focus in this topic is on the SRv6 End.DT4 and the SRv6 End.DT6 functions, performing a prefix lookup in the global IPv4 route table (End.DT4) or in the global IPv6 route table (End.DT6). The argument is not used in SR OS 22.2.R1 and is set to all zeros.

The local router installs its locator prefix in its IPv6 route table and Forwarding Information Base (FIB), and advertises its locator prefix in IS-IS with the SRv6 locator sub-TLV. Each remote router populates its route table and FIB with the received locator prefixes, including the tunneled next hop to the originating router.

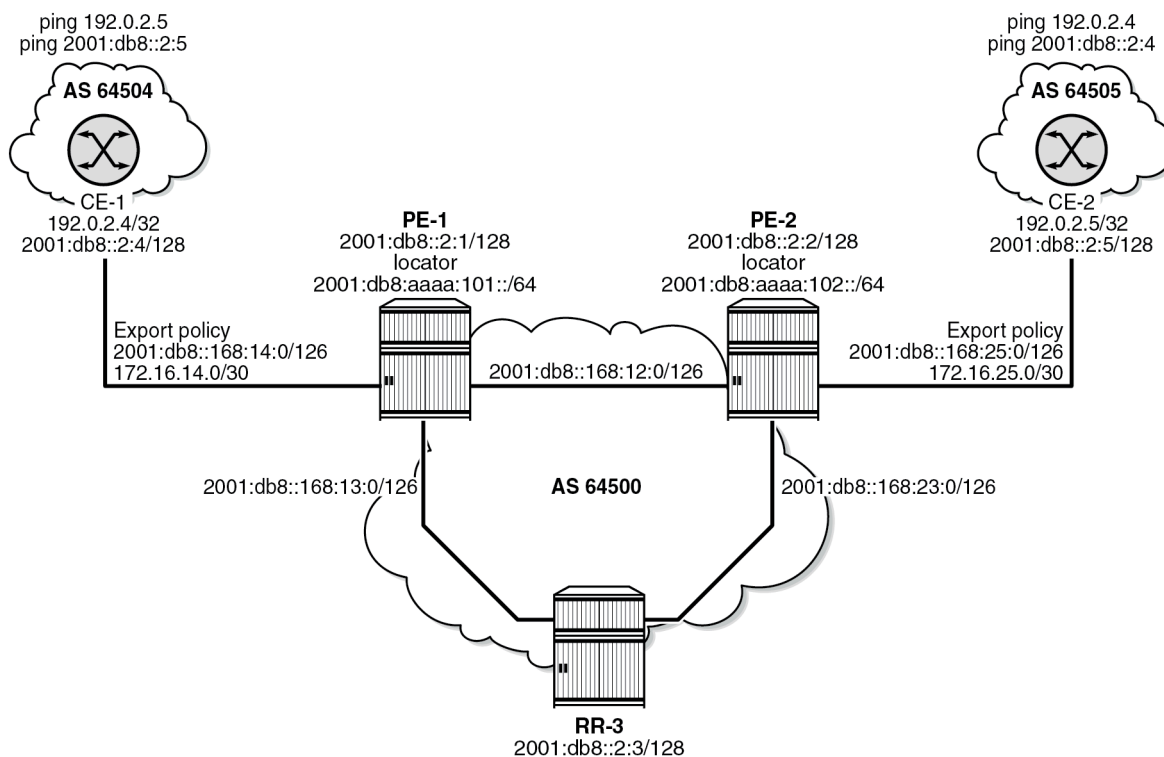
SRv6 data transport requires additional processing at both the ingress and egress data planes. This processing relies on Forwarding Path Extension (FPE), as described in the [Segment Routing over IPv6](#) chapter.

Configuration

Figure 372: Example topology shows the example topology with five routers. Two routers (CE-1 and CE-2) simulate an IPv6-enabled network. They are connected to an SRv6-enabled network, comprising of PE-1

and PE-2, and a route reflector (RR) RR-3 in the control plane. The SRv6-enabled network has only IPv6 addresses and interfaces.

Figure 372: Example topology



For the transport of native IPv4 and IPv6 data from CE-1 to CE-2, PE-1 acts as the SRv6 ingress PE node, while PE-2 acts as the SRv6 egress PE node. For the transport of native IPv4 and IPv6 data from CE-2 to CE-1, PE-2 acts as the SRv6 ingress PE node, while PE-1 acts as the SRv6 egress PE node. To explain the SRv6 encapsulation concept, the topology does not need an SRv6 transit router because SRv6 transit routers simply forward SRv6-encapsulated packets via IPv6 route table lookup without any other processing.

SRv6 and FPE are configured only on PE-1 and on PE-2. RR-3 acts as the BGP RR in the control plane and does not participate in the SRv6 data transport that only exists between PE-1 and PE-2.

The **ping** and **traceroute** commands between IPv4 and IPv6 system addresses of CE-1 and CE-2 simulate data transport.

The configuration for this example topology is completely symmetrical. All **configure** and **show** command outputs for PE-1 also apply for PE-2, and similar for CE-1 and CE-2.

The following sections describe the configuration steps needed to establish SRv6 Encapsulation in the base routing instance.

Configure the transport network:

This configuration includes:

- ports and IPv6-only interfaces on PE-1, PE-2, and RR-3
- port cross connect (PXC) and FPE on PE-1 and PE-2 (using internal loopbacks on an FP4 MAC chip), as described in the [Segment Routing over IPv6](#) chapter .
- IS-IS on PE-1, PE-2, and RR-3 including:
 - level 2 capability with wide metrics (for the 128-bit identifiers)
 - native IPv6 routing
 - as best practice, include also: **traffic-engineering** and **traffic-engineering-options** on PE-1 and PE-2
 - advertise the router capability within the autonomous system (AS) (not for RR-3)
- BGP on PE-1, PE-2, and RR-3, with internal group “gr_v6_internal” that includes:
 - IPv4 and IPv6 address families
 - **extended-nh-encoding** for IPv4
 - **advertise-ipv6-next-hops** for IPv4
 - **next-hop-self** (not for RR-3)
 - BGP neighbor **system** IPv6 addresses

The core network topology uses IPv6 for BGP peering (with 16-byte next hop addresses), so to advertise and receive IPv4 routes (which have 4-byte next hop addresses) with IPv6 next hop addresses the commands **advertise-ipv6-next-hops** and **extended-nh-encoding** need to be configured at the BGP, group, or neighbor level. The **advertise-ipv6-next-hops** command instructs the system to advertise IPv4 routes with an IPv6 next hop address. The **extended-nh-encoding** command configures BGP to advertise the capability to receive IPv4 routes with an IPv6 next hop address.

The following example configuration applies for PE-1 and is similar for PE-2.

The following example configuration applies for RR-3:

```
[/]  
A:admin@PE-1# configure {  
  router "Base" {  
    autonomous-system 64500  
    interface "int-PE-1-PE-2" {  
      description "interface between PE-1 and PE-2"  
      port 1/1/c1/1:1000  
      ipv6 {  
        address 2001:db8::168:12:1 {  
          prefix-length 126  
        }  
      }  
    }  
    interface "int-PE-1-RR-3" {  
      description "interface between PE-1 and RR-3"  
      port 1/1/c2/1:1000  
      ipv6 {  
        address 2001:db8::168:13:1 {  
          prefix-length 126  
        }  
      }  
    }  
  }  
  interface "system" {  
    description "system interface of PE-1"  
    ipv6 {  
      address 2001:db8::2:1 {
```

```

        prefix-length 128
    }
}
isis 0 {
    admin-state enable
    advertise-router-capability as
    ipv6-routing native
    level-capability 2 # required for SRV6
    router-id 1.1.1.1
    traffic-engineering true
    area-address [49.0001]
    traffic-engineering-options {
        ipv6 true
        application-link-attributes {
        }
    }
    interface "int-PE-1-PE-2" {
        interface-type point-to-point
    }
    interface "int-PE-1-RR-3" {
        interface-type point-to-point
    }
    interface "system" {
        passive true
    }
    level 2 {
        wide-metrics-only true # required for SRV6
    }
}
bgp {
    min-route-advertisement 1
    router-id 2.2.2.1
    rapid-withdrawal true
    split-horizon true
    ebgp-default-reject-policy {
        import false # do not refuse eBGP imported policies
        export false # do not prevent eBGP exported policies
    }
    group "gr_v6_internal" {
        description "internal bgp group on PE-1"
        next-hop-self true
        type internal
        family {
            ipv4 true
            ipv6 true
        }
        extended-nh-encoding {
            ipv4 true
        }
        advertise-ipv6-next-hops {
            ipv4 true
        }
    }
    neighbor "2001:db8::2:3" { # RR-3 system address
        group "gr_v6_internal"
    }
}
exit all

```

The following example configuration applies for RR-3:

```
[/]
```

```
A:admin@RR-3# configure {
  router "Base" {
    autonomous-system 64500
    interface "int-RR-3-PE-1" {
      description "interface between RR-3 and PE-1"
      port 1/1/c1/1:1000
      ipv6 {
        address 2001:db8::168:13:2 {
          prefix-length 126
        }
      }
    }
    interface "int-RR-3-PE-2" {
      description "interface between RR-3 and PE-2"
      port 1/1/c2/1:1000
      ipv6 {
        address 2001:db8::168:23:2 {
          prefix-length 126
        }
      }
    }
    interface "system" {
      description "system interface of RR-3"
      ipv6 {
        address 2001:db8::2:3 {
          prefix-length 128
        }
      }
    }
    isis 0 {
      admin-state enable
      ipv6-routing native
      level-capability 2 # required for SRv6
      router-id 1.1.1.3
      area-address [49.0001]
      interface "int-RR-3-PE-1" {
        interface-type point-to-point
      }
      interface "int-RR-3-PE-2" {
        interface-type point-to-point
      }
      interface "system" {
        passive true
      }
      level 2 {
        wide-metrics-only true # required for SRv6
      }
    }
    bgp {
      min-route-advertisement 1
      router-id 2.2.2.3
      rapid-withdrawal true
      split-horizon true
      group "gr_v6_internal" {
        description "internal bgp group on RR-3"
        type internal
        family {
          ipv4 true
          ipv6 true
        }
        cluster {
          cluster-id 3.3.3.3
        }
        extended-nh-encoding {

```

```

        ipv4 true
    }
    advertise-ipv6-next-hops {
        ipv4 true
    }
}
neighbor "2001:db8::2:1" {    # PE-1 system address
    group "gr_v6_internal"
}
neighbor "2001:db8::2:2" {    # PE-2 system address
    group "gr_v6_internal"
}
}
exit all

```

Configure CE-1 and CE-2 for native IPv4 and IPv6 data

This configuration includes:

- ports and IPv4 and IPv6 interfaces between CE-1 and PE-1 and between CE-2 and PE-2
- an IPv4 system address and an IPv6 system address for CE-1 and for CE-2
- BGP, with external group “gr_v6_external” that includes the following capabilities:
 - IPv4 and IPv6 address families
 - extended-nh-encoding for IPv4
 - advertise-ipv6-next-hops for IPv4
 - BGP neighbor **interface** IPv6 addresses, with BGP neighbors in a different external autonomous system

The following example configuration applies for PE-1 and is similar for PE-2. The **strip-srv6-tlvs** command (per address family) prevents PE-1 from advertising SRv6 TLVs to the BGP neighbor.

```

[/]
A:admin@PE-1# configure {
  router "Base" {
    interface "int-PE-1-CE-1" {
      description "interface between PE-1 and CE-1"
      port 1/1/c6/1:1000
      ipv4 {
        primary {
          address 172.16.14.1
          prefix-length 30
        }
      }
      ipv6 {
        address 2001:db8::168:14:1 {
          prefix-length 126
        }
      }
    }
    bgp {
      group "gr_v6_external" {
        description "external bgp group on PE-1"
        family {
          ipv4 true
          ipv6 true
        }
      }
    }
  }
}

```

```

        extended-nh-encoding {
            ipv4 true
        }
        advertise-ipv6-next-hops {
            ipv4 true
        }
    }
    neighbor "2001:db8::168:14:2" {
        group "gr_v6_external"
        type external
        peer-as 64504
        segment-routing-v6 {
            route-advertisement {
                family ipv4 {
                    strip-srv6-tlvs true
                }
                family ipv6 {
                    strip-srv6-tlvs true
                }
            }
        }
    }
}
exit all

```

The following example configuration applies for CE-2 and is similar for CE-1.

```

[/]
A:admin@CE-2# configure {
    router "Base" {
        autonomous-system 64505
        interface "int-CE-2-PE-2" {
            description "interface between CE-2 and PE-2"
            port 1/1/c1/1:1000
            ipv4 {
                primary {
                    address 172.16.25.2
                    prefix-length 30
                }
            }
            ipv6 {
                address 2001:db8::168:25:2 {
                    prefix-length 126
                }
            }
        }
        interface "system" {
            description "system interface of CE-2"
            ipv4 {
                primary {
                    address 192.0.2.5    # used for IPv4 ping
                    prefix-length 32
                }
            }
            ipv6 {
                address 2001:db8::2:5 {    # used for IPv6 ping
                    prefix-length 128
                }
            }
        }
    }
    bgp {
        min-route-advertisement 1
        router-id 2.2.2.5
    }
}

```



```
rapid-withdrawal true
split-horizon true
ebgp-default-reject-policy {
  import false # do not refuse eBGP imported policies
  export false # do not prevent eBGP exported policies
}
group "gr_v6_external" {
  description "external bgp group on CE-2"
  family {
    ipv4 true
    ipv6 true
  }
  extended-nh-encoding {
    ipv4 true
  }
  advertise-ipv6-next-hops {
    ipv4 true
  }
}
neighbor "2001:db8::168:25:1" {
  group "gr_v6_external"
  type external
  peer-as 64500
}
}
exit all
```

Ensure the export of the system addresses of CE-1 and CE-2

Configure a policy on CE-2 that imports the IPv4 and IPv6 prefixes into BGP. Configure a similar policy on CE-1.

```
[/]
A:admin@CE-2# configure {
  policy-options {
    prefix-list "CE-2_prefixes" {
      prefix 192.0.2.5/32 type exact {
      }
      prefix 2001:db8::2:5/128 type exact {
      }
    }
  }
  policy-statement "policy-export-bgp" {
    entry 10 {
      from {
        prefix-list ["CE-2_prefixes"]
      }
      action {
        action-type accept
      }
    }
  }
}
exit all
```

Apply this policy on CE-2 to the BGP neighbor PE-2. Perform a similar configuration on CE-1 to the BGP neighbor PE-1.

```
[ex:/configure router "Base"]
A:admin@CE-2#
  bgp {
    neighbor "2001:db8::168:25:1" {
```

```

export {
  policy ["policy-export-bgp"]
}
exit all

```

Verify the IPv4 and IPv6 route tables. The corresponding FIBs can be verified with the **show router fib 1 ipv4** and **show router fib 1 ipv6** commands.

On CE-1:

192.0.2.4/32 is the IPv4 system address of CE-1. 192.0.2.5/32 is the IPv4 system address of CE-2, which is reached via PE-1.

```

[/]
A:admin@CE-1# show router route-table ipv4
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age      Pref
  Next Hop[Interface Name]      Metric
-----
172.16.14.0/30              Local  Local  00h05m19s  0
  int-CE-1-PE-1              0
192.0.2.4/32                Local  Local  00h05m19s  0
  system                      0
192.0.2.5/32              Remote BGP  00h00m26s  170
  2001:db8::168:14:1      0
-----
No. of Routes: 3
---snip---
=====

```

2001:db8::2:4/128 is the IPv6 system address of CE-1. 2001:db8::2:5/128 is the IPv6 system address of CE-2, which is reached via PE-1.

```

[/]
A:admin@CE-1# show router route-table ipv6
=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age      Pref
  Next Hop[Interface Name]      Metric
-----
2001:db8::2:4/128          Local  Local  00h05m19s  0
  system                    0
2001:db8::2:5/128      Remote BGP  00h00m26s  170
  2001:db8::168:14:1  0
2001:db8::168:14:0/126    Local  Local  00h05m18s  0
  int-CE-1-PE-1            0
-----
No. of Routes: 3
---snip---
=====

```

On PE-1:

```

[/]
A:admin@PE-1# show router route-table ipv4

```

```

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
  Next Hop[Interface Name]         Metric
-----
172.16.14.0/30                    Local  Local  00h06m21s  0
      int-PE-1-CE-1                0
192.0.2.4/32                     Remote BGP    00h00m40s  170
      2001:db8::168:14:2           0
192.0.2.5/32                     Remote BGP    00h00m21s  170
      fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"  10
-----
No. of Routes: 3
---snip---
=====

```

```

[/]
A:admin@PE-1# show router route-table ipv6

=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
  Next Hop[Interface Name]         Metric
-----
2001:db8::2:1/128                 Local  Local  00h12m43s  0
      system                        0
2001:db8::2:2/128                 Remote  ISIS   00h11m03s  18
      fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"  10
2001:db8::2:3/128                 Remote  ISIS   00h10m52s  18
      fe80::612:1ff:fe01:1-"int-PE-1-RR-3"  10
2001:db8::2:4/128                 Remote BGP    00h00m40s  170
      2001:db8::168:14:2           0
2001:db8::2:5/128                 Remote BGP    00h00m21s  170
      fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"  10
2001:db8::168:12:0/126            Local  Local  00h12m43s  0
      int-PE-1-PE-2                0
2001:db8::168:13:0/126            Local  Local  00h12m42s  0
      int-PE-1-RR-3                0
2001:db8::168:14:0/126            Local  Local  00h06m20s  0
      int-PE-1-CE-1                0
2001:db8::168:23:0/126            Remote  ISIS   00h11m03s  18
      fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"  20
-----
No. of Routes: 9
---snip---
=====

```

On PE-2:

```

[/]
A:admin@PE-2# show router route-table ipv4

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
  Next Hop[Interface Name]         Metric
-----
172.16.25.0/30                    Local  Local  00h06m04s  0
      int-PE-2-CE-2                0
-----

```

```

192.0.2.4/32          Remote BGP      00h00m42s 170
                    fe80::60a:1ff:fe01:1-"int-PE-2-PE-1"
                    10
192.0.2.5/32        Remote BGP      00h00m24s 170
                    2001:db8::168:25:2
                    0
-----
No. of Routes: 3
---snip---
=====

```

```

[/]
A:admin@PE-2# show router route-table ipv6

=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age           Pref
Next Hop[Interface Name]  Metric
-----
2001:db8::2:1/128          Remote ISIS   00h11m00s 18
                    fe80::60a:1ff:fe01:1-"int-PE-2-PE-1"
                    10
2001:db8::2:2/128          Local  Local   00h12m27s  0
                    system
                    0
2001:db8::2:3/128          Remote ISIS   00h10m54s 18
                    fe80::612:1ff:fe01:b-"int-PE-2-RR-3"
                    10
2001:db8::2:4/128        Remote BGP      00h00m42s 170
                    fe80::60a:1ff:fe01:1-"int-PE-2-PE-1"
                    10
2001:db8::2:5/128        Remote BGP      00h00m24s 170
                    2001:db8::168:25:2
                    0
2001:db8::168:12:0/126     Local  Local   00h12m26s  0
                    int-PE-2-PE-1
                    0
2001:db8::168:13:0/126     Remote ISIS   00h11m00s 18
                    fe80::60a:1ff:fe01:1-"int-PE-2-PE-1"
                    20
2001:db8::168:23:0/126     Local  Local   00h12m26s  0
                    int-PE-2-RR-3
                    0
2001:db8::168:25:0/126     Local  Local   00h06m03s  0
                    int-PE-2-CE-2
                    0
-----
No. of Routes: 9
---snip---
=====

```

IPv4 data transport is not possible between CE-1 and CE-2. Verify this with a **ping** from CE-1 to the IPv4 system address that CE-2 advertises.

```

[/]
A:admin@CE-1# ping 192.0.2.5
PING 192.0.2.5 56 data bytes
... .. Request timed out. icmp_seq=1.
Request timed out. icmp_seq=2.
---snip---
---- 192.0.2.5 PING Statistics ----
5 packets transmitted, 0 packets received, 100% packet loss

```

IPv6 data transport is possible between CE-1 and CE-2, although not by using SRv6 between PE-1 and PE-2 but by using native IPv6. Verify this with a **ping** and a **traceroute** from CE-1 to the IPv6 system address that CE-2 advertises. Native IPv6 data flows over an IPv6 interface from CE-1 to PE-1, from there over an IPv6 interface to PE-2, and from there over an IPv6 interface to CE-2. The same is true for data transport between CE-2 and CE-1.

```
[/]
```

```
A:admin@CE-1# ping 2001:db8::2:5
PING 2001:db8::2:5 56 data bytes
64 bytes from 2001:db8::2:5 icmp_seq=1 hlim=62 time=2.18ms.
---snip---
---- 2001:db8::2:5 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 1.85ms, avg = 2.07ms, max = 2.18ms, stddev = 0.122ms
```

```
[/]
A:admin@CE-1# traceroute 2001:db8::2:5
traceroute to 2001:db8::2:5, 30 hops max, 60 byte packets
 1 2001:db8::168:14:1 (2001:db8::168:14:1)  1.02 ms  0.790 ms  0.837 ms
 2 2001:db8::168:12:2 (2001:db8::168:12:2)  1.28 ms  1.26 ms  1.41 ms
 3 2001:db8::2:5 (2001:db8::2:5)  1.80 ms  1.99 ms  1.82 ms
```

Configure SRv6 in the router Base context on PE-1 and PE-2

Configure the locator in the **router Base segment-routing segment-routing-v6** context on PE-2. Perform a similar configuration on PE-1, with **ip-prefix 2001:db8:aaaa:101::/64** for locator "PE-1_loc".

```
[ex:/configure router "Base" segment-routing]
A:admin@PE-2#
    segment-routing-v6 {
        locator "PE-2_loc" {
            admin-state enable
            block-length 48
            prefix {
                ip-prefix 2001:db8:aaaa:102::/64
            }
        }
    }
exit all
```

Configure the FPEs on PE-1 and PE-2.

```
[ex:/configure]
A:admin@PE-2#
    fwd-path-ext {
        fpe 1 {
            path {
                pxc 1
            }
            application {
                srv6 {
                    type origination
                }
            }
        }
        fpe 2 {
            path {
                pxc 2
            }
            application {
                srv6 {
                    type termination
                }
            }
        }
    }
exit all
```

Use FPE 1 as the SRv6 origination FPE in the **router Base segment-routing segment-routing-v6** context and FPE 2 as the SRv6 termination FPE in the **router Base segment-routing segment-routing-v6 locator** context on PE-2. Perform a similar configuration on PE-1 for locator "PE-1_loc".

```
[ex:/configure router "Base" segment-routing]
A:admin@PE-2#
    segment-routing-v6 {
        origination-fpe [1]
        locator "PE-2_loc" {
            admin-state enable
            termination-fpe [2]
        }
    }
exit all
```

Configure the SRv6 End function (equivalent to an IPv4 node SID) and SRv6 End.X functions (equivalent to IPv4 Adjacency SIDs) in the **router Base segment-routing segment-routing-v6 base-routing-instance locator** context on PE-2. Perform a similar configuration on PE-1 for locator "PE-1_loc".

```
[ex:/configure router "Base" segment-routing]
A:admin@PE-2#
    segment-routing-v6 {
        base-routing-instance {
            locator "PE-2_loc" {
                function {
                    end 1 {
                        srh-mode usp
                    }
                    end-x-auto-allocate psp protection unprotected { }
                }
            }
        }
    }
exit all
```

Advertise the locator in IS-IS while ensuring level 2 capability on PE-2. Perform a similar configuration on PE-1 for locator "PE-1_loc".

```
[ex:/configure router "Base" isis 0]
A:admin@PE-2#
    segment-routing-v6 {
        admin-state enable
        locator "PE-2_loc" {
            level-capability 2
        }
    }
exit all
```

A summary on locator and origination FPE configuration can be verified with the **show router segment-routing-v6 summary** command.

Verify the SRv6 local SIDs on PE-2 and similar on PE-1. Three SRv6 local SIDs are created: one for the statically configured SRv6 End function (configured in the base context) and two for the auto-allocated SRv6 End.X functions (one facing PE-1 and one facing RR-3). All three SRv6 local SIDs are concatenated with the locator. The statically configured SRv6 End function appears first with function number 1. The auto-allocated SRv6 End.X functions get subsequent function numbers, 2 and 4 respectively. RR-3 has no SRv6 configuration and does not have these SRv6 local SIDs and SRv6 functions.

```
[/]
A:admin@PE-2# show router segment-routing-v6 local-sid
```

```
Segment Routing v6 Local SIDs
=====
SID                                     Type      Function
Locator
Context
-----
2001:db8:aaaa:102:0:1000::             End       1
  PE-2_loc
  Base
2001:db8:aaaa:102:0:2000::             End.X    2
  PE-2_loc
  None
2001:db8:aaaa:102:0:4000::             End.X    4
  PE-2_loc
  None
-----
SIDs : 3
=====
```

Verify the SRv6 base routing instance details on PE-2 and similar on PE-1. The SRv6 End function is statically configured. There is an auto-allocated SRv6 End.X function for each IS-IS neighbor.

```
[/]
A:admin@PE-2# show router segment-routing-v6 base-routing-instance

=====
Segment Routing v6 Base Routing Instance
=====
Locator
Type      Function      SID      Status/InstId
SRH-mode Protection Interface
-----
PE-2_loc
End       1 2001:db8:aaaa:102:0:1000::    ok
  USP
-----
Auto-allocated End.X: PSP Unprotected,
-----
End.X     *2 2001:db8:aaaa:102:0:2000::    0
  PSP     Unprotected int-PE-2-PE-1
  ISIS Level: L2 Mac Address: 04:0a:01:01:00:01 Nbr Sys Id: 0010.0100.1001
End.X     *4 2001:db8:aaaa:102:0:4000::    0
  PSP     Unprotected int-PE-2-RR-3
  ISIS Level: L2 Mac Address: 04:12:01:01:00:0b Nbr Sys Id: 0010.0100.1003
-----
Legend: * - System allocated
```

Verify the IPv6 route table on PE-1. The IPv6 route table has also routes to the local and the learned remote locators and to the local SRv6 function SIDs. The remotely configured locator prefix of PE-2 is reached via an SRv6 tunnel. The routes with protocol “SRv6” correspond with the locally configured locator prefix of PE-1, or the locally configured SRv6 End function.

```
[/]
A:admin@PE-1# show router route-table ipv6

=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]      Type      Proto      Age      Pref
Next Hop[Interface Name] Metric
-----
```

```

-----snip-----
2001:db8:aaaa:101::/64          Local   SRV6    00h02m44s  3
    fe80::201- "_tmnx_fpe_2.a"          0
2001:db8:aaaa:101:0:1000::/128  Local   SRV6    00h01m33s  3
    Black Hole                          0
2001:db8:aaaa:101:0:2000::/128  Local   ISIS    00h00m45s  18
    fe80::60e:1ff:fe01:1-"int-PE-1-PE-2" 10
2001:db8:aaaa:101:0:4000::/128  Local   ISIS    00h00m45s  18
    fe80::612:1ff:fe01:1-"int-PE-1-RR-3" 10
2001:db8:aaaa:102::/64          Remote  ISIS    00h00m31s  18
    2001:db8:aaaa:102::/64 (tunneled:SRV6-ISIS) 10
-----
No. of Routes: 14
---snip---
=====

```

Verify the IS-IS routes on PE-1 and similar on PE-2. This corresponds with the information in the route table (and FIB). IS-IS is not configured on CE-1 and CE-2, so CE-1 and CE-2 have no IS-IS routes.

```

[/]
A:admin@PE-1# show router isis routes

=====
Rtr Base ISIS Instance 0 Route Table
=====
Prefix[Flags]
NextHop          Metric    Lvl/Typ    Ver.  SysID/Hostname
MT              AdminTag/SID[F]
-----
2001:db8::2:1/128      0         2/Int.     2     PE-1
::                0         0
2001:db8::2:2/128      10        2/Int.     9     PE-2
    fe80::60e:1ff:fe01:1-"int-PE-1-PE-2" 0
2001:db8::2:3/128      10        2/Int.     9     RR-3
    fe80::612:1ff:fe01:1-"int-PE-1-RR-3" 0
2001:db8::168:12:0/126 10         2/Int.     4     PE-1
::                0         0
2001:db8::168:13:0/126 10         2/Int.     4     PE-1
::                0         0
2001:db8::168:23:0/126 20         2/Int.     9     PE-2
    fe80::60e:1ff:fe01:1-"int-PE-1-PE-2" 0
2001:db8:aaaa:101::/64  0         2/Int.     11    PE-1
::                0         0
2001:db8:aaaa:102::/64  10        2/Int.     10    PE-2
    fe80::60e:1ff:fe01:1-"int-PE-1-PE-2" 0
-----
No. of Routes: 8 (8 paths)
---snip---
=====

```

The locator prefixes and who advertises them can be verified with the **show router isis segment-routing-v6 locator** command. The SRV6 End SIDs and who advertises them can be verified with the **show router isis segment-routing-v6 end-sid** command.

The IS-IS data base can be verified with the **show router isis database detail** command.

The output of this command provides information on each IS-IS-enabled router. Per uniquely identified IS-IS-enabled router, the SRV6 information indicates:

- the IS-IS-advertised router capabilities
- the advertised SRV6 locator TLV

- the advertised configured SRv6 End SID and auto-allocated SRv6 End.X SIDs

The BGP groups can be verified with the **show router bgp group** command. PE-1 and PE-2 know the iBGP and eBGP peers. RR-3 only knows the iBGP peers. CE-1 and CE-2 only know the eBGP peers.

The BGP next hops can be verified with the **show router bgp next-hop ipv4** and **show router bgp next-hop ipv6** commands.

Configure SRv6 End.DT4 and SRv6 End.DT6 functions on PE-1 and PE-2

Configure SRv6 End.DT4 and SRv6 End.DT6 functions in the **router Base segment-routing segment-routing-v6 base-routing-instance locator function** context on PE-1. They can have statically or automatically allocated values. For statically allocated values, an SRv6 reserved label block must be configured. Perform an identical configuration on PE-2.

```
[ex:/configure router "Base"]
A:admin@PE-1#
  mpls-labels {
    sr-labels {
      start 20000
      end 20999
    }
    reserved-label-block "SRv6" {
      start-label 30100
      end-label 30199
    }
  }
exit all
```

This SRv6 reserved label block must be referenced in the **router Base segment-routing segment-routing-v6 locator static-function** context on PE-2, where also the total number of static functions, including the already existing SRv6 End function (with value 1), must be set. Perform a similar configuration on PE-1 for locator "PE-1_loc".

```
[ex:/configure router "Base" segment-routing]
A:admin@PE-2#
  segment-routing-v6 {
    source-address 2001:db8::2:2
    locator "PE-2_loc" {
      admin-state enable
      static-function {
        max-entries 3
        label-block "SRv6"
      }
    }
    base-routing-instance {
      locator "PE-2_loc" {
        function {
          end-dt4 {
            value 2
          }
          end-dt6 {
            value 3
          }
        }
      }
    }
  }
exit all
```

The SRv6 End.DT4 and SRv6 End.DT6 functions are allocated the unique static values of 2 and 3 respectively. The values allocated must not exceed the **max-entries** value.

Each PE must resolve the BGP next hop to an SRv6 End.DT4 or End.DT6 Segment ID. Therefore, each PE must advertise route prefixes within a BGP update message that includes an SRv6 Services TLV. This is achieved by configuring the **add-srv6-tlvs** command along with the locator value for each address family, IPv4 and IPv6.

When a PE receives a BGP update that includes the SRv6 Services TLV, the default behavior is to ignore this TLV, and resolve the next hop to the tunnel type configured in an **auto-bind-tunnel** statement. To override this behavior, **ignore-received-srv6-tlvs** must be set to false for IPv4 and IPv6 address families on PE-1. Perform a similar configuration on PE-2 for locator "PE-2_loc".

```
[ex:/configure router "Base" bgp]
A:admin@PE-1#
    segment-routing-v6 {
        family ipv4 {
            ignore-received-srv6-tlvs false
            add-srv6-tlvs {
                locator-name "PE-1_loc"
            }
        }
        family ipv6 {
            ignore-received-srv6-tlvs false
            add-srv6-tlvs {
                locator-name "PE-1_loc"
            }
        }
    }
exit all
```

CE-2 sends BGP updates to PE-2 for the IPv4 and the IPv6 address families respectively. Each BGP update advertises the IPv4 or IPv6 address family, the reachable network prefixes, and the autonomous system to which they belong. PE-2 adds an SRv6 Services TLV, indicating that resolution to an SRv6 SID is available, making use of the endpoint behavior that is configured for the IPv4 or IPv6 address family on the locator. PE-2 advertises the BGP updates to PE-1 via the RR. PE-1 programs the route prefixes in its route table and FIB with an SRv6 tunnel next hop, and forwards the BGP updates to CE-1. CE-1 programs the learned route prefixes in its route table and FIB.

Similar BGP updates flow from CE-1 to CE-2, via PE-1, RR-3, and PE-2. PE-1 and PE-2 advertise only the SRv6 SIDs for the SRv6 End.DT4 and SRv6 End.DT6 functions.

After the BGP updates, the IS-IS data base remains the same, except for the renumbering of the SRv6 End.X functions. This can be verified with the **show router isis database detail** command.

The BGP next hops remain the same, except for the next hops to the system addresses of PE-1 and PE-2 that switch owner from "ISIS" to "N/A". This can be verified with the **show router bgp next-hop ipv4** and **show router bgp next-hop ipv6** commands.

When debug logging for BGP updates is configured, the configuration results in the following BGP update logs for the IPv4 address family.

Focus as an example on prefix 192.0.2.5/32 and on prefix 192.0.2.4.1/32, but in the other direction.

Verify the IPv4 BGP routes.

CE-2 advertises route prefix 192.0.2.5/32 to PE-2 (in RIB Out Entries).

```
[/]
A:admin@CE-2# show router bgp routes 192.0.2.5 hunt
=====
```

```

BGP Router ID:2.2.2.5      AS:64505      Local AS:64505
=====
---snip---
=====
BGP IPv4 Routes
=====
-----
RIB In Entries
-----
-----
RIB Out Entries
-----
Network      : 192.0.2.5/32
NextHop      : 2001:db8::168:25:2
Path Id      : None
To           : 2001:db8::168:25:1
Res. Protocol : INVALID      Res. Metric   : 0
Res. NextHop  : n/a
Local Pref.   : n/a          Interface Name : NotAvailable
Aggregator AS : None         Aggregator    : None
Atomic Aggr.  : Not Atomic   MED           : None
AIGP Metric   : None         IGP Cost      : n/a
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None         Peer Router Id : 2.2.2.2
Origin        : IGP
AS-Path       : 64505
Route Tag     : 0
Neighbor-AS   : 64505
Orig Validation: NotFound
Source Class  : 0            Dest Class    : 0
-----
Routes : 1
=====

```

PE-2 receives the BGP update which CE-2 sends for the IPv4 address family:

```

[/]
A:admin@PE-2# show log log-id "log_2"

---snip---
5 2022/07/19 11:29:41.236 CEST MINOR: DEBUG #2001 Base Peer 1: 2001:db8::168:25:2
"Peer 1: 2001:db8::168:25:2: UPDATE
Peer 1: 2001:db8::168:25:2 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 43
  Flag: 0x90 Type: 14 Len: 26 Multiprotocol Reachable NLRI:
    Address Family IPV4
      NextHop len 16 Global NextHop 2001:db8::168:25:2
      192.0.2.5/32
      Flag: 0x40 Type: 1 Len: 1 Origin: 0
      Flag: 0x40 Type: 2 Len: 6 AS Path:
        Type: 2 Len: 1 < 64505 >
"
---snip---

```

Upon receipt of the BGP update from CE-2, PE-2 programs route prefix 192.0.2.5/32 in its route table and FIB, with the interface towards CE-2 as next hop.

Verify the resulting IPv4 route table on PE-2:

```
[/]
A:admin@PE-2# show router route-table ipv4

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
  Next Hop[Interface Name]          Metric
-----
172.16.25.0/30                    Local Local  00h16m25s  0
      int-PE-2-CE-2                  0
192.0.2.4/32                      Remote BGP    00h00m49s  170
      2001:db8:aaaa:101:0:2000:: (tunneled:SRV6)  10
192.0.2.5/32                    Remote BGP  00h10m45s  170
      2001:db8::168:25:2                0
-----
No. of Routes: 3
---snip---
```

Verify the corresponding IPv4 BGP routes on PE-2:

```
[/]
A:admin@PE-2# show router bgp routes 192.0.2.5 hunt

=====
BGP Router ID:2.2.2.2          AS:64500          Local AS:64500
=====
---snip---
```

BGP IPv4 Routes

```
=====
RIB In Entries
-----
Network       : 192.0.2.5/32
NextHop       : 2001:db8::168:25:2
Path Id       : None
From          : 2001:db8::168:25:2
Res. Protocol : LOCAL          Res. Metric   : 0
Res. NextHop  : 2001:db8::168:25:2
Local Pref.   : None
Aggregator AS : None          Interface Name : int-PE-2-CE-2
Atomic Aggr.  : Not Atomic  Aggregator    : None
AIGP Metric   : None          MED           : None
Connector     : None          IGP Cost      : 0
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None          Peer Router Id : 2.2.2.5
Fwd Class     : None          Priority       : None
Flags         : Used Valid Best IGP In-RTM
Route Source  : External
AS-Path       : 64505
Route Tag     : 0
Neighbor-AS   : 64505
Orig Validation: NotFound
Source Class  : 0          Dest Class    : 0
Add Paths Send : Default
RIB Priority   : Normal
Last Modified : 00h10m45s
-----
```

```

RIB Out Entries
-----
Network       : 192.0.2.5/32
Nexthop      : 2001:db8::2:2
Path Id         : None
To           : 2001:db8::2:3
Res. Protocol   : INVALID                Res. Metric    : 0
Res. Nexthop    : n/a
Local Pref.     : 100                    Interface Name : NotAvailable
Aggregator AS  : None                    Aggregator     : None
Atomic Aggr.   : Not Atomic              MED            : None
AIGP Metric     : None                    IGP Cost       : 0
Connector      : None
Community       : No Community Members
Cluster        : No Cluster Members
Originator Id  : None                    Peer Router Id : 2.2.2.3
Origin         : IGP
AS-Path      : 64505
Route Tag       : 0
Neighbor-AS    : 64505
Orig Validation: NotFound
Source Class    : 0                      Dest Class     : 0
SRv6 TLV Type : SRv6 L3 Service TLV (5)
SRv6 SubTLV  : SRv6 SID Information (1)
Sid          : 2001:db8:aaaa:102:0:2000::
Behavior     : End.DT4 (19)
SRv6 SubSubTLV : SRv6 SID Structure (1)
Loc-Block-Len : 48                    Loc-Node-Len  : 16
Func-Len      : 20                    Arg-Len       : 0
Tpose-Len     : 0                      Tpose-offset  : 0
-----
Routes : 2
=====

```

PE-2 then advertises route prefix 192.0.2.5/32 (in RIB Out Entries), via RR-3, and inserts the SRv6 Services TLV. This TLV carries an SRv6 Service Information sub-TLV that contains the End.DT4 SID.

PE-1 receives (via RR-3) the BGP update which PE-2 sends for the IPv4 address family:

```

[/]
A:admin@PE-1# show log log-id "log_2"

---snip---
3 2022/07/19 11:39:38.404 CEST MINOR: DEBUG #2001 Base Peer 1: 2001:db8::2:3
"Peer 1: 2001:db8::2:3: UPDATE
Peer 1: 2001:db8::2:3 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 104
  Flag: 0x90 Type: 14 Len: 26 Multiprotocol Reachable NLRI:
    Address Family IPV4
      NextHop len 16 Global NextHop 2001:db8::2:2
      192.0.2.5/32
      Flag: 0x40 Type: 1 Len: 1 Origin: 0
      Flag: 0x40 Type: 2 Len: 6 AS Path:
        Type: 2 Len: 1 < 64505 >
      Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
      Flag: 0x80 Type: 9 Len: 4 Originator ID: 2.2.2.2
      Flag: 0x80 Type: 10 Len: 4 Cluster ID:
        3.3.3.3
      Flag: 0xc0 Type: 40 Len: 37 Prefix-SID-attr:
        SRv6 Services TLV (37 bytes):-
          Type: SRV6 L3 Service TLV (5)

```

```

Length: 34 bytes, Reserved: 0x0
SRv6 Service Information Sub-TLV (33 bytes)
  Type: 1 Len: 30 Rsvd1: 0x0
  SRv6 SID: 2001:db8:aaaa:102:0:2000::
  SID Flags: 0x0 Endpoint Behavior: 0x13 Rsvd2: 0x0
  SRv6 SID Sub-Sub-TLV
    Type: 1 Len: 6
    BL:48 NL:16 FL:20 AL:0 TL:0 T0:0
"
---snip---

```

Upon receipt of the BGP update from RR-3 on behalf of PE-2, PE-1 programs route prefix 192.0.2.5/32 in its route table and FIB. The presence of the SRv6 Services TLV indicates that the next hop is the SRv6 End.DT4 SID which, in turn, is resolved to the remote locator for PE-2.

PE-1 then advertises route prefix 192.0.2.5/32 to CE-1 (in RIB Out Entries).

Verify the resulting IPv4 route table on PE-1. The IPv4 route table has a route to the remote IPv4 system address of CE-2, via the End.DT4 SID of the remotely configured locator prefix of PE-2.

```

[/]
A:admin@PE-1# show router route-table ipv4

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age           Pref
  Next Hop[Interface Name]           Metric
-----
172.16.14.0/30              Local  Local  00h16m41s    0
  int-PE-1-CE-1              0
192.0.2.4/32                Remote BGP     00h11m01s   170
  2001:db8::168:14:2          0
192.0.2.5/32              Remote BGP  00h00m46s   170
  2001:db8:aaaa:102:0:2000:: (tunneled:SRV6)  10
-----
No. of Routes: 3
---snip---
=====

```

Verify the corresponding IPv4 BGP routes on PE-1:

```

[/]
A:admin@PE-1# show router bgp routes 192.0.2.5 hunt

=====
BGP Router ID:2.2.2.1      AS:64500      Local AS:64500
=====
---snip---
=====
BGP IPv4 Routes
=====
RIB In Entries
-----
Network       : 192.0.2.5/32
Nexthop      : 2001:db8::2:2
Path Id      : None
From         : 2001:db8::2:3
Res. Protocol : ISIS           Res. Metric   : 10
Res. Nexthop : fe80::60e:1ff:fe01:1
Local Pref.  : 100
Aggregator AS : None           Aggregator    : None

```

```

Atomic Aggr. : Not Atomic          MED : None
AIGP Metric  : None                IGP Cost : 10
Connector    : None
Community    : No Community Members
Cluster      : 3.3.3.3
Originator Id : 2.2.2.2            Peer Router Id : 2.2.2.3
Fwd Class    : None                Priority : None
Flags        : Used Valid Best IGP In-RTM
Route Source : Internal
AS-Path      : 64505
Route Tag    : 0
Neighbor-AS  : 64505
Orig Validation: NotFound
Source Class : 0                    Dest Class : 0
Add Paths Send : Default
RIB Priority  : Normal
Last Modified : 00h00m46s
SRv6 TLV Type : SRv6 L3 Service TLV (5)
SRv6 SubTLV  : SRv6 SID Information (1)
Sid          : 2001:db8:aaaa:102:0:2000::
Behavior     : End.DT4 (19)
SRv6 SubSubTLV : SRv6 SID Structure (1)
Loc-Block-Len : 48                 Loc-Node-Len : 16
Func-Len      : 20                 Arg-Len       : 0
Tpose-Len     : 0                  Tpose-offset  : 0

```

RIB Out Entries

```

Network      : 192.0.2.5/32
Nexthop      : 2001:db8::168:14:1
Path Id      : None
To           : 2001:db8::168:14:2
Res. Protocol : INVALID           Res. Metric : 0
Res. Nexthop  : n/a
Local Pref.   : n/a               Interface Name : NotAvailable
Aggregator AS : None              Aggregator   : None
Atomic Aggr.  : Not Atomic         MED           : None
AIGP Metric   : None               IGP Cost      : 10
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None                Peer Router Id : 2.2.2.4
Origin        : IGP
AS-Path       : 64500 64505
Route Tag     : 0
Neighbor-AS   : 64500
Orig Validation: NotFound
Source Class  : 0                    Dest Class    : 0

```

Routes : 2
=====

CE-1 receives the BGP update which PE-1 sends for the IPv4 address family:

```

[/]
A:admin@CE-1# show log log-id "log_2"

---snip---
1 2022/07/19 11:39:39.220 CEST MINOR: DEBUG #2001 Base Peer 1: 2001:db8::168:14:1
"Peer 1: 2001:db8::168:14:1: UPDATE
Peer 1: 2001:db8::168:14:1 - Received BGP UPDATE:

```

```

Withdrawn Length = 0
Total Path Attr Length = 47
Flag: 0x90 Type: 14 Len: 26 Multiprotocol Reachable NLRI:
  Address Family IPV4
  NextHop len 16 Global NextHop 2001:db8::168:14:1
  192.0.2.5/32
Flag: 0x40 Type: 1 Len: 1 Origin: 0
Flag: 0x40 Type: 2 Len: 10 AS Path:
  Type: 2 Len: 2 < 64500 64505 >
"
---snip---

```

Upon receipt of the BGP update from PE-1, CE-1 programs route prefix 192.0.2.5/32 in its route table and FIB with the interface towards PE-1 as the next hop.

Verify the resulting IPv4 route table on CE-1:

```

[/]
A:admin@CE-1# show router route-table ipv4

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]         Metric
-----
172.16.14.0/30                    Local  Local   00h15m40s    0
  int-CE-1-PE-1                    0
192.0.2.4/32                      Local  Local   00h15m40s    0
  system                            0
192.0.2.5/32                    Remote BGP  00h00m51s  170
  2001:db8::168:14:1              0
-----
No. of Routes: 3
---snip---
=====

```

Verify the corresponding IPv4 BGP routes on CE-1:

```

[/]
A:admin@CE-1# show router bgp routes 192.0.2.5 hunt

=====
BGP Router ID:2.2.2.4           AS:64504           Local AS:64504
=====
---snip---
=====
BGP IPv4 Routes
=====
-----
RIB In Entries
-----
Network       : 192.0.2.5/32
Nexthop       : 2001:db8::168:14:1
Path Id       : None
From          : 2001:db8::168:14:1
Res. Protocol : LOCAL           Res. Metric   : 0
Res. Nexthop  : 2001:db8::168:14:1
Local Pref.   : None           Interface Name : int-CE-1-PE-1
Aggregator AS : None           Aggregator    : None
Atomic Aggr.  : Not Atomic    MED           : None
AIGP Metric   : None           IGP Cost      : 0
Connector     : None
Community     : No Community Members

```



```

Cluster      : No Cluster Members
Originator Id : None                Peer Router Id : 2.2.2.1
Fwd Class    : None                Priority       : None
Flags        : Used Valid Best IGP In-RTM
Route Source : External
AS-Path      : 64500 64505
Route Tag    : 0
Neighbor-AS  : 64500
Orig Validation: NotFound
Source Class : 0                    Dest Class    : 0
Add Paths Send : Default
RIB Priority  : Normal
Last Modified : 00h00m51s
    
```

RIB Out Entries

Routes : 1
=====

Similar BGP update logs are generated for the IPv6 address family.

Focus as an example on prefix 2001:db8::2:5/128 and on prefix 2001:db8::2:4/128, but in the other direction.

Verify the IPv6 BGP routes.

CE-2 advertises route prefix 2001:db8::2:5/128 to PE-2 (in RIB Out Entries).

```

[/]
A:admin@CE-2# show router bgp routes 2001:db8::2:5/128 hunt
=====
BGP Router ID:2.2.2.5          AS:64505          Local AS:64505
=====
---snip---
=====
BGP IPv6 Routes
=====
-----
RIB In Entries
-----
-----
RIB Out Entries
-----
Network      : 2001:db8::2:5/128
Nexthop      : 2001:db8::168:25:2
Path Id      : None
To           : 2001:db8::168:25:1
Res. Protocol : INVALID          Res. Metric      : 0
Res. Nexthop  : n/a
Local Pref.   : n/a              Interface Name   : NotAvailable
Aggregator AS : None              Aggregator      : None
Atomic Aggr. : Not Atomic        MED             : None
AIGP Metric   : None              IGP Cost        : n/a
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None                Peer Router Id  : 2.2.2.2
Origin        : IGP
AS-Path       : 64505
Route Tag     : 0
Neighbor-AS   : 64505
    
```

```

Orig Validation: NotFound
Source Class   : 0                      Dest Class   : 0
-----
Routes : 1
=====

```

PE-2 receives the BGP update which CE-2 sends for the IPv6 address family:

```

[/]
A:admin@PE-2# show log log-id "log_2"

---snip---
6 2022/07/19 11:29:41.237 CEST MINOR: DEBUG #2001 Base Peer 1: 2001:db8::168:25:2
"Peer 1: 2001:db8::168:25:2: UPDATE
Peer 1: 2001:db8::168:25:2 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 55
  Flag: 0x90 Type: 14 Len: 38 Multiprotocol Reachable NLRI:
    Address Family IPV6
    NextHop len 16 Global NextHop 2001:db8::168:25:2
    2001:db8::2:5/128
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 6 AS Path:
    Type: 2 Len: 1 < 64505 >
"
---snip---

```

Upon receipt of the BGP update from CE-2, PE-2 programs route prefix 2001:db8::2:5/128 in its route table and FIB, with the interface towards CE-2 as next hop.

Verify the resulting IPv6 route table on PE-2:

```

[/]
A:admin@PE-2# show router route-table ipv6

=====
IPv6 Route Table (Router: Base)
=====

```

Dest Prefix[Flags] Next Hop[Interface Name]	Type	Proto	Age Metric	Pref
2001:db8::2:4/128 2001:db8:aaaa:101:0:3000:: (tunneled:SRV6)	Remote	BGP	00h00m49s 10	170
2001:db8::2:5/128 2001:db8::168:25:2	Remote	BGP	00h10m45s 0	170
2001:db8:aaaa:101::/64 2001:db8:aaaa:101::/64 (tunneled:SRV6-ISIS)	Remote	ISIS	00h03m20s 10	18
2001:db8:aaaa:102::/64 fe80::201-"_tmnx_fpe_2.a"	Local	SRV6	00h01m42s 0	3
2001:db8:aaaa:102:0:1000::/128 Black Hole	Local	SRV6	00h01m42s 0	3
2001:db8:aaaa:102:0:4000::/128 fe80::60a:1ff:fe01:1-"int-PE-2-PE-1"	Local	ISIS	00h01m42s 10	18
2001:db8:aaaa:102:0:5000::/128 fe80::612:1ff:fe01:b-"int-PE-2-RR-3"	Local	ISIS	00h01m42s 10	18

```

-----
No. of Routes: 14
---snip---
=====

```

Verify the corresponding IPv6 BGP routes on PE-2:

```
[/]
A:admin@PE-2# show router bgp routes 2001:db8::2:5/128 hunt
=====
BGP Router ID:2.2.2.2          AS:64500          Local AS:64500
=====
---snip---
=====
BGP IPv6 Routes
=====
-----
RIB In Entries
-----
-----
Network       : 2001:db8::2:5/128
Nexthop      : 2001:db8::168:25:2
Path Id         : None
From          : 2001:db8::168:25:2
Res. Protocol : LOCAL                      Res. Metric   : 0
Res. Nexthop  : 2001:db8::168:25:2
Local Pref.     : None                      Interface Name : int-PE-2-CE-2
Aggregator AS  : None                      Aggregator    : None
Atomic Aggr.   : Not Atomic                MED           : None
AIGP Metric    : None                      IGP Cost      : 0
Connector      : None
Community      : No Community Members
Cluster        : No Cluster Members
Originator Id  : None                      Peer Router Id : 2.2.2.5
Fwd Class      : None                      Priority       : None
Flags          : Used Valid Best IGP In-RTM
Route Source  : External
AS-Path       : 64505
Route Tag      : 0
Neighbor-AS    : 64505
Orig Validation: NotFound
Source Class   : 0                          Dest Class    : 0
Add Paths Send : Default
RIB Priority    : Normal
Last Modified  : 00h10m45s
-----
-----
RIB Out Entries
-----
-----
Network       : 2001:db8::2:5/128
Nexthop      : 2001:db8::2:2
Path Id         : None
To           : 2001:db8::2:3
Res. Protocol : INVALID                      Res. Metric   : 0
Res. Nexthop  : n/a
Local Pref.     : 100                      Interface Name : NotAvailable
Aggregator AS  : None                      Aggregator    : None
Atomic Aggr.   : Not Atomic                MED           : None
AIGP Metric    : None                      IGP Cost      : 0
Connector      : None
Community      : No Community Members
Cluster        : No Cluster Members
Originator Id  : None                      Peer Router Id : 2.2.2.3
Origin         : IGP
AS-Path       : 64505
Route Tag      : 0
Neighbor-AS    : 64505
Orig Validation: NotFound
Source Class   : 0                          Dest Class    : 0
SRv6 TLV Type : SRv6 L3 Service TLV (5)
```

```

SRv6 SubTLV      : SRv6 SID Information (1)
Sid              : 2001:db8:aaaa:102:0:3000::
Behavior         : End.DT6 (18)
SRv6 SubSubTLV  : SRv6 SID Structure (1)
Loc-Block-Len   : 48                      Loc-Node-Len   : 16
Func-Len        : 20                      Arg-Len        : 0
Tpose-Len       : 0                      Tpose-offset   : 0
-----
Routes : 2
=====

```

PE-2 then advertises route prefix 2001:db8::2:5/128, via RR-3, and inserts the SRv6 Services TLV. This TLV carries an SRv6 Service Information sub-TLV that contains the End.DT6 SID.

PE-1 receives (via RR-3) the BGP update which PE-2 sends for the IPv6 address family:

```

[/]
A:admin@PE-1# show log log-id "log_2"

---snip---
4 2022/07/19 11:39:38.404 CEST MINOR: DEBUG #2001 Base Peer 1: 2001:db8::2:3
"Peer 1: 2001:db8::2:3: UPDATE
Peer 1: 2001:db8::2:3 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 116
  Flag: 0x90 Type: 14 Len: 38 Multiprotocol Reachable NLRI:
    Address Family IPV6
    NextHop len 16 Global NextHop 2001:db8::2:2
    2001:db8::2:5/128
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 6 AS Path:
    Type: 2 Len: 1 < 64505 >
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
  Flag: 0x80 Type: 9 Len: 4 Originator ID: 2.2.2.2
  Flag: 0x80 Type: 10 Len: 4 Cluster ID:
    3.3.3.3
  Flag: 0xc0 Type: 40 Len: 37 Prefix-SID-attr:
    SRv6 Services TLV (37 bytes):-
      Type: SRV6 L3 Service TLV (5)
      Length: 34 bytes, Reserved: 0x0
    SRv6 Service Information Sub-TLV (33 bytes)
      Type: 1 Len: 30 Rsvd1: 0x0
      SRv6 SID: 2001:db8:aaaa:102:0:3000::
      SID Flags: 0x0 Endpoint Behavior: 0x12 Rsvd2: 0x0
      SRv6 SID Sub-Sub-TLV
        Type: 1 Len: 6
        BL:48 NL:16 FL:20 AL:0 TL:0 T0:0
"
---snip---

```

Upon receipt of the BGP update from RR-3 on behalf of PE-2, PE-1 programs route prefix 2001:db8::2:5/128 in its route table and FIB. The presence of the SRv6 Services TLV indicates that the next hop is the SRv6 End.DT6 SID which, in turn, is resolved to the remote locator for PE-2.

PE-1 then advertises route prefix 2001:db8::2:5/128 to CE-1 (in RIB Out Entries).

Verify the resulting IPv6 route table on PE-1. The IPv6 route table has a route to the remote IPv6 system address of CE-2, now resolved to the End.DT6 SID of the remotely configured locator prefix of PE-2. The

local auto-allocated SRv6 End.X functions have a renumbered SID, because their initial SID is now used for the statically configured SRv6 End.DT4 and SRv6 End.DT6 functions.

```
[/]
A:admin@PE-1# show router route-table ipv6

=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type   Proto   Age      Pref
  Next Hop[Interface Name]                       Metric
-----
---snip---
2001:db8::2:5/128                               Remote BGP      00h00m46s 170
  2001:db8:aaaa:102:0:3000:: (tunneled:SRV6)      10
---snip---
2001:db8:aaaa:101::/64                             Local   SRV6     00h02m02s  3
  fe80::201-"tmnx_fpe_2.a"                          0
2001:db8:aaaa:101:0:1000::/128                     Local   SRV6     00h02m02s  3
  Black Hole                                          0
2001:db8:aaaa:101:0:4000::/128                     Local   ISIS     00h02m02s  18
  fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"              10
2001:db8:aaaa:101:0:5000::/128                     Local   ISIS     00h02m02s  18
  fe80::612:1ff:fe01:1-"int-PE-1-RR-3"              10
2001:db8:aaaa:102::/64                             Remote  ISIS     00h03m18s  18
  2001:db8:aaaa:102::/64 (tunneled:SRV6-ISIS)        10
-----
No. of Routes: 14
---snip---
=====
```

Verify the corresponding IPv6 BGP routes on PE-1:

```
[/]
A:admin@PE-1# show router bgp routes 2001:db8::2:5/128 hunt

=====
BGP Router ID:2.2.2.1          AS:64500          Local AS:64500
=====
---snip---
=====
BGP IPv6 Routes
=====
-----
RIB In Entries
-----
Network       : 2001:db8::2:5/128
Nexthop      : 2001:db8::2:2
Path Id         : None
From         : 2001:db8::2:3
Res. Protocol : ISIS           Res. Metric   : 10
Res. Nexthop  : fe80::60e:1ff:fe01:1
Local Pref.    : 100
Interface Name : int-PE-1-PE-2
Aggregator AS  : None           Aggregator    : None
Atomic Aggr.   : Not Atomic     MED           : None
AIGP Metric    : None           IGP Cost     : 10
Connector      : None
Community      : No Community Members
Cluster        : 3.3.3.3
Originator Id : 2.2.2.2           Peer Router Id : 2.2.2.3
Fwd Class      : None           Priority      : None
Flags          : Used Valid Best IGP In-RTM
Route Source  : Internal
AS-Path       : 64505
```

```

Route Tag      : 0
Neighbor-AS   : 64505
Orig Validation: NotFound
Source Class  : 0                               Dest Class    : 0
Add Paths Send : Default
RIB Priority   : Normal
Last Modified  : 00h00m46s
SRv6 TLV Type : SRv6 L3 Service TLV (5)
SRv6 SubTLV   : SRv6 SID Information (1)
Sid           : 2001:db8:aaaa:102:0:3000::
Behavior      : End.DT6 (18)
SRv6 SubSubTLV : SRv6 SID Structure (1)
Loc-Block-Len : 48                               Loc-Node-Len  : 16
Func-Len      : 20                               Arg-Len       : 0
Tpose-Len     : 0                               Tpose-offset  : 0
-----
RIB Out Entries
-----
Network       : 2001:db8::2:5/128
NextHop      : 2001:db8::168:14:1
Path Id      : None
To           : 2001:db8::168:14:2
Res. Protocol : INVALID                               Res. Metric    : 0
Res. NextHop  : n/a
Local Pref.   : n/a                               Interface Name : NotAvailable
Aggregator AS : None                               Aggregator    : None
Atomic Aggr.  : Not Atomic                         MED           : None
AIGP Metric   : None                               IGP Cost      : 10
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None                               Peer Router Id : 2.2.2.4
Origin        : IGP
AS-Path       : 64500 64505
Route Tag     : 0
Neighbor-AS   : 64500
Orig Validation: NotFound
Source Class  : 0                               Dest Class    : 0
-----
Routes : 2
=====

```

CE-1 receives the BGP update which PE-1 sends for the IPv6 address family:

```

[/]
A:admin@CE-1# show log log-id "log_2"
---snip---
2 2022/07/19 11:39:39.220 CEST MINOR: DEBUG #2001 Base Peer 1: 2001:db8::168:14:1
"Peer 1: 2001:db8::168:14:1: UPDATE
Peer 1: 2001:db8::168:14:1 - Received BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 59
  Flag: 0x90 Type: 14 Len: 38 Multiprotocol Reachable NLRI:
    Address Family IPV6
    NextHop len 16 Global NextHop 2001:db8::168:14:1
    2001:db8::2:5/128
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 10 AS Path:
    Type: 2 Len: 2 < 64500 64505 >
"

```

---snip---

Upon receipt of the BGP update from PE-1, CE-1 programs prefix 2001:db8::2:5/128 in its route table and FIB, with the interface towards PE-1 as next hop.

Verify the resulting IPv6 route table on CE-1:

```
[/]
A:admin@CE-1# show router route-table ipv6

=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
  Next Hop[Interface Name]         Metric
-----
2001:db8::2:4/128                 Local  Local  00h15m40s  0
  system
2001:db8::2:5/128                Remote BGP  00h00m51s  170
  2001:db8::168:14:1
2001:db8::168:14:0/126            Local  Local  00h15m39s  0
  int-CE-1-PE-1
-----
No. of Routes: 3
---snip---
```

Verify the corresponding IPv6 BGP routes on CE-1:

```
[/]
A:admin@CE-1# show router bgp routes 2001:db8::2:5/128 hunt

=====
BGP Router ID:2.2.2.4      AS:64504      Local AS:64504
=====
---snip---
```

```
=====
BGP IPv6 Routes
=====
RIB In Entries
-----
Network       : 2001:db8::2:5/128
Nexthop       : 2001:db8::168:14:1
Path Id       : None
From          : 2001:db8::168:14:1
Res. Protocol : LOCAL           Res. Metric   : 0
Res. Nexthop  : 2001:db8::168:14:1
Local Pref.   : None           Interface Name : int-CE-1-PE-1
Aggregator AS : None           Aggregator    : None
Atomic Aggr.  : Not Atomic    MED           : None
AIGP Metric   : None           IGP Cost      : 0
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None           Peer Router Id : 2.2.2.1
Fwd Class     : None           Priority       : None
Flags         : Used Valid Best IGP In-RTM
Route Source  : External
AS-Path       : 64500 64505
Route Tag     : 0
Neighbor-AS   : 64500
Orig Validation: NotFound
Source Class  : 0           Dest Class    : 0
```

```
Add Paths Send : Default
RIB Priority    : Normal
Last Modified  : 00h00m51s
```

```
-----
RIB Out Entries
-----
```

```
Routes : 1
=====
```

Verify the SRv6 local SIDs on PE-2 and similar on PE-1. The SRv6 local SIDs 2001:db8:aaaa:102:0:2000:: and 2001:db8:aaaa:102:0:3000:: now correspond with the additional SRv6 End.DT4 and SRv6 End.DT6 behavior that is configured on the locator for the data transport between CE-1 and CE-2. RR-3, CE-1, and CE-2 do not have SRv6 configuration and do not have SRv6 local SIDs.

```
[/]
A:admin@PE-2# show router segment-routing-v6 local-sid
```

```
=====
Segment Routing v6 Local SIDs
=====
```

SID	Locator Context	Type	Function
2001:db8:aaaa:102:0:1000::	PE-2_loc Base	End	1
2001:db8:aaaa:102:0:2000::	PE-2_loc Base	End.DT4	2
2001:db8:aaaa:102:0:3000::	PE-2_loc Base	End.DT6	3
2001:db8:aaaa:102:0:4000::	PE-2_loc None	End.X	4
2001:db8:aaaa:102:0:5000::	PE-2_loc None	End.X	5

```
-----
SIDs : 5
-----
=====
```

Verify the SRv6 base routing instance on PE-2 and similar on PE-1.

```
[/]
A:admin@PE-2# show router segment-routing-v6 base-routing-instance
```

```
=====
Segment Routing v6 Base Routing Instance
=====
```

Locator	Type	Function	SID	Status/InstId
	SRH-mode	Protection	Interface	
PE-2_loc				
End.DT4			2 2001:db8:aaaa:102:0:2000::	ok
End.DT6			3 2001:db8:aaaa:102:0:3000::	ok
End			1 2001:db8:aaaa:102:0:1000::	ok


```

USP
-----
Auto-allocated End.X: PSP Unprotected,
-----
End.X          *4 2001:db8:aaaa:102:0:4000::          0
PSP            Unprotected int-PE-2-PE-1
ISIS Level: L2 Mac Address: 04:0a:01:01:00:01 Nbr Sys Id: 0010.0100.1001
End.X          *5 2001:db8:aaaa:102:0:5000::          0
PSP            Unprotected int-PE-2-RR-3
ISIS Level: L2 Mac Address: 04:12:01:01:00:0b Nbr Sys Id: 0010.0100.1003
-----
=====
Legend: * - System allocated

```

Verify that the tunnel from PE-1 to the remote locator has SRv6 encapsulation and similar for the tunnel from PE-2 to the remote locator. The tunnel tables on RR-3 and on CE-1 are empty.

```

[/]
A:admin@PE-1# show router tunnel-table ipv6

=====
IPv6 Tunnel Table (Router: Base)
=====
Destination                               Owner      Encap TunnelId  Pref
NextHop                                   Color
-----
2001:db8:aaaa:102::/64                    srv6-isis SRV6  524289    0
fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"      10
-----
---snip---
=====

```

Verify that the tunnel from PE-1 to the remote locator uses the “int-PE-1-PE-2” interface and similar for the tunnel from PE-2 to the remote locator, where that tunnel uses the “int-PE-2-PE-1” interface. Interface “int-PE-1-PE-2” is configured on port 1/1/c1/1:1000. The FP tunnel tables on RR-3 and on CE-1 are empty.

```

[/]
A:admin@PE-1# show router fp-tunnel-table 1 ipv6

=====
IPv6 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                               Protocol   Tunnel-ID
Lbl/SID
NextHop                                   Intf/Tunnel
Lbl/SID (backup)
NextHop (backup)
-----
2001:db8:aaaa:102::/64                    SRV6      524289
-
fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"      1/1/c1/1:1000
-----
Total Entries : 1
=====

```

Verify that data transport is possible between CE-1 and CE-2. IPv4 data flows from CE-1 to PE-1, where it is SRv6-encapsulated and forwarded via the SRv6 tunnel to PE-2. At PE-2, the data is decapsulated and is forwarded to CE-2. Between PE-1 and PE-2, the IPv4 data cannot flow unencapsulated because there is no IPv4 interface between PE-1 and PE-2.

```
[/]  
A:admin@CE-1# ping 192.0.2.5  
PING 192.0.2.5 56 data bytes  
64 bytes from 192.0.2.5: icmp_seq=1 ttl=62 time=2.13ms.  
---snip---  
---- 192.0.2.5 PING Statistics ----  
5 packets transmitted, 5 packets received, 0.00% packet loss  
round-trip min = 1.76ms, avg = 1.98ms, max = 2.35ms, stddev = 0.225ms
```

```
[/]  
A:admin@CE-1# traceroute 192.0.2.5  
traceroute to 192.0.2.5, 30 hops max, 40 byte packets  
 1 172.16.14.1 (172.16.14.1) 0.783 ms 0.864 ms 0.893 ms  
 2 0.0.0.0 * * *  
 3 192.0.2.5 (192.0.2.5) 2.07 ms 1.83 ms 1.87 ms
```

IPv6 data flows from CE-1 to PE-1, where it is SRv6 encapsulated and forwarded via the SRv6 tunnel to PE-2. At PE-2, the data is decapsulated and is forwarded to CE-2. The IPv6 data does not flow with native IPv6 between PE-1 and PE-2 because then it would use the 2001:db8::168:12:1 IPv6 interface instead of the 2001:db8::2:2 IPv6 system address in the second hop. The same is true for data transport between CE-2 and CE-1.

```
[/]  
A:admin@CE-1# ping 2001:db8::2:5  
PING 2001:db8::2:5 56 data bytes  
64 bytes from 2001:db8::2:5 icmp_seq=1 hlim=62 time=1.68ms.  
---snip---  
---- 2001:db8::2:5 PING Statistics ----  
5 packets transmitted, 5 packets received, 0.00% packet loss  
round-trip min = 1.68ms, avg = 1.75ms, max = 1.83ms, stddev = 0.056ms
```

```
[/]  
A:admin@CE-1# traceroute 2001:db8::2:5  
traceroute to 2001:db8::2:5, 30 hops max, 60 byte packets  
 1 2001:db8::168:14:1 (2001:db8::168:14:1) 0.835 ms 0.915 ms 0.809 ms  
 2 2001:db8::2:2 (2001:db8::2:2) 1.54 ms 1.59 ms 1.51 ms  
 3 2001:db8::2:5 (2001:db8::2:5) 1.88 ms 1.73 ms 2.00 ms
```

Conclusion

SRv6 Encapsulation in the base routing instance can be used to transport native IPv4 and IPv6 data across an SRv6-enabled provider network.

SRv6 Loop-Free Alternate

This chapter provides information about loop-free alternate for segment routing over IPv6 .

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information and configuration in this chapter are based on SR OS Release 22.2.R1. Segment routing over IPv6 (SRv6) is supported on FP4-based equipment in SR OS Release 21.5.R2 and later.

Overview

SR OS Release 21.5.R2 and later support loop-free alternate (LFA) for segment routing over IPv6 (SRv6). This includes regular LFA, remote LFA (R-LFA) and topology independent LFA (TI-LFA) for routers in a service originating role and for routers in a transit role, with or without segment termination.

The local router installs its locator prefix in its IPv6 route table and IPv6 forwarding information base (FIB), and advertises its locator prefix in IS-IS with the SRv6 locator sub-TLV. Each remote router populates its IPv6 route table and IPv6 FIB with the received locator prefixes, including the tunneled next hop to the originating router.

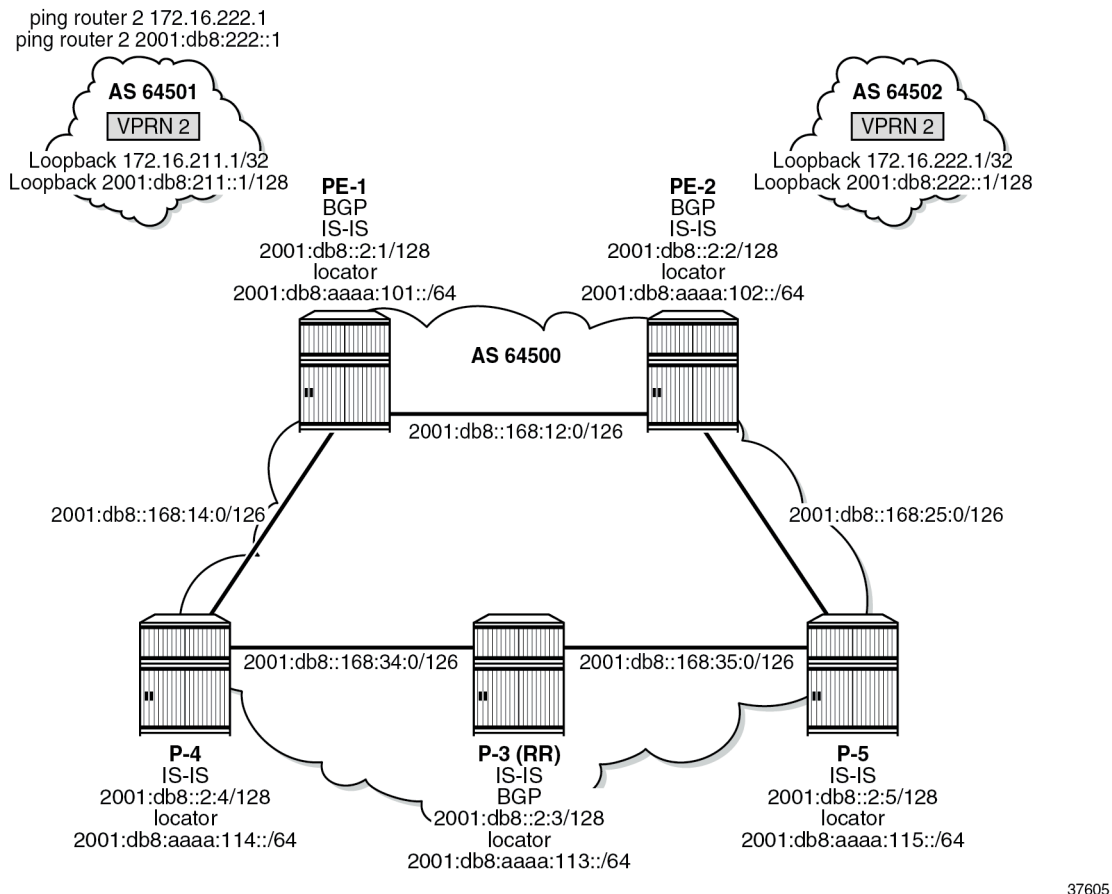
The LFA backup path for a local End.X segment identifier (SID) or a local LAN End.X SID is programmed in the IPv6 route table and in the IPv6 FIB with the specific entry corresponding to the local locator prefix.

The LFA backup path for a remote locator prefix entry is programmed in the IPv6 route table and in the IPv6 FIB. The LFA backup path for a remote End SID, End.DT4 SID, End.DT6 SID, or End.DX2 SID uses that remote locator prefix.

Configuration

[Figure 373: Example topology](#) shows the example topology with five SRv6-capable routers. The SRv6-enabled network that it represents comprises PE-1, PE-2, and P-3 in the control and data planes, and P-4 and P-5 in the data plane only. The SRv6-enabled network has only IPv6 addresses and interfaces.

Figure 373: Example topology



37605

For the transport of IPv4 and IPv6 data from the VPRN on PE-1 to the VPRN on PE-2, PE-1 acts as the SRv6 ingress PE node, while PE-2 acts as the SRv6 egress PE node.

SRv6 and forwarding path extension (FPE) are configured on all routers. P-3 acts as the BGP route reflector in the control plane. As long as the link between PE-1 and PE-2 is operational, P-3 does not participate in the SRv6 data transport between PE-1 and PE-2. When the link between PE-1 and PE-2 fails, SRv6 data transport uses an LFA backup path via P-3.

The **ping** and **traceroute** commands between IPv4 and IPv6 loopback addresses in the VPRNs simulate data transport.

SRv6 for VPRN is established between PE-1 and PE-2, as described in the Segment Routing over IPv6 for VPRN chapter. The metric on all links, except one, is set to 10. When the metric on the link between PE-2 and P-5 is set to 21, configuring TI-LFA on PE-1 for all destination prefixes using the protected link PE-1–PE-2, results in a PQ-router P-5. In this case, the End SID of P-5 suffices. When the metric on the link between P-3 and P-5 is set to 21, configuring TI-LFA on PE-1 for all destination prefixes using the protected link PE-1-PE-2, results in disjointed P-router P-3 and Q-router P-5. In that case, the End.X SID referencing the interface on P-3 facing P-5 suffices to reach the Q node.

Configure the router

This configuration includes:

- ports and IPv6-only interfaces on all routers
- port cross-connect (PXC) on all routers, using internal loopbacks on an FP4 MAC chip, as described in the Segment Routing over IPv6 chapter
- IS-IS on all routers, including:
 - level 2 capability with wide metrics (for the 128-bit identifiers)
 - level 2 metric is 10 on all IS-IS interfaces, but 21 on the IS-IS interface between PE-2 and P-5
 - native IPv6 routing
 - as a best practice to advertise the router capability within the autonomous system (AS), also configure:
 - **traffic-engineering**
 - **traffic-engineering-options**
- BGP on PE-1, PE-2, and P-3, with internal group “gr_v6_internal” that includes:
 - IPv4, IPv6, VPN-IPv4 and VPN-IPv6 families
 - **extended-nh-encoding** for IPv4 and VPN-IPv4
 - **advertise-ipv6-next-hops** for IPv4, VPN-IPv4 and VPN-IPv6
 - BGP neighbor **system** IPv6 addresses
 - On PE-1 and PE-2 only: **next-hop-self**

The core network topology uses IPv6 for BGP peering (with 16 byte next hop addresses), so to advertise and receive IPv4 routes (which have 4 byte next hop addresses) with IPv6 next hop addresses, the commands **advertise-ipv6-next-hops** and **extended-nh-encoding** need to be configured at the BGP, group, or neighbor level. The **advertise-ipv6-next-hops** command instructs the system to advertise IPv4 routes with IPv6 next hop addresses. The **extended-nh-encoding** command configures BGP to advertise the capability to receive IPv4 routes with IPv6 next hop addresses.

The following example configuration applies for PE-1 and is similar for the other routers, with the following differences:

- P-3 acts as a BGP route reflector
- BGP is not configured on P-4 and P-5

```
[/]
A:admin@PE-1# configure {
  router "Base" {
    autonomous-system 64500
    interface "int-PE-1-P-4" {
      description "interface between PE-1 and P-4"
      port 1/1/c2/1:1000
      ipv6 {
        address 2001:db8::168:14:1 {
          prefix-length 126
        }
      }
    }
    interface "int-PE-1-PE-2" {
      description "interface between PE-1 and PE-2"
      port 1/1/c1/1:1000
    }
  }
}
```

```
    ipv6 {
      address 2001:db8::168:12:1 {
        prefix-length 126
      }
    }
  }
  interface "system" {
    description "system interface of PE-1"
    ipv6 {
      address 2001:db8::2:1 {
        prefix-length 128
      }
    }
  }
  isis 0 {
    admin-state enable
    advertise-router-capability as
    ipv6-routing native
    level-capability 2 # required for SRv6
    router-id 1.1.1.1 # must be unique and in the format of an IPv4 address
    traffic-engineering true
    area-address [49.0001]
    traffic-engineering-options {
      ipv6 true
      application-link-attributes {
      }
    }
    interface "int-PE-1-P-4" {
      interface-type point-to-point
      level 1 {
        metric 10
      }
      level 2 {
        metric 10
      }
    }
    interface "int-PE-1-PE-2" {
      interface-type point-to-point
      level 1 {
        metric 10
      }
      level 2 {
        metric 10
      }
    }
    interface "system" {
      passive true
    }
    level 2 {
      wide-metrics-only true # required for SRv6
    }
  }
  bgp {
    min-route-advertisement 1
    router-id 2.2.2.1 # must be unique and in the format of an IPv4 address
    rapid-withdrawal true
    split-horizon true
    ebgp-default-reject-policy {
      import false
      export false
    }
    rapid-update {
      vpn-ipv4 true
      vpn-ipv6 true
    }
  }
}
```

```

    }
    group "gr_v6_internal" {
      description "internal bgp group on PE-1"
      next-hop-self true
      type internal
      family {
        ipv4 true
        vpn-ipv4 true
        ipv6 true
        vpn-ipv6 true
      }
      extended-nh-encoding {
        vpn-ipv4 true
        ipv4 true
      }
      advertise-ipv6-next-hops {
        vpn-ipv6 true
        vpn-ipv4 true
        ipv4 true
      }
    }
  }
  neighbor "2001:db8::2:3" { # P-3 system address
    group "gr_v6_internal"
  }
}
exit all

```



Note:

Do not advertise tunnel links, because that enables forwarding adjacencies. IS-IS does not compute a remote LFA or a TI-LFA backup for an SR-ISIS tunnel when forwarding adjacency (configured via the **advertise-tunnel-links** command) is enabled in the IS-IS instance, even if these two types of LFAs are enabled in the configuration of that same IS-IS instance.

Configure the VPRN services on PE-1 and on PE-2

This configuration includes:

- an IPv4 address and an IPv6 address for a loopback interface "lb_if_vprn"
- BGP, with external group "gr_v6_vprn" that includes the following capabilities:
 - IPv4 and IPv6 families
 - **extended-nh-encoding** for IPv4
 - **advertise-ipv6-next-hops** for IPv4
 - BGP neighbor **interface** IPv6 addresses, with BGP neighbors in a different external AS

The following example configuration applies for VPRN 2 on PE-1 and is similar for VPRN 2 on PE-2.

```

[/]
A:admin@PE-1# configure {
  service {
    vprn "VPRN_2" {
      admin-state enable
      description "VPRN 2 on PE-1"
      service-id 2
      customer "1"
      autonomous-system 64500
      bgp {
        ebgp-default-reject-policy {

```

```

import false
export false
}
group "gr_v6_vprn" {
description "external bgp group for VPRN 2 on PE-1"
family {
ipv4 true
ipv6 true
}
extended-nh-encoding {
ipv4 true
}
advertise-ipv6-next-hops {
ipv4 true
}
}
neighbor "2001:db8:101::1" {
group "gr_v6_vprn"
type external
peer-as 64501
}
}
interface "lb_itf_vprn" {
description "VPRN 2 interface on PE-1 for external subnet"
loopback true
ipv4 {
primary {
address 172.16.211.1
prefix-length 32
}
}
ipv6 {
address 2001:db8:211::1 {
prefix-length 128
}
}
}
}
exit all

```

Configure SRv6 in the router Base context on all routers

Configure the locator in the **router Base segment-routing segment-routing-v6** context on PE-2 and similar on the other routers, with different **ip-prefix** for the locators.

```

[/]
A:admin@PE-2# configure {
router "Base" {
segment-routing {
segment-routing-v6 {
locator "PE-2_loc" {
admin-state enable
block-length 48
prefix {
ip-prefix 2001:db8:aaaa:102::/64
}
}
}
}
}
exit all

```


Configure the FPEs on PE-2 and identical on the other routers.

```
[/]
A:admin@PE-2# configure {
  fwd-path-ext {
    fpe 1 {
      path {
        pxc 1
      }
      application {
        srv6 {
          type origination
        }
      }
    }
    fpe 2 {
      path {
        pxc 2
      }
      application {
        srv6 {
          type termination
        }
      }
    }
  }
}
exit all
```

Use FPE 1 as the SRv6 origination FPE in the **router Base segment-routing segment-routing-v6** context and FPE 2 as the SRv6 termination FPE in the **router Base segment-routing segment-routing-v6 locator** context on PE-2. The configuration is similar on the other routers, with different locators. For more information, see the [Segment Routing over IPv6](#) chapter.

```
[/]
A:admin@PE-2# configure {
  router "Base" {
    segment-routing {
      segment-routing-v6 {
        origination-fpe [1]
        locator "PE-2_loc" {
          admin-state enable
          termination-fpe [2]
        }
      }
    }
  }
}
exit all
```

Configure the SRv6 End function (equivalent to an IPv4 node SID) and SRv6 End.X functions (equivalent to IPv4 adjacency SIDs) in the **router Base segment-routing segment-routing-v6 base-routing-instance locator** context on all routers, with different locators.

```
[/]
A:admin@PE-2# configure {
  router "Base" {
    mpls-labels {
      sr-labels {
        start 20000
        end 20999
      }
      reserved-label-block "SRv6" {
        start-label 30100
        end-label 30199
      }
    }
  }
}
```

```

}
segment-routing {
  segment-routing-v6 {
    locator "PE-2_loc" {
      static-function {
        max-entries 3
        label-block "SRv6"
      }
    }
  }
  base-routing-instance {
    locator "PE-2_loc" {
      function {
        end 1 {
          srh-mode usp
        }
        end-x-auto-allocate usp protection protected { }
      }
    }
  }
}
}
exit all

```

While not strictly needed, allow for three static functions. New SRv6 functions (for example End.DT4 and End.DT6), can then be configured without needing to reshuffle the automatic SRv6 function numbering. Ensure that the End.X functions have protection on. As a result, the End.X functions are only instantiated when **loopfree-alternate** is configured in the **router Base isis** context.

Advertise the locator in IS-IS while ensuring level 2 capability on PE-2. Configure other routers similarly, with different locators.

```

[/]
A:admin@PE-2# configure {
  router "Base" {
    isis 0 {
      segment-routing-v6 {
        admin-state enable
        locator "PE-2_loc" {
          level-capability 2
        }
      }
    }
  }
}
exit all

```

Use the **show router segment-routing-v6 summary** command to verify the locator and origination FPE configuration.

Configure SRv6 for the VPRNs on PE-1 and on PE-2

Create an SRv6 instance for the VPRN service. Use the locator from the **router Base segment-routing segment-routing-v6** context and configure End.DT4 and End.DT6 functions for it.

Use the created SRv6 instance in the **service vprn bgp-ipvpn segment-routing-v6** context, with the configured locator as the default locator. Ensure a unique route distinguisher. Use the unique PE-2 system IPv6 address as the source address. Use a similar configuration on PE-1, with the PE-1 locator as default locator, the PE-1 system IPv6 address as the source address, and a different route distinguisher.

```

[/]
A:admin@PE-2# configure {
  service {

```

```

vprn "VPRN_2" {
  segment-routing-v6 1 {
    locator "PE-2_loc" {
      function {
        end-dt4 {
        }
        end-dt6 {
        }
      }
    }
  }
}
bgp-ipvpn {
  segment-routing-v6 1 {
    admin-state enable
    route-distinguisher "192.0.2.2:2"
    source-address 2001:db8::2:2
    vrf-target {
      community "target:64506:2"
    }
    srv6 {
      instance 1
      default-locator "PE-2_loc"
    }
  }
}
exit all

```

This configuration results in BGP update exchanges between PE-2 and PE-1, via P-3, and between PE-1 and PE-2, via P-3.

At this point, verify that data transport is possible between the local VPRN on PE-1 and the remote VPRN on PE-2.

```

[/]
A:admin@PE-1# ping 172.16.222.1 router-instance "VPRN_2"
PING 172.16.222.1 56 data bytes
64 bytes from 172.16.222.1: icmp_seq=1 ttl=64 time=1.85ms.
---snip---
---- 172.16.222.1 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 1.74ms, avg = 1.82ms, max = 1.93ms, stddev = 0.065ms

```

```

[/]
A:admin@PE-1# traceroute 172.16.222.1 router-instance "VPRN_2"
traceroute to 172.16.222.1, 30 hops max, 40 byte packets
 1 172.16.222.1 (172.16.222.1) 2.18 ms 1.71 ms 1.76 ms

```

```

[/]
A:admin@PE-1# ping 2001:db8:222::1 router-instance "VPRN_2"
PING 2001:db8:222::1 56 data bytes
64 bytes from 2001:db8:222::1 icmp_seq=1 hlim=64 time=1.54ms.
---snip---
---- 2001:db8:222::1 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 1.54ms, avg = 1.60ms, max = 1.73ms, stddev = 0.072ms

```

```

[/]
A:admin@PE-1# traceroute 2001:db8:222::1 router-instance "VPRN_2"
traceroute to 2001:db8:222::1, 30 hops max, 60 byte packets
 1 2001:db8:222::1 (2001:db8:222::1) 1.83 ms 1.80 ms 1.74 ms

```

The result of the verification complies with the route tables for the local VPRN on PE-1, which contains routes for the loopback addresses in the remote VPRN on PE-2. The same is true for data transport between the remote VPRN on PE-2 and the local VPRN on PE-1.

```
[/]
A:admin@PE-1# show router 2 route-table ipv4

=====
Route Table (Service: 2)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]                Metric
-----
172.16.211.1/32                    Local  Local   00h21m39s    0
  lb_itf_vprn                        0
172.16.222.1/32                    Remote BGP VPN 00h02m28s    170
  2001:db8:aaaa:102:78a6:c000:: (tunneled:SRV6) 10
-----
No. of Routes: 2
---snip---
```

```
[/]
A:admin@PE-1# show router 2 route-table ipv6

=====
IPv6 Route Table (Service: 2)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]                Metric
-----
2001:db8:211::1/128                Local  Local   00h21m37s    0
  lb_itf_vprn                        0
2001:db8:222::1/128                Remote BGP VPN 00h02m28s    170
  2001:db8:aaaa:102:78a6:b000:: (tunneled:SRV6) 10
-----
No. of Routes: 2
---snip---
```

The IPv4 route table and IPv4 FIB remain empty, while the IPv6 route table and IPv6 FIB contain routes for the local, IS-IS, and SRv6 protocols. The remote destinations to PE-2 and P-5 are reached via the “int-PE-1-PE-2” interface. There are no routes yet for the local End.X functions. The local End.X functions are not yet instantiated, because there is no regular LFA protection while protection is enabled for End.X function. Verify the IPv6 route table.

```
[/]
A:admin@PE-1# show router route-table ipv6

=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]                Metric
-----
2001:db8::2:1/128                  Local  Local   00h31m23s    0
  system                            0
2001:db8::2:2/128                  Remote ISIS 00h08m15s    18
  fe80::60e:1ff:fe01:1-"int-PE-1-PE-2" 10
2001:db8::2:3/128                  Remote ISIS 00h08m15s    18
```

```

                fe80::616:1ff:fe01:1-"int-PE-1-P-4"                20
2001:db8::2:4/128 Remote ISIS 00h08m15s 18
                fe80::616:1ff:fe01:1-"int-PE-1-P-4"                10
2001:db8::2:5/128 Remote ISIS 00h08m15s 18
                fe80::616:1ff:fe01:1-"int-PE-1-P-4"                30
2001:db8::168:12:0/126 Local Local 00h31m22s 0
                int-PE-1-PE-2 0
2001:db8::168:14:0/126 Local Local 00h31m22s 0
                int-PE-1-P-4 0
2001:db8::168:25:0/126 Remote ISIS 00h08m15s 18
                fe80::60e:1ff:fe01:1-"int-PE-1-PE-2" 31
2001:db8::168:34:0/126 Remote ISIS 00h08m15s 18
                fe80::616:1ff:fe01:1-"int-PE-1-P-4"                20
2001:db8::168:35:0/126 Remote ISIS 00h08m15s 18
                fe80::616:1ff:fe01:1-"int-PE-1-P-4"                30
2001:db8:aaaa:101::/64 Local SRV6 00h10m40s 3
                fe80::201-"_tmnx_fpe_2.a" 0
2001:db8:aaaa:101:0:1000::/128 Local SRV6 00h10m40s 3
                Black Hole 0
2001:db8:aaaa:102::/64 Remote ISIS 00h08m01s 18
                2001:db8:aaaa:102::/64 (tunneled:SRV6-ISIS) 10
2001:db8:aaaa:113::/64 Remote ISIS 00h07m51s 18
                2001:db8:aaaa:113::/64 (tunneled:SRV6-ISIS) 20
2001:db8:aaaa:114::/64 Remote ISIS 00h07m30s 18
                2001:db8:aaaa:114::/64 (tunneled:SRV6-ISIS) 10
2001:db8:aaaa:115::/64 Remote ISIS 00h07m08s 18
                2001:db8:aaaa:115::/64 (tunneled:SRV6-ISIS) 30
-----
No. of Routes: 16
---snip---
=====

```

Verify the corresponding IPv6 FIB.

```

[/]
A:admin@PE-1# show router fib 1 ipv6

=====
FIB Display
=====
Prefix [Flags]
NextHop
-----
2001:db8::2:1/128 LOCAL
  2001:db8::2:1 (system)
2001:db8::2:2/128 ISIS
  fe80::60e:1ff:fe01:1 (int-PE-1-PE-2)
2001:db8::2:3/128 ISIS
  fe80::616:1ff:fe01:1 (int-PE-1-P-4)
2001:db8::2:4/128 ISIS
  fe80::616:1ff:fe01:1 (int-PE-1-P-4)
2001:db8::2:5/128 ISIS
  fe80::616:1ff:fe01:1 (int-PE-1-P-4)
2001:db8::168:12:0/126 LOCAL
  2001:db8::168:12:0 (int-PE-1-PE-2)
2001:db8::168:14:0/126 LOCAL
  2001:db8::168:14:0 (int-PE-1-P-4)
2001:db8::168:25:0/126 ISIS
  fe80::60e:1ff:fe01:1 (int-PE-1-PE-2)
2001:db8::168:34:0/126 ISIS
  fe80::616:1ff:fe01:1 (int-PE-1-P-4)
2001:db8::168:35:0/126 ISIS
  fe80::616:1ff:fe01:1 (int-PE-1-P-4)

```

```

2001:db8:aaaa:101::/64                               SRV6
  fe80::201 (_tmnx_fpe_2.a)
2001:db8:aaaa:101:0:1000::/128                     SRV6
  Blackhole
2001:db8:aaaa:102::/64                               ISIS
  2001:db8:aaaa:102::/64 (Transport:SRV6:524289)
2001:db8:aaaa:113::/64                               ISIS
  2001:db8:aaaa:113::/64 (Transport:SRV6:524290)
2001:db8:aaaa:114::/64                               ISIS
  2001:db8:aaaa:114::/64 (Transport:SRV6:524291)
2001:db8:aaaa:115::/64                               ISIS
  2001:db8:aaaa:115::/64 (Transport:SRV6:524292)
-----
Total Entries : 16
-----
=====

```

The IS-IS data base contains the following information. Only the End functions are already instantiated, on their respective locators.

```

[/]
A:admin@PE-1# show router isis database detail

=====
Rtr Base ISIS Instance 0 Database (detail)
=====

Displaying Level 1 database
-----
Level (1) LSP Count : 0

Displaying Level 2 database
-----
LSP ID   : PE-1.00-00                               Level   : L2
---snip---
SYS ID   : 0010.0100.1001          SysID Len : 6          Used Len : 368

TLVs :
---snip---
Router Cap : 1.1.1.1, D:0, S:0
  TE Node Cap : B E M P
  SRV6 Cap: 0x0000
  SR Alg: metric based SPF
  Node MSD Cap: BMI : 0 SRH-MAX-SL : 10 SRH-MAX-END-POP : 9 SRH-MAX-H-ENCAPS : 1 SRH-MAX-END-
D : 9
---snip---
TE IS Nbrs :
  Nbr   : PE-2.00
  Default Metric : 10
---snip---
TE IS Nbrs :
  Nbr   : P-4.00
  Default Metric : 10
---snip---
IPv6 Reach:
---snip---
Metric: ( I ) 0
Prefix  : 2001:db8:aaaa:101::/64
SRv6 Locator :
  MT ID : 0
  Metric: ( ) 0 Algo:0
  Prefix  : 2001:db8:aaaa:101::/64
  Sub TLV :

```

```
End-SID : 2001:db8:aaaa:101:0:1000::, flags:0x0, endpoint:End-USP
-----
LSP ID : PE-2.00-00 Level : L2
---snip---
SYS ID : 0010.0100.1002 SysID Len : 6 Used Len : 368

TLVs :
---snip---
Router Cap : 1.1.1.2, D:0, S:0
TE Node Cap : B E M P
SRv6 Cap: 0x0000
SR Alg: metric based SPF
Node MSD Cap: BMI : 0 SRH-MAX-SL : 10 SRH-MAX-END-POP : 9 SRH-MAX-H-ENCAPS : 1 SRH-MAX-END-
D : 9
---snip---
TE IS Nbrs :
Nbr : PE-1.00
Default Metric : 10
---snip---
TE IS Nbrs :
Nbr : P-5.00
Default Metric : 21
---snip---
IPV6 Reach:
---snip---
Metric: ( I ) 21
Prefix : 2001:db8::168:25:0/126
Metric: ( I ) 0
Prefix : 2001:db8:aaaa:102::/64
SRv6 Locator :
MT ID : 0
Metric: ( ) 0 Algo:0
Prefix : 2001:db8:aaaa:102::/64
Sub TLV :
End-SID : 2001:db8:aaaa:102:0:1000::, flags:0x0, endpoint:End-USP
-----
LSP ID : P-3.00-00 Level : L2
---snip---
SYS ID : 0010.0100.1003 SysID Len : 6 Used Len : 367

TLVs :
---snip---
Router Cap : 1.1.1.3, D:0, S:0
TE Node Cap : B E M P
SRv6 Cap: 0x0000
SR Alg: metric based SPF
Node MSD Cap: BMI : 0 SRH-MAX-SL : 10 SRH-MAX-END-POP : 9 SRH-MAX-H-ENCAPS : 1 SRH-MAX-END-
D : 9
---snip---
TE IS Nbrs :
Nbr : P-4.00
Default Metric : 10
---snip---
TE IS Nbrs :
Nbr : P-5.00
Default Metric : 10
---snip---
IPV6 Reach:
---snip---
Metric: ( I ) 0
Prefix : 2001:db8:aaaa:113::/64
SRv6 Locator :
```

```
MT ID : 0
Metric: ( ) 0 Algo:0
Prefix : 2001:db8:aaaa:113::/64
Sub TLV :
  End-SID : 2001:db8:aaaa:113:0:1000::, flags:0x0, endpoint:End-USP
-----
LSP ID : P-4.00-00                               Level : L2
---snip---
SYS ID : 0010.0100.1004          SysID Len : 6          Used Len : 367
TLVs :
---snip---
Router Cap : 1.1.1.4, D:0, S:0
TE Node Cap : B E M P
SRv6 Cap: 0x0000
SR Alg: metric based SPF
Node MSD Cap: BMI : 0 SRH-MAX-SL : 10 SRH-MAX-END-POP : 9 SRH-MAX-H-ENCAPS : 1 SRH-MAX-END-
D : 9
---snip---
TE IS Nbrs :
  Nbr : P-3.00
  Default Metric : 10
---snip---
TE IS Nbrs :
  Nbr : PE-1.00
  Default Metric : 10
---snip---
IPv6 Reach:
---snip---
Metric: ( I ) 0
Prefix : 2001:db8:aaaa:114::/64
SRv6 Locator :
  MT ID : 0
  Metric: ( ) 0 Algo:0
  Prefix : 2001:db8:aaaa:114::/64
  Sub TLV :
    End-SID : 2001:db8:aaaa:114:0:1000::, flags:0x0, endpoint:End-USP
-----
LSP ID : P-5.00-00                               Level : L2
---snip---
SYS ID : 0010.0100.1005          SysID Len : 6          Used Len : 367
TLVs :
---snip---
Router Cap : 1.1.1.5, D:0, S:0
TE Node Cap : B E M P
SRv6 Cap: 0x0000
SR Alg: metric based SPF
Node MSD Cap: BMI : 0 SRH-MAX-SL : 10 SRH-MAX-END-POP : 9 SRH-MAX-H-ENCAPS : 1 SRH-MAX-END-
D : 9
---snip---
TE IS Nbrs :
  Nbr : P-3.00
  Default Metric : 10
---snip---
TE IS Nbrs :
  Nbr : PE-2.00
  Default Metric : 21
---snip---
IPv6 Reach:
---snip---
Metric: ( I ) 21
```



```

Prefix : 2001:db8::168:25:0/126
---snip---
Metric: ( I ) 0
Prefix : 2001:db8:aaaa:115::/64
SRv6 Locator :
MT ID : 0
Metric: ( ) 0 Algo:0
Prefix : 2001:db8:aaaa:115::/64
Sub TLV :
End-SID : 2001:db8:aaaa:115:0:1000::, flags:0x0, endpoint:End-USP

Level (2) LSP Count : 5
-----
---snip---
=====

```

Verify the SRv6 local SIDs and SRv6 base routing instances on PE-1 and similar on PE-2. The End.X functions are not yet instantiated.

```

[/]
A:admin@PE-1# show router segment-routing-v6 local-sid

=====
Segment Routing v6 Local SIDs
=====
SID                               Type           Function
Locator
Context
-----
2001:db8:aaaa:101:0:1000::        End            1
PE-1_loc
Base
2001:db8:aaaa:101:78a6:b000::     End.DT6        494187
PE-1_loc
SvcId: 2 Name: VPRN_2
2001:db8:aaaa:101:78a6:c000::     End.DT4        494188
PE-1_loc
SvcId: 2 Name: VPRN_2
-----
SIDs : 3
-----
=====

```

The End.X functions not yet being instantiated can also be verified in the SRv6 base routing instances on PE-1 and similar on PE-2. Only the End function is already instantiated.

```

[/]
A:admin@PE-1# show router segment-routing-v6 base-routing-instance

=====
Segment Routing v6 Base Routing Instance
=====
Locator                               Type           Function      SID                               Status/InstId
Type           SRH-mode Protection Interface
-----
PE-1_loc
End            1 2001:db8:aaaa:101:0:1000::      ok
USP
-----
Auto-allocated End.X: USP Protected,
-----

```

```
-----
Legend: * - System allocated
=====
```

Verify that the tunnels have SRv6 encapsulation.

```
[/]
A:admin@PE-1# show router tunnel-table ipv6

=====
IPv6 Tunnel Table (Router: Base)
=====
Destination                               Owner      Encap TunnelId  Pref
NextHop                                   Color
-----
2001:db8:aaaa:102::/64                    srv6-isis SRV6 524289    0
    fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"    10
2001:db8:aaaa:113::/64                    srv6-isis SRV6 524290    0
    fe80::616:1ff:fe01:1-"int-PE-1-P-4"     20
2001:db8:aaaa:114::/64                    srv6-isis SRV6 524291    0
    fe80::616:1ff:fe01:1-"int-PE-1-P-4"     10
2001:db8:aaaa:115::/64                    srv6-isis SRV6 524292    0
    fe80::616:1ff:fe01:1-"int-PE-1-P-4"     30
-----
---snip---
```

Verify the interfaces that the tunnels are using. Interface "int-PE-1-PE-2" is configured on port 1/1/c1/1:1000. Interface "int-PE-1-P-4" is configured on port 1/1/c2/1:1000. SRv6 data is transported to PE-2 over the link between PE-1 and PE-2, via next hop fe80::60e:1ff:fe01:1-"int-PE-1-PE-2".

```
[/]
A:admin@PE-1# show router fp-tunnel-table 1 ipv6

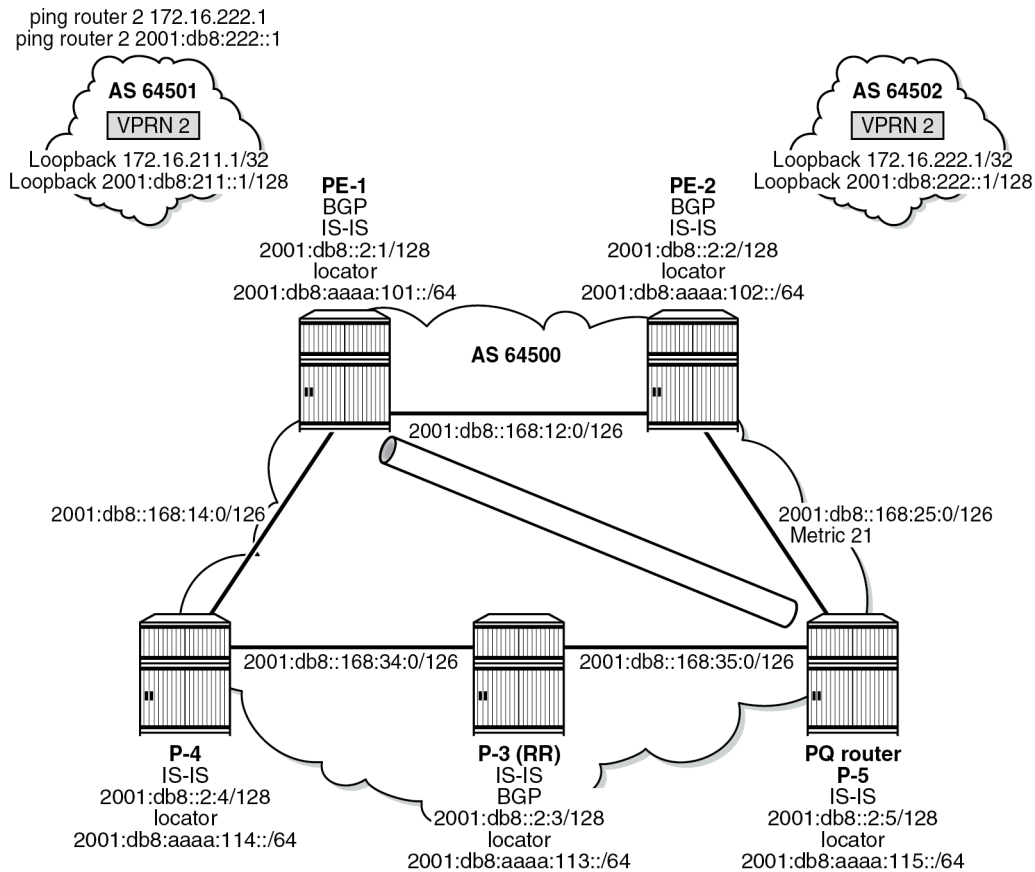
=====
IPv6 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                               Protocol   Tunnel-ID
Lbl/SID
NextHop                                   Intf/Tunnel
Lbl/SID (backup)
NextHop (backup)
-----
2001:db8:aaaa:102::/64                    SRV6      524289
-
    fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"    1/1/c1/1:1000
2001:db8:aaaa:113::/64                    SRV6      524290
-
    fe80::616:1ff:fe01:1-"int-PE-1-P-4"    1/1/c2/1:1000
2001:db8:aaaa:114::/64                    SRV6      524291
-
    fe80::616:1ff:fe01:1-"int-PE-1-P-4"    1/1/c2/1:1000
2001:db8:aaaa:115::/64                    SRV6      524292
-
    fe80::616:1ff:fe01:1-"int-PE-1-P-4"    1/1/c2/1:1000
-----
Total Entries : 4
=====
```

Configure LFA on PE-1

Figure 2 shows the example topology with initial metrics that is used to verify the behavior when a PQ-router provides TI-LFA protection.

Figure 374: Example topology with metric 21 between PE-2 and P-5



37606

Configure regular LFA:

```
[/]
A:admin@PE-1# configure {
  router "Base" {
    isis 0 {
      loopfree-alternate {
      }
    }
  }
  exit all
}
```

Verify the IPv6 route table. There are two additional routes, corresponding with the End.X functions for locator "PE-1_loc" that are instantiated. The existing route to P-5 is loop-protected with regular LFA.

```
[/]
A:admin@PE-1# show router route-table ipv6

=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]
Type Proto Age Metric Pref
-----
---snip---
2001:db8::2:5/128 [L] Remote ISIS 00h10m43s 18
fe80::616:1ff:fe01:1-"int-PE-1-P-4" 30
---snip---
2001:db8:aaaa:101:78a6:d000::/128 Local ISIS 00h00m44s 18
2001:db8:aaaa:101:78a6:d000:: (tunneled:SRV6-ISIS) 10
2001:db8:aaaa:101:78a6:e000::/128 Local ISIS 00h00m44s 18
2001:db8:aaaa:101:78a6:e000:: (tunneled:SRV6-ISIS) 10
---snip---
-----
No. of Routes: 18
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      ---snip---
=====
```

Verify the corresponding IPv6 FIB.

```
[/]
A:admin@PE-1# show router fib 1 ipv6

=====
FIB Display
=====
Prefix [Flags] Protocol
NextHop
-----
---snip---
2001:db8::2:5/128 ISIS
fe80::616:1ff:fe01:1 (int-PE-1-P-4)
---snip---
2001:db8:aaaa:101:78a6:d000::/128 ISIS
2001:db8:aaaa:101:78a6:d000:: (Transport:SRV6:524293)
2001:db8:aaaa:101:78a6:e000::/128 ISIS
2001:db8:aaaa:101:78a6:e000:: (Transport:SRV6:524294)
---snip---
-----
Total Entries : 18
-----
=====
```

The IS-IS database contains additional information about the End.X functions that are instantiated on PE-1. The End.X functions for the locator "PE-1_loc" are instantiated and advertised. There are no changes for the other routers.

```
[/]
A:admin@PE-1# show router isis database detail
```

```

=====
Rtr Base ISIS Instance 0 Database (detail)
=====
---snip---
Displaying Level 2 database
-----
LSP ID      : PE-1.00-00                               Level      : L2
---snip---
TLVs :
---snip---
TE IS Nbrs :
  Nbr      : PE-2.00
  Default Metric : 10
---snip---
End.X-SID: 2001:db8:aaaa:101:78a6:e000:: flags:B algo:0 weight:0 endpoint:End.X-USP
TE IS Nbrs :
  Nbr      : P-4.00
  Default Metric : 10
---snip---
End.X-SID: 2001:db8:aaaa:101:78a6:d000:: flags:B algo:0 weight:0 endpoint:End.X-USP
---snip---
Level (2) LSP Count : 5
---snip---
=====

```

Verify the SRv6 local SIDs and SRv6 base routing instance on PE-1. The End.X functions are also instantiated.

```

[/]
A:admin@PE-1# show router segment-routing-v6 local-sid

=====
Segment Routing v6 Local SIDs
=====
SID                                     Type      Function
Locator
Context
-----
---snip---
2001:db8:aaaa:101:78a6:d000::         End.X    494189
PE-1_loc
None
2001:db8:aaaa:101:78a6:e000::         End.X    494190
PE-1_loc
None
-----
SIDs : 5
=====

```

The SRv6 functions are listed.

```

[/]
A:admin@PE-1# show router segment-routing-v6 base-routing-instance

=====
Segment Routing v6 Base Routing Instance
=====
Locator
Type      Function      SID
SRH-mode Protection Interface
-----

```

```

PE-1_loc
End          1 2001:db8:aaaa:101:0:1000::      ok
USP

-----
Auto-allocated End.X: USP Protected,
-----
End.X      *494189 2001:db8:aaaa:101:78a6:d000::      0
USP        Protected int-PE-1-P-4
ISIS Level: L2 Mac Address: 04:16:01:01:00:01 Nbr Sys Id: 0010.0100.1004
End.X      *494190 2001:db8:aaaa:101:78a6:e000::      0
USP        Protected int-PE-1-PE-2
ISIS Level: L2 Mac Address: 04:0e:01:01:00:01 Nbr Sys Id: 0010.0100.1002
-----
=====
Legend: * - System allocated
    
```

Verify the IPv6 tunnel table. There are two new SRv6 tunnels for the End.X functions and the existing SRv6 tunnel to P-5 is loop-protected via regular LFA.

```

[/]
A:admin@PE-1# show router tunnel-table ipv6

=====
IPv6 Tunnel Table (Router: Base)
=====
Destination                               Owner      Encap TunnelId  Pref
NextHop                                   Color      Metric
-----
2001:db8:aaaa:101:78a6:d000::/128         srv6-isis SRV6  524293    0
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"      10
2001:db8:aaaa:101:78a6:e000::/128         srv6-isis SRV6  524294    0
  fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"     10
---snip---
2001:db8:aaaa:115::/64 [L]                 srv6-isis SRV6  524292    0
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"      30
-----
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       ---snip---
=====
    
```

Verify the interfaces that the tunnels are using.

```

[/]
A:admin@PE-1# show router fp-tunnel-table 1 ipv6

=====
IPv6 Tunnel Table Display
=====
Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                               Protocol   Tunnel-ID
Lbl/SID
NextHop                                   Intf/Tunnel
Lbl/SID (backup)
NextHop (backup)
-----
---snip---
2001:db8:aaaa:115::/64                     SRV6      524292
-
    
```

```

fe80::616:1ff:fe01:1-"int-PE-1-P-4"          1/1/c2/1:1000
-
fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"(B)    1/1/c1/1:1000
2001:db8:aaaa:101:78a6:d000::/128          SRV6      524293
-
fe80::616:1ff:fe01:1-"int-PE-1-P-4"      1/1/c2/1:1000
2001:db8:aaaa:101:78a6:e000::/128          SRV6      524294
-
fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"    1/1/c1/1:1000
-----
Total Entries : 6
-----
=====

```

Configure TI-LFA:

```

[/]
A:admin@PE-1# configure {
  router "Base" {
    isis 0 {
      loopfree-alternate {
        ti-lfa {
        }
      }
    }
  }
  exit all
}

```

There are no changes to the IPv6 route table, IPv6 FIB, IS-IS database, SRv6 local SIDs, and SRv6 base routing instance, while the change in LFA computation results in LFA protection for the tunnels to the remote routers. The existing SRv6 tunnels to PE-2, P-3 and P-4 are now also loop-protected. Verify the IPv6 tunnel table.

```

[/]
A:admin@PE-1# show router tunnel-table ipv6

=====
IPv6 Tunnel Table (Router: Base)
=====

```

Destination Nexthop	Owner Color	Encap	TunnelId Metric	Pref
2001:db8:aaaa:101:78a6:d000::/128 fe80::616:1ff:fe01:1-"int-PE-1-P-4"	srv6-isis	SRV6	524293 10	0
2001:db8:aaaa:101:78a6:e000::/128 fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"	srv6-isis	SRV6	524294 10	0
2001:db8:aaaa:102::/64 [L] fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"	srv6-isis	SRV6	524289 10	0
2001:db8:aaaa:113::/64 [L] fe80::616:1ff:fe01:1-"int-PE-1-P-4"	srv6-isis	SRV6	524290 20	0
2001:db8:aaaa:114::/64 [L] fe80::616:1ff:fe01:1-"int-PE-1-P-4"	srv6-isis	SRV6	524291 10	0
2001:db8:aaaa:115::/64 [L] fe80::616:1ff:fe01:1-"int-PE-1-P-4"	srv6-isis	SRV6	524292 30	0

```

-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      ---snip---
=====

```

Verify the interfaces that the tunnels are using. When the link between PE-1 and PE-2 is operational, SRv6 data is transported to PE-2 over this link, via next hop fe80::60e:1ff:fe01:1-"int-PE-1-PE-2". When the link between PE-1 and PE-2 fails, SRv6 data is transported to PE-2 using a fast reroute (FRR)

backup link between PE-1 and P-4, via backup next hop fe80::616:1ff:fe01:1-"int-PE-1-P-4". The SRv6 data is transported to PE-2 then, via an SRv6 tunnel to the End function on P-5, as the backup SID 2001:db8:aaaa:115:0:1000:: indicates.

```
[/]
A:admin@PE-1# show router fp-tunnel-table 1 ipv6

=====
IPv6 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol    Tunnel-ID
Lbl/SID
NextHop                                    Intf/Tunnel
Lbl/SID (backup)
NextHop (backup)
-----
2001:db8:aaaa:102::/64                     SRV6       524289
-
  fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"     1/1/c1/1:1000
2001:db8:aaaa:115:0:1000::
  fe80::616:1ff:fe01:1-"int-PE-1-P-4" (B)  1/1/c2/1:1000
2001:db8:aaaa:113::/64                     SRV6       524290
-
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"     1/1/c2/1:1000
2001:db8:aaaa:115:0:1000::
  fe80::60e:1ff:fe01:1-"int-PE-1-PE-2" (B)  1/1/c1/1:1000
2001:db8:aaaa:114::/64                     SRV6       524291
-
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"     1/1/c2/1:1000
2001:db8:aaaa:115:0:1000::
  fe80::60e:1ff:fe01:1-"int-PE-1-PE-2" (B)  1/1/c1/1:1000
2001:db8:aaaa:115::/64                     SRV6       524292
-
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"     1/1/c2/1:1000
-
  fe80::60e:1ff:fe01:1-"int-PE-1-PE-2" (B)  1/1/c1/1:1000
2001:db8:aaaa:101:78a6:d000::/128          SRV6       524293
-
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"     1/1/c2/1:1000
2001:db8:aaaa:101:78a6:e000::/128          SRV6       524294
-
  fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"     1/1/c1/1:1000
-----
Total Entries : 6
=====
```

With the topology as shown in Figure 2, this behavior is described as follows:

There is no regular LFA protection for the destination prefix to PE-2 using the protected PE-1-PE-2 link, which can be understood when the regular LFA inequality is determined using a shortest-path distance (Spd) calculation:

$$Spd(N, D) < Spd(N, S) + Spd(S, D)$$

where

Spd is the shortest path distance (according to level 2 metrics)

S is the source router (PE-1)

D is the destination router (PE-2)

N is the alternate next hop router or neighboring node (P-4)

If the outcome of the calculation is true, then regular LFA protection is valid; if the outcome is false, then there is no LFA protection.

In this case the outcome is false.

$$\text{Spd}(P-4, PE-2) < \text{Spd}(P-4, PE-1) + \text{Spd}(PE-1, PE-2)$$

$$(10 + 10 + 21) < 10 + 10$$

There is TI-LFA protection for all destination prefixes using the protected PE-1-PE-2 link, which is determined using the calculation for TI-LFA.

The TI-LFA inequality for the extended P-space P' is:

$$\text{Spd}(N, Y_i) < \text{Spd}(N, S) + \text{Spd}(S, Y_i)$$

$$\text{Spd}(P-4, Y_i) < \text{Spd}(P-4, PE-1) + \text{Spd}(PE-1, Y_i)$$

where Y_i is the set of routers {P-3, P-5} that are reachable from PE-1 and its neighbor P-4 on the post-convergence path to PE-2, without traversing the link between PE-1 and PE-2.

Apply this inequality to the set of routers Y_i :

For $Y_i=P-3$, the outcome is true. So P-3 is in P' :

$$\text{Spd}(P-4, P-3) < \text{Spd}(P-4, PE-1) + \text{Spd}(PE-1, P-3)$$

$$10 < 10 + (10 + 10)$$

For $Y_i=P-5$, the outcome is true. So P-5 is in P' :

$$\text{Spd}(P-4, P-5) < \text{Spd}(P-4, PE-1) + \text{Spd}(PE-1, P-5)$$

$$(10 + 10) < 10 + (10 + 10 + 10)$$

So, the extended P-space $P' = \{P-3, P-5\}$

The TI-LFA inequality for the Q-space Q is:

$$\text{Spd}(Z_i, D) < \text{Spd}(Z_i, S) + \text{Spd}(S, D)$$

$$\text{Spd}(Z_i, PE-2) < \text{Spd}(Z_i, PE-1) + \text{Spd}(PE-1, PE-2)$$

where Z_i is the set of routers {P-3, P-5} that are reachable from PE-2 using reverse SPF on the post-convergence path to PE-1 without traversing the link between PE-1 and PE-2.

Apply this inequality to the set of routers Z_i :

For $Z_i=P-3$, the outcome is false. So P-3 is **not** in Q :

$Spd(P-3, PE-2) < Spd(P-3, PE-1) + Spd(PE-1, PE-2)$

$(10 + 21) < (10 + 10) + 10$

For $Z_i=P-5$, the outcome is true. So P-5 is in Q:

$Spd(P-5, PE-2) < Spd(P-5, PE-1) + Spd(PE-1, PE-2)$

$21 < (10 + 10 + 10) + 10$

So, the Q-space $Q = \{P-5\}$

So, the link between PE-1 and PE-2 is TI-LFA protected with the PQ-router P-5 that belongs to the intersection of the extended P space P' and the Q space.

SRv6 data is transported to P-4, P-3, and P-5 over the link between PE-1 and P-4, via next hop `fe80::616:1ff:fe01:1-"int-PE-1-P-4"`. When the link between PE-1 and P-4 fails, SRv6 data is transported to P-4, P-3, and P-5 using a FRR backup link between PE-1 and PE-2, via backup next hop `fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"`. The SRv6 data is transported to P-4 and P-3 then via an SRv6 tunnel to the End function on P-5, as the backup SID `2001:db8:aaaa:115:0:1000::` indicates. The SRv6 data is transported to P-5 then without using an SRv6 tunnel, as the absence of a backup SID indicates.

Disable the link between PE-1 and PE-2:

```
[/]
A:admin@PE-1# configure {
  router "Base" {
    interface "int-PE-1-PE-2" {
      admin-state disable
    }
  }
  exit all
```

Because PE-2 disappears as a traffic-engineered (TE) IS-IS neighbor of PE-1, the End.X function that corresponds with the interface "int-PE-1-PE-2" is no longer instantiated. The IPv6 route table and IPv6 FIB indicate that data transport from PE-1 to PE-2 and P-5 now follows a path with a higher metric via P-4. The route to the End.X function that corresponds with the interface "int-PE-1-PE-2" is no longer present. There is no longer LFA protection for the route to P-5. Verify the IPv6 route table.

```
[/]
A:admin@PE-1# show router route-table ipv6

=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type  Proto  Age      Pref
  Next Hop[Interface Name]                       Metric
-----
---snip---
2001:db8::2:2/128                                Remote ISIS   00h00m46s 18
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"           51
---snip---
2001:db8::2:5/128                                Remote ISIS   00h16m33s 18
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"           30
2001:db8::168:12:0/126                           Remote ISIS   00h00m46s 18
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"           61
---snip---
2001:db8::168:25:0/126                            Remote ISIS   00h00m46s 18
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"           51
```

```

---snip---
-----
No. of Routes: 17
---snip---
=====

```

Verify the corresponding IPv6 FIB.

```

[/]
A:admin@PE-1# show router fib 1 ipv6

=====
FIB Display
=====
Prefix [Flags]                                Protocol
NextHop
-----
---snip---
2001:db8::2:2/128                             ISIS
  fe80::616:1ff:fe01:1 (int-PE-1-P-4)
---snip---
2001:db8::168:12:0/126                       ISIS
  fe80::616:1ff:fe01:1 (int-PE-1-P-4)
---snip---
2001:db8::168:25:0/126                       ISIS
  fe80::616:1ff:fe01:1 (int-PE-1-P-4)
---snip---
-----
Total Entries : 17
-----
=====

```

Verify the SRv6 local SIDs and SRv6 base routing instance on PE-1. The SID that corresponds with the interface “int-PE-1-PE-2” is no longer present and is no longer advertised to the other routers.

```

[/]
A:admin@PE-1# show router segment-routing-v6 local-sid

=====
Segment Routing v6 Local SIDs
=====
SID                                           Type      Function
Locator
Context
-----
2001:db8:aaaa:101:0:1000::                 End       1
  PE-1_loc
  Base
2001:db8:aaaa:101:78a6:b000::              End.DT6   494187
  PE-1_loc
  SvcId: 2 Name: VPRN_2
2001:db8:aaaa:101:78a6:c000::              End.DT4   494188
  PE-1_loc
  SvcId: 2 Name: VPRN_2
2001:db8:aaaa:101:78a6:d000::              End.X     494189
  PE-1_loc
  None
-----
SIDs : 4
-----
=====

```

The End.X function with SID 2001:db8:aaaa:101:78a6:e000:: that corresponds with the interface "int-PE-1-PE-2" is no longer instantiated.

```
[/]
A:admin@PE-1# show router segment-routing-v6 base-routing-instance

=====
Segment Routing v6 Base Routing Instance
=====
Locator
Type      Function      SID              Status/InstId
SRH-mode Protection Interface
-----
PE-1_loc
End              1 2001:db8:aaaa:101:0:1000::      ok
USP
-----
Auto-allocated End.X: USP Protected,
-----
End.X      *494189 2001:db8:aaaa:101:78a6:d000::      0
USP        Protected int-PE-1-P-4
ISIS Level: L2 Mac Address: 04:16:01:01:00:01 Nbr Sys Id: 0010.0100.1004
-----
Legend: * - System allocated
```

Verify the IPv6 tunnel table. There are no longer any backup tunnels and SRv6 data is transported to all destinations via the link between PE-1 and P-4.

```
[/]
A:admin@PE-1# show router tunnel-table ipv6

=====
IPv6 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref
NextHop          Color
-----
2001:db8:aaaa:101:78a6:d000::/128      srv6-isis SRV6  524293  0
fe80::616:1ff:fe01:1-"int-PE-1-P-4"    10
2001:db8:aaaa:102::/64                srv6-isis SRV6  524289  0
fe80::616:1ff:fe01:1-"int-PE-1-P-4" 51
2001:db8:aaaa:113::/64                  srv6-isis SRV6  524290  0
fe80::616:1ff:fe01:1-"int-PE-1-P-4"    20
2001:db8:aaaa:114::/64                  srv6-isis SRV6  524291  0
fe80::616:1ff:fe01:1-"int-PE-1-P-4"    10
2001:db8:aaaa:115::/64                  srv6-isis SRV6  524292  0
fe80::616:1ff:fe01:1-"int-PE-1-P-4"    30
-----
---snip---
```

Verify the interfaces that the tunnels are using. There is no longer any possibility for alternate routes.

```
[/]
A:admin@PE-1# show router fp-tunnel-table 1 ipv6

=====
IPv6 Tunnel Table Display
Legend:
label stack is ordered from bottom-most to top-most
```

```

B - FRR Backup
=====
Destination                               Protocol   Tunnel-ID
  Lbl/SID                                   Intf/Tunnel
  NextHop                                   Intf/Tunnel
  Lbl/SID (backup)
  NextHop (backup)
-----
2001:db8:aaaa:102::/64                     SRV6      524289
-
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"      1/1/c2/1:1000
2001:db8:aaaa:113::/64                     SRV6      524290
-
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"      1/1/c2/1:1000
2001:db8:aaaa:114::/64                     SRV6      524291
-
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"      1/1/c2/1:1000
2001:db8:aaaa:115::/64                     SRV6      524292
-
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"      1/1/c2/1:1000
2001:db8:aaaa:101:78a6:d000::/128          SRV6      524293
-
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"      1/1/c2/1:1000
-----
Total Entries : 5
=====

```

Enable the link between PE-1 and PE-2 to restore the initial topology:

```

[/]
A:admin@PE-1# configure {
  router "Base" {
    interface "int-PE-1-PE-2" {
      admin-state enable
    }
  }
  exit all
}

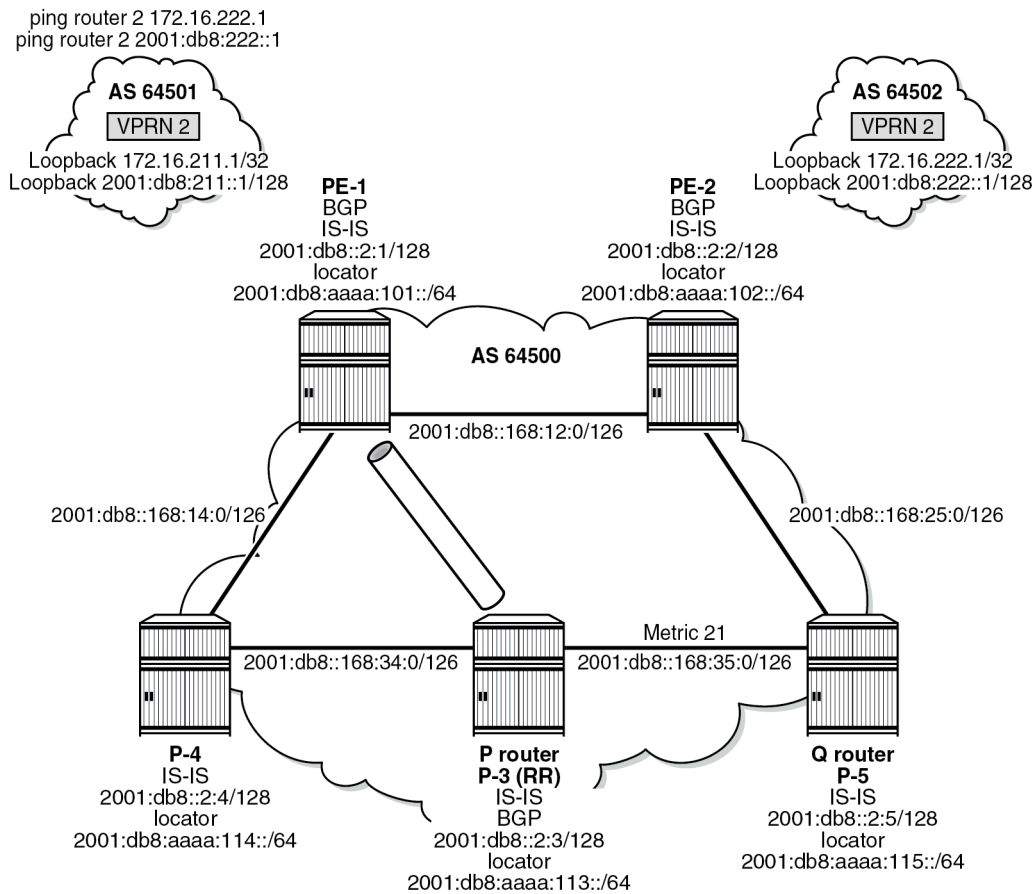
```

The End.X function that corresponds with the interface "int-PE-1-PE-2" is re-instantiated, but with SID 2001:db8:aaaa:101:78a6:f000:: and SRv6 Tunnel-ID 524295.

Modify metrics so that the P-router and the Q-router no longer coincide

Figure 3 shows the example topology with modified metrics that is used to verify the behavior when a disjointed P-router and Q-router provide TI-LFA protection.

Figure 375: Example topology with metric 21 between P-3 and P-5



37607

Metrics can be modified for the interface "int-PE-2-P-5" on PE-2 with the command **configure router "Base" isis 0 interface "int-PE-2-P-5" level 2 metric <value>**. Similar commands apply for the interface "int-P-3-P-5" on P-3, and for the interfaces "int-P-5-PE-2" and "int-P-5-P-3" on P-5.

Verify the IPv6 route table. P-5 is now reached via interface "int-PE-1-PE-2".

```
[/]
A:admin@PE-1# show router route-table ipv6

=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]                               Type  Proto  Age           Pref
Next Hop[Interface Name]                       Metric
-----
---snip---
2001:db8::2:5/128                               Remote ISIS  00h01m07s  18
    fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"        20
---snip---
2001:db8::168:35:0/126                         Remote ISIS  00h00m31s  18
    fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"        41
---snip---
```

```
No. of Routes: 18
---snip---
=====
```

Verify the corresponding IPv6 FIB.

```
[/]
A:admin@PE-1# show router fib 1 ipv6

=====
FIB Display
=====
Prefix [Flags]                                Protocol
NextHop
-----
---snip---
2001:db8::2:5/128                              ISIS
  fe80::60e:1ff:fe01:1 (int-PE-1-PE-2)
---snip---
2001:db8::168:35:0/126                         ISIS
  fe80::60e:1ff:fe01:1 (int-PE-1-PE-2)
---snip---
-----
Total Entries : 18
-----
=====
```

On PE-1, apart from the metrics, the IS-IS data base, the SRv6 local SIDs, and the SRv6 base routing instance do not change.

```
[/]
A:admin@PE-1# show router tunnel-table ipv6

=====
IPv6 Tunnel Table (Router: Base)
=====
Destination                                Owner      Encap TunnelId  Pref
NextHop                                    Color      Encap  TunnelId  Pref
-----
---snip---
2001:db8:aaaa:115::/64                   srv6-isis SRV6 524292  0
  fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"    20
-----
---snip---
=====
```

Verify the interfaces that the tunnels are using.

```
[/]
A:admin@PE-1# show router fp-tunnel-table 1 ipv6

=====
IPv6 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol      Tunnel-ID
Lbl/SID
NextHop                                    Intf/Tunnel
Lbl/SID (backup)
```

```

NextHop (backup)
-----
---snip---
2001:db8:aaaa:115::/64          SRV6          524292
-
  fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"  1/1/c1/1:1000
---snip---
-----
Total Entries : 6
-----
=====

```

Without support for LFA on P-3, the TI-LFA computation on PE-1 does not lead to a PQ-router, because the End.X functions on P-3 are neither instantiated nor advertised to the other routers.

On P-3, verify the SRv6 local SIDs and SRv6 base routing instance. The End.X functions are not yet instantiated.

```

[/]
A:admin@P-3# show router segment-routing-v6 local-sid

=====
Segment Routing v6 Local SIDs
=====
SID                               Type          Function
Locator
Context
-----
2001:db8:aaaa:113:0:1000::        End           1
P-3_loc
Base
-----
SIDs : 1
-----
=====

```

Only the End function is already instantiated. The End.X functions are not yet instantiated.

```

[/]
A:admin@P-3# show router segment-routing-v6 base-routing-instance

=====
Segment Routing v6 Base Routing Instance
=====
Locator
Type          Function      SID              Status/InstId
SRH-mode Protection Interface
-----
P-3_loc
End           1 2001:db8:aaaa:113:0:1000::  ok
USP
-----
Auto-allocated End.X: USP Protected,
-----
-----
Legend: * - System allocated

```


Configure LFA on P-3

```
[/]
A:admin@P-3# configure {
  router "Base" {
    isis 0 {
      loopfree-alternate {
      }
    }
  }
  exit all
}
```

The IS-IS database contains additional information about the End.X functions that are instantiated on P-3.

```
[/]
A:admin@PE-1# show router isis database detail

=====
Rtr Base ISIS Instance 0 Database (detail)
=====
---snip---
Displaying Level 2 database
-----
---snip---
-----
LSP ID   : P-3.00-00                               Level   : L2
---snip---
TLVs :
---snip---
TE IS Nbrs :
  Nbr   : P-4.00
  Default Metric : 10
---snip---
End.X-SID: 2001:db8:aaaa:113:0:4000:: flags:B algo:0 weight:0 endpoint:End.X-USP
TE IS Nbrs :
  Nbr   : P-5.00
  Default Metric : 21
---snip---
End.X-SID: 2001:db8:aaaa:113:0:5000:: flags:B algo:0 weight:0 endpoint:End.X-USP
---snip---
Level (2) LSP Count : 5
-----
---snip--
=====
```

On PE-1, the IS-IS data base, the IPv6 route table, the IPv6 FIB, the SRv6 local SIDs, and the SRv6 base routing instance do not change.

On PE-3, verify the SRv6 local SIDs and SRv6 base routing instance. The End.X functions are also instantiated.

```
[/]
A:admin@P-3# show router segment-routing-v6 local-sid

=====
Segment Routing v6 Local SIDs
=====
SID                               Type           Function
Locator
Context
-----
2001:db8:aaaa:113:0:1000::       End            1
P-3_loc
Base
```

```

2001:db8:aaaa:113:0:4000::          End.X          4
P-3_loc
None
2001:db8:aaaa:113:0:5000::          End.X          5
P-3_loc
None
-----
SIDs : 3
-----
=====

```

```

[/]
A:admin@P-3# show router segment-routing-v6 base-routing-instance

=====
Segment Routing v6 Base Routing Instance
=====
Locator
Type      Function      SID                      Status/InstId
SRH-mode Protection Interface
-----
P-3_loc
End              1 2001:db8:aaaa:113:0:1000::      ok
USP
-----
Auto-allocated End.X: USP Protected,
-----
End.X          *4 2001:db8:aaaa:113:0:4000::      0
USP            Protected int-P-3-P-4
ISIS Level: L2 Mac Address: 04:16:01:01:00:0b Nbr Sys Id: 0010.0100.1004
End.X          *5 2001:db8:aaaa:113:0:5000::      0
USP            Protected int-P-3-P-5
ISIS Level: L2 Mac Address: 04:1a:01:01:00:0b Nbr Sys Id: 0010.0100.1005
-----
Legend: * - System allocated

```

Verify the IPv6 tunnel table. The existing routes to PE-2 and P-5 are loop-protected.

```

[/]
A:admin@PE-1# show router tunnel-table ipv6

=====
IPv6 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId Pref
NextHop          Color
-----
---snip---
2001:db8:aaaa:102::/64 [L]          srv6-isis SRV6 524289 0
fe80::60e:1ff:fe01:1-"int-PE-1-PE-2" 10
---snip---
2001:db8:aaaa:115::/64 [L]          srv6-isis SRV6 524292 0
fe80::60e:1ff:fe01:1-"int-PE-1-PE-2" 20
-----
Flags: B = BGP or MPLS backup hop available
      L = Loop-Free Alternate (LFA) hop available
      ---snip---
=====

```

Verify the interfaces that the tunnels are using. Interface “int-PE-1-PE-2” is configured on port 1/1/c1/1:1000. Interface “int-PE-1-P-4” is configured on port 1/1/c2/1:1000.

When the link between PE-1 and PE-2 is operational, SRv6 data is transported to PE-2 over this link, via next hop fe80::60e:1ff:fe01:1-"int-PE-1-PE-2". When the link between PE-1 and PE-2 fails, SRv6 data is transported to PE-2 using a FRR backup link between PE-1 and P-4, via backup next hop fe80::616:1ff:fe01:1-"int-PE-1-P-4". The SRv6 data is transported to PE-2 then via an SRv6 tunnel to the End.X function on P-3, as the backup SID 2001:db8:aaaa:113:0:5000:: indicates, followed by source routing to P-5.

```
[/]
A:admin@PE-1# show router fp-tunnel-table 1 ipv6

=====
IPv6 Tunnel Table Display

Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol      Tunnel-ID
Lbl/SID
NextHop                                     Intf/Tunnel
Lbl/SID (backup)
NextHop (backup)
-----
2001:db8:aaaa:102::/64                      SRV6          524289
-
  fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"      1/1/c1/1:1000
  2001:db8:aaaa:113:0:5000::
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"(B)    1/1/c2/1:1000
---snip---
2001:db8:aaaa:115::/64                      SRV6          524292
-
  fe80::60e:1ff:fe01:1-"int-PE-1-PE-2"      1/1/c1/1:1000
  2001:db8:aaaa:113:0:5000::
  fe80::616:1ff:fe01:1-"int-PE-1-P-4"(B)    1/1/c2/1:1000
---snip---
-----
Total Entries : 6
-----
=====
```

With the topology as shown in [Figure 375: Example topology with metric 21 between P-3 and P-5](#), this behavior is described as follows:

There is no regular LFA protection for the destination prefix to PE-2 using the protected PE-1-PE-2 link, which can be understood when the regular LFA inequality is determined using a shortest-path distance (Spd) calculation:

$$Spd(N, D) < Spd(N, S) + Spd(S, D)$$

where

Spd is the shortest path distance (according to level 2 metrics)

S is the source router (PE-1)

D is the destination router (PE-2)

N is the alternate next hop router or neighboring node (P-4)

If the outcome of the calculation is true, then regular LFA protection is valid; if the outcome is false, then there is no LFA protection.

In this case the outcome is false.

$$\text{Spd}(P-4, PE-2) < \text{Spd}(P-4, PE-1) + \text{Spd}(PE-1, PE-2)$$

$$(10 + 21 + 10) < 10 + 10$$

There is TI-LFA protection for all destination prefixes using the protected PE-1-PE-2 link:

The TI-LFA inequality for the extended P-space P' is:

$$\text{Spd}(N, Y_i) < \text{Spd}(N, S) + \text{Spd}(S, Y_i)$$

$$\text{Spd}(P-4, Y_i) < \text{Spd}(P-4, PE-1) + \text{Spd}(PE-1, Y_i)$$

where Y_i is the set of routers {P-3, P-5} that are reachable from PE-1 and its neighbor P-4 on the post-convergence path to PE-2, without traversing the link between PE-1 and PE-2.

Apply this inequality to the set of routers Y_i :

For $Y_i=P-3$, the outcome is true. So P-3 is in P':

$$\text{Spd}(P-4, P-3) < \text{Spd}(P-4, PE-1) + \text{Spd}(PE-1, P-3)$$

$$10 < 10 + (10 + 10)$$

For $Y_i=P-5$, the outcome is false. So P-5 is **not** in P':

$$\text{Spd}(P-4, P-5) < \text{Spd}(P-4, PE-1) + \text{Spd}(PE-1, P-5)$$

$$(10 + 21) < 10 + (10 + 10)$$

So, the extended P-space $P' = \{P-3\}$

The TI-LFA inequality for the Q-space Q is:

$$\text{Spd}(Z_i, D) < \text{Spd}(Z_i, S) + \text{Spd}(S, D)$$

$$\text{Spd}(Z_i, PE-2) < \text{Spd}(Z_i, PE-1) + \text{Spd}(PE-1, PE-2)$$

where Z_i is the set of routers {P-3, P-5} that are reachable from PE-2 using reverse SPF on the post-convergence path to PE-1 without traversing the link between PE-1 and PE-2.

Apply this inequality to the set of routers Z_i :

For $Z_i=P-3$, the outcome is false. So P-3 is **not** in Q:

$$\text{Spd}(P-3, PE-2) < \text{Spd}(P-3, PE-1) + \text{Spd}(PE-1, PE-2)$$

$$(21 + 10) < (10 + 10) + 10$$

For $Z_i=P-5$, the outcome is true. So P-5 is in Q:

$$\text{Spd}(P-5, PE-2) < \text{Spd}(P-5, PE-1) + \text{Spd}(PE-1, PE-2)$$

$$10 < (21 + 10 + 10) + 10$$

So, the Q-space $Q = \{P-5\}$

So, the link between PE-1 and PE-2 is TI-LFA protected with the P-router P-3 and Q-router P-5.

Disable the link between PE-1 and PE-2:

```
[/]
A:admin@PE-1# configure {
  router "Base" {
    interface "int-PE-1-PE-2" {
      admin-state disable
    }
  }
  exit all
```

Because PE-2 disappears as a TE IS-IS neighbor of PE-1, the End.X function that corresponds with the interface "int-PE-1-PE-2" is no longer instantiated. The IPv6 route table and IPv6 FIB indicate that data transport from PE-1 to PE-2 and P-5 now follows a path with a higher metric via P-4. Verify the IPv6 route table.

```
[/]
A:admin@PE-1# show router route-table ipv6

=====
IPv6 Route Table (Router: Base)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]                Type   Proto   Age           Pref
Metric
-----
---snip---
2001:db8::2:2/128                        Remote ISIS   00h00m23s  18
fe80::616:1ff:fe01:1-"int-PE-1-P-4"      51
---snip---
2001:db8::2:5/128                        Remote ISIS   00h00m23s  18
fe80::616:1ff:fe01:1-"int-PE-1-P-4"      41
2001:db8::168:12:0/126                   Remote ISIS   00h00m23s  18
fe80::616:1ff:fe01:1-"int-PE-1-P-4"      61
---snip---
2001:db8::168:25:0/126                   Remote ISIS   00h00m23s  18
fe80::616:1ff:fe01:1-"int-PE-1-P-4"      51
---snip---
2001:db8::168:35:0/126                   Remote ISIS   00h00m23s  18
fe80::616:1ff:fe01:1-"int-PE-1-P-4"      41
---snip---
-----
No. of Routes: 17
---snip---
=====
```

Verify the corresponding IPv6 FIB.

```
[/]
A:admin@PE-1# show router fib 1 ipv6

=====
FIB Display
=====
Prefix [Flags]
NextHop                                     Protocol
-----
```

```

---snip---
2001:db8::2:2/128                                ISIS
  fe80::616:1ff:fe01:1 (int-PE-1-P-4)
---snip---
2001:db8::2:5/128                                ISIS
  fe80::616:1ff:fe01:1 (int-PE-1-P-4)
2001:db8::168:12:0/126                           ISIS
  fe80::616:1ff:fe01:1 (int-PE-1-P-4)
---snip---
2001:db8::168:25:0/126                           ISIS
  fe80::616:1ff:fe01:1 (int-PE-1-P-4)
---snip---
2001:db8::168:35:0/126                           ISIS
  fe80::616:1ff:fe01:1 (int-PE-1-P-4)
---snip---
-----
Total Entries : 17
=====

```

Verify the SRv6 local SIDs and SRv6 base routing instance on PE-1. The SID that corresponds with the interface “int-PE-1-PE-2” is no longer present and is no longer advertised to the other routers.

```

[/]
A:admin@PE-1# show router segment-routing-v6 local-sid

=====
Segment Routing v6 Local SIDs
=====

```

SID	Type	Function
2001:db8:aaaa:101:0:1000:: PE-1_loc Base	End	1
2001:db8:aaaa:101:78a6:b000:: PE-1_loc SvcId: 2 Name: VPRN_2	End.DT6	494187
2001:db8:aaaa:101:78a6:c000:: PE-1_loc SvcId: 2 Name: VPRN_2	End.DT4	494188
2001:db8:aaaa:101:78a6:d000:: PE-1_loc None	End.X	494189

```

-----
SIDs : 4
=====

```

The End.X function that corresponds with the interface “int-PE-1-PE-2” is no longer instantiated.

```

[/]
A:admin@PE-1# show router segment-routing-v6 base-routing-instance

=====
Segment Routing v6 Base Routing Instance
=====

```

Locator	Type	Function	SID	Status/InstId
	SRH-mode	Protection	Interface	

```

PE-1_loc
End          1 2001:db8:aaaa:101:0:1000::      ok
USP
-----
Auto-allocated End.X: USP Protected,
-----
End.X        *494189 2001:db8:aaaa:101:78a6:d000::  0
USP          Protected int-PE-1-P-4
ISIS Level: L2 Mac Address: 04:16:01:01:00:01 Nbr Sys Id: 0010.0100.1004
-----
Legend: * - System allocated
    
```

Verify the IPv6 tunnel table. There are no longer any backup tunnels and SRV6 data is transported to all destinations via the link between PE-1 and P-4.

```

[/]
A:admin@PE-1# show router tunnel-table ipv6

=====
IPv6 Tunnel Table (Router: Base)
=====
Destination      Owner      Encap TunnelId  Pref
NextHop          Color
-----
2001:db8:aaaa:101:78a6:d000::/128  srv6-isis SRV6  524293  0
   fe80::616:1ff:fe01:1-"int-PE-1-P-4"  10
2001:db8:aaaa:102::/64             srv6-isis SRV6  524289  0
   fe80::616:1ff:fe01:1-"int-PE-1-P-4"  51
2001:db8:aaaa:113::/64             srv6-isis SRV6  524290  0
   fe80::616:1ff:fe01:1-"int-PE-1-P-4"  20
2001:db8:aaaa:114::/64             srv6-isis SRV6  524291  0
   fe80::616:1ff:fe01:1-"int-PE-1-P-4"  10
2001:db8:aaaa:115::/64             srv6-isis SRV6  524292  0
   fe80::616:1ff:fe01:1-"int-PE-1-P-4"  41
-----
---snip---
=====
    
```

Verify the interfaces that the tunnels are using. There is no longer any possibility for alternate routes.

```

[/]
A:admin@PE-1# show router fp-tunnel-table 1 ipv6

=====
IPv6 Tunnel Table Display
=====
Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination      Protocol  Tunnel-ID
Lbl/SID
NextHop          Intf/Tunnel
Lbl/SID (backup)
NextHop (backup)
-----
2001:db8:aaaa:102::/64             SRV6      524289
-
   fe80::616:1ff:fe01:1-"int-PE-1-P-4"  1/1/c2/1:1000
2001:db8:aaaa:113::/64             SRV6      524290
-
    
```

```

    fe80::616:1ff:fe01:1-"int-PE-1-P-4"      1/1/c2/1:1000
2001:db8:aaaa:114::/64                      SRV6    524291
-
    fe80::616:1ff:fe01:1-"int-PE-1-P-4"      1/1/c2/1:1000
2001:db8:aaaa:115::/64                      SRV6    524292
-
    fe80::616:1ff:fe01:1-"int-PE-1-P-4"      1/1/c2/1:1000
2001:db8:aaaa:101:78a6:d000::/128          SRV6    524293
-
    fe80::616:1ff:fe01:1-"int-PE-1-P-4"      1/1/c2/1:1000
-----
Total Entries : 5
-----
=====

```

Enable the link between PE-1 and PE-2 to restore the initial topology:

```

[/]
A:admin@PE-1# configure {
  router "Base" {
    interface "int-PE-1-PE-2" {
      admin-state enable
    }
  }
  exit all

```

The End.X function that corresponds with the interface "int-PE-1-PE-2" is re-instantiated, but with SID 2001:db8:aaaa:101:78a6:e000:: and Tunnel-ID 524296.

Conclusion

To guard against the failure of the initial data path, LFA protection via an LFA backup path is possible for SRv6 data transport.

VSR Installation and Setup

This section provides VSR installation and setup information for the following topics:

- [VSR Hypervisor Configuration](#)

VSR Hypervisor Configuration

This chapter provides information about VSR hypervisor configuration.

Topics in this chapter include:

- [Applicability](#)
- [Overview](#)
- [Configuration](#)
- [Conclusion](#)

Applicability

The information and configuration in this chapter applies to all VSR releases.

Overview

Deployment of the Nokia Virtualized Service Router (VSR) virtual machine (VM) requires a highly tuned hypervisor. Configuration of a KVM hypervisor server requires changing the BIOS settings, the kernel parameters, and, finally, the XML file that defines the VSR VM. Running non-VSR VMs on the same hypervisor is only allowed if there is no resource overlap or oversubscription with the VSR VM. This includes dedicated memory and CPU pinning so that no other VM can use the VSR's CPUs.

Configuration

The detailed configuration requirements are described in the *VSR VNF Installation and Setup Guide*. The following examples show VSR VM deployments on commonly used hypervisor hardware, such as RHEL 7 or Centos 7, and must be adapted as needed. Some commands shown may not be installed by default.

The configuration consists of the following topics which apply across all server types:

- BIOS requirements
- CPU settings, such as NUMA and hyperthreading
- Host OS and Network Interface Card (NIC) requirements

Specific configuration examples are provided for the following servers:

- Nokia VSR Appliance (VSR-a) AirFrame server
- Dell server (with and without hyperthreading)
- HPE server (with and without hyperthreading)

A complete XML file example is also provided.

Required BIOS settings

The required BIOS settings are:

- SR-IOV enabled (if used)
- Intel VT-x enabled
- Intel VT-d enabled
- x2APIC enabled
- NUMA enabled (if used)
- hardware prefetcher disabled
- I/O non posted prefetching disabled
- adjacent cache line prefetching disabled
- PCIe Active State Power Management (ASPM) support disabled
- Advanced Configuration and Power Interface (ACPI)
 - P-state disabled
 - C-state disabled
- NUMA node interleaving disabled (if NUMA enabled)
- turbo boost enabled
- power management set to maximum or high performance
- CPU frequency set to maximum supported without overclocking
- memory speed set to maximum supported without overclocking

CPU terminology

This chapter uses the following terminology to describe CPUs:

- socket: the physical socket on the motherboard that the CPU is inserted into, typically one or two per motherboard
- CPU: depending on the context, this is either the physical CPU package that is inserted into the core or a logical processor that can be assigned to the VM in the XML file. (CPU can mean physical package, die, core, or logical processor.)
- core: the physical CPU die contains multiple CPU cores or processors that can be further logically divided into threads (two per core)
- processor: logical processor, CPU, or thread; either one per core if hyperthreading is disabled or two per core if hyperthreading is enabled
- pCPU: either one of the CPU cores if hyperthreading is disabled or one of the threads if hyperthreading is enabled
- vCPU: a virtual CPU assigned to the VM, running on one of the pCPUs.

Multiple-socket servers

To use multiple-socket servers with multiple NUMA nodes, NUMA must be enabled in the BIOS. When using multiple NUMA nodes on a server, ensure that all resources belonging to a single VSR do not span across more than one NUMA node. Consequently, to fully utilize a dual socket server, it is necessary to provision multiple VSR VMs or use non-VSR VMs on one of the NUMA nodes.

Hyperthreading

VSR CPU pinning configuration is more complex if hyperthreading is enabled. As hyperthreading does not always provide a performance benefit, hyperthreading can be disabled if it is not needed. Hyperthreading is recommended for CPU-intensive packet applications like Application Assurance (AA) and IPsec.

Host OS

Ensure that the host OS and kernel are supported by consulting the *SR OS Software Release Notes* for the specific VSR release.

To verify the host OS:

```
[admin@hyp56 ~]$ cat /etc/*release*
CentOS Linux release 7.9.2009 (Core)
Derived from Red Hat Enterprise Linux 7.9 (Source)
...
```

To verify the host kernel:

```
[admin@hyp56 ~]$ uname -a
Linux hyp56 3.10.0-1160.66.1.el7.x86_64 #1 SMP Wed May 18 16:02:34 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
```

Host NICs

Ensure that the host NICs are supported by consulting the *SR OS Software Release Notes* for the specific VSR release. The firmware version must also be supported, as well as the drivers, if applicable (not applicable to PCI-PT for example). The **lshw** command output shows all host NICs and their PCI addresses:

```
[root@hyp56 ~]# lshw -c network -businfo
Bus info          Device          Class           Description
=====
pci@0000:04:00.0  p6p1           network        MT27700 Family [ConnectX-4]
pci@0000:04:00.1  p6p2           network        MT27700 Family [ConnectX-4]
pci@0000:01:00.0  em1            network        Ethernet Controller 10-Gigabit X540-AT2
pci@0000:01:00.1  em2            network        Ethernet Controller 10-Gigabit X540-AT2
pci@0000:06:00.0  p5p1           network        Ethernet Controller X710 for 10GbE SFP+
pci@0000:06:00.1  p5p2           network        Ethernet Controller X710 for 10GbE SFP+
pci@0000:06:00.2  p5p3           network        Ethernet Controller X710 for 10GbE SFP+
pci@0000:06:00.3  p5p4           network        Ethernet Controller X710 for 10GbE SFP+
pci@0000:09:00.0  em3            network        I350 Gigabit Network Connection
pci@0000:09:00.1  em4            network        I350 Gigabit Network Connection
pci@0000:81:00.0  network       network        Ethernet 10G 2P X520 Adapter
pci@0000:81:00.1  network       network        Ethernet 10G 2P X520 Adapter
pci@0000:83:00.0  p4p1           network        MT27700 Family [ConnectX-4]
pci@0000:83:00.1  p4p2           network        MT27700 Family [ConnectX-4]
pci@0000:83:00.2  p4p1_0        network        MT27700 Family [ConnectX-4 Virtual Function]
pci@0000:83:00.4  p4p2_0        network        MT27700 Family [ConnectX-4 Virtual Function]
pci@0000:85:00.0  p3p1           network        Ethernet 10G 2P X520 Adapter
pci@0000:85:00.1  p3p2           network        Ethernet 10G 2P X520 Adapter
pci@0000:87:00.0  p2p1           network        Ethernet 10G 2P X520 Adapter
pci@0000:87:00.1  p2p2           network        Ethernet 10G 2P X520 Adapter
br-mgmt          network        Ethernet interface
```

virbr0-nic	network	Ethernet interface
virbr0	network	Ethernet interface
vnet0	network	Ethernet interface

This server has several types of NICs, including Mellanox ConnectX-4 and Intel X710. The PCI address format is *domain:bus:device.function*. Different ports on the same physical NIC have addresses that differ only by the function number; for example, in the preceding **lshw** output, the first two ports are on the same physical NIC. Virtual functions (VFs) may have a different device or function number than their physical NIC, and VF numbering schemes may vary. Another way to check the installed NICs is using the **lspci** command:

```
[admin@hyp62 ~]$ lspci -v | grep "Ethernet controller"
02:00.0 Ethernet controller: Broadcom Inc. and subsidiaries NetXtreme BCM5719 Gigabit Ethernet PCIe (rev 01)
02:00.1 Ethernet controller: Broadcom Inc. and subsidiaries NetXtreme BCM5719 Gigabit Ethernet PCIe (rev 01)
02:00.2 Ethernet controller: Broadcom Inc. and subsidiaries NetXtreme BCM5719 Gigabit Ethernet PCIe (rev 01)
02:00.3 Ethernet controller: Broadcom Inc. and subsidiaries NetXtreme BCM5719 Gigabit Ethernet PCIe (rev 01)
04:00.0 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
04:00.1 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
05:00.0 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 01)
05:00.1 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 01)
05:00.2 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 01)
05:00.3 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 01)
05:02.0 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 01)
81:00.0 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
81:00.1 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
```

Nokia AirFrame server (VSR-a)

The VSR-a hypervisor is an AirFrame server that has been customized by Nokia R&D.

Server model and motherboard

The server model can be identified with the **dmidecode** command:

```
[root@hyp70 admin]# dmidecode -t 2
# dmidecode 3.0
Getting SMBIOS data from sysfs.
SMBIOS 3.2 present.
# SMBIOS implementations newer than version 3.0 are not
# fully supported by this version of dmidecode.

Handle 0x0002, DMI type 2, 15 bytes
Base Board Information
    Manufacturer: Nokia Solutions and Networks
    Product Name: AR-D52BT-A/AF0310.01
    ...
```

The Product Name field describes the server type and motherboard; this example corresponds to a VSR-a SN8.

CPU

The **lscpu** command shows the number of sockets, cores, threads, and their numbering scheme:

```
[root@hyp70 admin]# lscpu
Architecture:          x86_64
CPU op-mode(s):       32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                48
On-line CPU(s) list:  0-47
Thread(s) per core:   2
Core(s) per socket:   24
Socket(s):             1
NUMA node(s):         1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 85
Model name:            Intel(R) Xeon(R) Platinum 8160 CPU @ 2.10GHz
Stepping:              4
CPU MHz:               2100.000
BogoMIPS:              4200.00
Virtualization:        VT-x
L1d cache:             32K
L1i cache:            32K
L2 cache:              1024K
L3 cache:              33792K
NUMA node0 CPU(s):    0-47
```

In this example, there is a single socket, the CPU has 24 cores, and hyperthreading is enabled, resulting in 48 vCPUs, numbered consecutively from 0 to 47.

When vCPUs are assigned to a VSR VM and hyperthreading is enabled in the BIOS, it is important to ensure that the first two vCPUs are siblings of the same pCPU core, the next two vCPUs are siblings of some other pCPU core, and so on. Verify the hyperthreading numbering scheme using the **cat /proc/cpuinfo** command:

```
[root@hyp70 admin]# cat /proc/cpuinfo | egrep "processor|physical id|core id"
processor      : 0
physical id   : 0
core id       : 0
processor      : 1
physical id   : 0
core id       : 1
processor      : 2
physical id   : 0
core id       : 2
processor      : 3
physical id   : 0
core id       : 3
...
processor      : 24
physical id   : 0
core id       : 0
processor      : 25
physical id   : 0
core id       : 1
processor      : 26
```

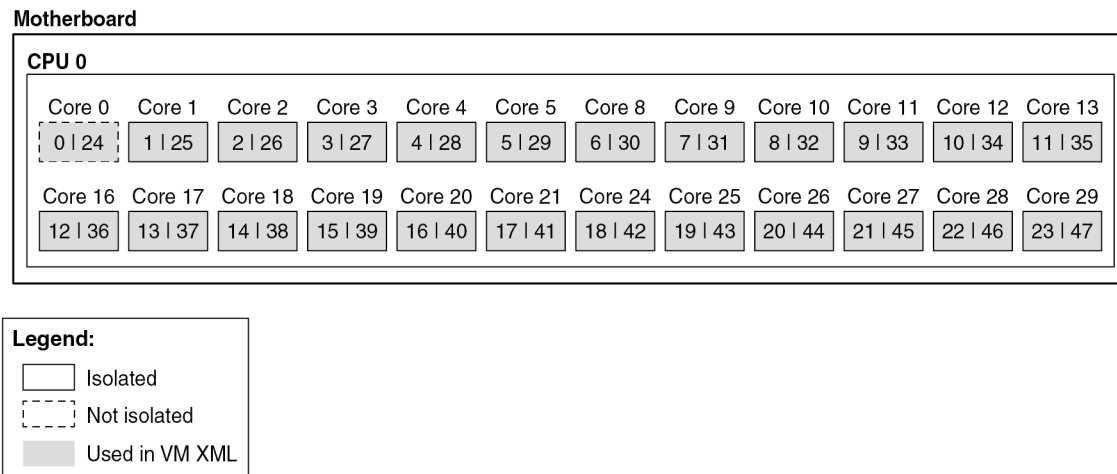
```
physical id : 0
core id    : 2
processor  : 27
physical id : 0
core id    : 3
...
```

This output is in order of processor: in order of CPU if hyperthreading is disabled, and in order of thread if hyperthreading is enabled. This output is abbreviated here to only show the first four vCPUs on each CPU core. The physical id (CPU socket) is always 0 because there is only one socket, and the core id (pCPU number) increases with the processor (vCPU) number until vCPU 23, then wraps around and the core id starts from zero again.

This numbering scheme makes all vCPU numbers consecutive whether hyperthreading is tuned on or off in the BIOS, and therefore if the CPU pinning in the XML file assumed hyperthreading off, the XML file will remain valid even if hyperthreading is turned on.

[Figure 376: Numbering scheme Airframe with hyperthreading enabled](#) shows the numbering scheme. All available CPUs are assigned to VM and emulator functions and core id numbers are not all consecutive, while processor numbers are.

Figure 376: Numbering scheme Airframe with hyperthreading enabled



38114

Network ports

This server has the following network ports:

```
[root@hyp70 xml]# lspci -v | grep "Ethernet controller"
17:00.0 Ethernet controller: Mellanox Technologies MT27800 Family [ConnectX-5]
17:00.1 Ethernet controller: Mellanox Technologies MT27800 Family [ConnectX-5]
6a:00.0 Ethernet controller: Intel Corporation Ethernet Connection X722 for 10GbE SFP+ (rev 04)
6a:00.1 Ethernet controller: Intel Corporation Ethernet Connection X722 for 10GbE SFP+ (rev 04)
b3:00.0 Ethernet controller: Mellanox Technologies MT27800 Family [ConnectX-5]
b3:00.1 Ethernet controller: Mellanox Technologies MT27800 Family [ConnectX-5]
```

Kernel parameters

The following kernel parameters include all required and recommended parameters for CPU isolation based on the installed CPU's numbering scheme and the number of hugepages corresponding to the maximum amount of memory usable by a VSR VM.

```
root@hyp70 admin]# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.10.0-862.51.1.el7.x86_64 root=UUID=dae36453-3a6b-4d93-b6b8-80b74de3b823
ro hugepagesz=1G default_hugepagesz=1G hugepages=64 isolcpus=1-23,25-47 selinux=0 audit=
0 kvm_intel.ple_gap=0 pci=realloc pci_aspm=off intel_iommu=on iommu=pt nopat ixgbe.allow_
unsupported_sfp=1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 intremap=no_x2apic_optout console=tty0 console=
ttyS0,115200 crashkernel=auto rd.md.uuid=c3dae94c:76b2416c:39008bea:d16a2541 rd.md.uuid=
6d84171b:6588e600:01020790:6c9744e2 net.ifnames=0 rhgb quiet
```

On this server, all vCPUs are isolated except for the threads on the first pCPU, threads 0 and 24. These two threads will be used for the emulatorpin in the XML file.

XML file

The following section configures the memory with 64GB of hugepages and ACPI:

```
<domain type='kvm'>
  <name>VSR-A</name>
  <uuid>afa0cce8-8ee0-496f-a6e0-d8c49764cba5</uuid>
  <description>VSR-a SN8</description>
  <memory unit='G'>64</memory>
  <memoryBacking>
    <hugepages>
      <page size='1' unit='G' nodeset='0' />
    </hugepages>
    <nosharepages />
  </memoryBacking>
  <features>
    <acpi />
  </features>
```

The next section configures CPU settings for a single thread per pCPU. This configuration still takes full advantage of all pCPUs and is valid even if hyperthreading is enabled in the BIOS:

```
<vcpu placement='static'>23</vcpu>
<cputune>
  <vcpupin vcpu='0' cpuset='1' />
  <vcpupin vcpu='1' cpuset='2' />
  <vcpupin vcpu='2' cpuset='3' />
  <vcpupin vcpu='3' cpuset='4' />
  <vcpupin vcpu='4' cpuset='5' />
  <vcpupin vcpu='5' cpuset='6' />
  <vcpupin vcpu='6' cpuset='7' />
  <vcpupin vcpu='7' cpuset='8' />
  <vcpupin vcpu='8' cpuset='9' />
  <vcpupin vcpu='9' cpuset='10' />
  <vcpupin vcpu='10' cpuset='11' />
  <vcpupin vcpu='11' cpuset='12' />
  <vcpupin vcpu='12' cpuset='13' />
  <vcpupin vcpu='13' cpuset='14' />
  <vcpupin vcpu='14' cpuset='15' />
  <vcpupin vcpu='15' cpuset='16' />
  <vcpupin vcpu='16' cpuset='17' />
  <vcpupin vcpu='17' cpuset='18' />
```



```

<vcupin vcpu='18' cpuset='19' />
<vcupin vcpu='19' cpuset='20' />
<vcupin vcpu='20' cpuset='21' />
<vcupin vcpu='21' cpuset='22' />
<vcupin vcpu='22' cpuset='23' />
<emulatorpin cpuset='0' />
</cputune>
<cpu mode='host-model'>
  <model fallback='allow' />
</cpu>

```

Note the emulatorpin directive, which pins emulator threads onto a dedicated CPU that was not isolated in the kernel parameters.

The following is an alternate CPU configuration that is only valid with hyperthreading enabled in the BIOS and all threads assigned to the proper pCPUs for this system, based on the **cat /proc/cpuinfo** command:

```

<vcpu placement='static'>46</vcpu>
<cputune>
  <vcupin vcpu='0' cpuset='1' />
  <vcupin vcpu='1' cpuset='25' />
  <vcupin vcpu='2' cpuset='2' />
  <vcupin vcpu='3' cpuset='26' />
  <vcupin vcpu='4' cpuset='3' />
  <vcupin vcpu='5' cpuset='27' />
  <vcupin vcpu='6' cpuset='4' />
  <vcupin vcpu='7' cpuset='28' />
  <vcupin vcpu='8' cpuset='5' />
  <vcupin vcpu='9' cpuset='29' />
  <vcupin vcpu='10' cpuset='6' />
  <vcupin vcpu='11' cpuset='30' />
  <vcupin vcpu='12' cpuset='7' />
  <vcupin vcpu='13' cpuset='31' />
  <vcupin vcpu='14' cpuset='8' />
  <vcupin vcpu='15' cpuset='32' />
  <vcupin vcpu='16' cpuset='9' />
  <vcupin vcpu='17' cpuset='33' />
  <vcupin vcpu='18' cpuset='10' />
  <vcupin vcpu='19' cpuset='34' />
  <vcupin vcpu='20' cpuset='11' />
  <vcupin vcpu='21' cpuset='35' />
  <vcupin vcpu='22' cpuset='12' />
  <vcupin vcpu='23' cpuset='36' />
  <vcupin vcpu='24' cpuset='13' />
  <vcupin vcpu='25' cpuset='37' />
  <vcupin vcpu='26' cpuset='14' />
  <vcupin vcpu='27' cpuset='38' />
  <vcupin vcpu='28' cpuset='15' />
  <vcupin vcpu='29' cpuset='39' />
  <vcupin vcpu='30' cpuset='16' />
  <vcupin vcpu='31' cpuset='40' />
  <vcupin vcpu='32' cpuset='17' />
  <vcupin vcpu='33' cpuset='41' />
  <vcupin vcpu='34' cpuset='18' />
  <vcupin vcpu='35' cpuset='42' />
  <vcupin vcpu='36' cpuset='19' />
  <vcupin vcpu='37' cpuset='43' />
  <vcupin vcpu='38' cpuset='20' />
  <vcupin vcpu='39' cpuset='44' />
  <vcupin vcpu='40' cpuset='21' />
  <vcupin vcpu='41' cpuset='45' />
  <vcupin vcpu='42' cpuset='22' />
  <vcupin vcpu='43' cpuset='46' />

```

```
<vcupin vcpu='44' cpuset='23' />
<vcupin vcpu='45' cpuset='47' />
<emulatorpin cpuset="0,24" />
</cputune>
<cpu mode='host-model'>
  <model fallback='allow' />
  <topology sockets='1' cores='23' threads='2' />
</cpu>
```

Next is the **smbios** section where many important configuration parameters are passed to the VSR VM. Several settings can be defined here in place of the bof.cfg file on cf3.

```
<sysinfo type='smbios'>
  <system>
    <entry name='product'>TIMOS: chassis=VSR-I slot=A card=cpm-v mda/1=m20-v mda/
2=isa-ms-v static-route=172.16.0.0/8@172.16.36.1 address=172.16.37.71/23@active primary-config=
cf3:/config.cfg license-file=cf3:/license.txt</entry>
  </system>
</sysinfo>
```

Next are the standard OS and clock features:

```
<os>
  <type arch='x86_64' machine='pc'>hvm</type>
  <boot dev='hd' />
  <smbios mode='sysinfo' />
</os>
<clock offset='utc'>
  <timer name='pit' tickpolicy='delay' />
  <timer name='rtc' tickpolicy='catchup' />
  <timer name='hpet' present='no' />
</clock>
```

The final section contains devices and starts with the disks, serial, and console ports:

```
<devices>
  <emulator>/usr/libexec/qemu-kvm</emulator>
  <disk type='file' device='disk'>
    <driver name='qemu' type='qcow2' cache='none' />
    <source file='/var/lib/libvirt/images/vsr-a.qcow2' />
    <target dev='hda' bus='virtio' />
  </disk>
  <disk type='file' device='disk'>
    <driver name='qemu' type='qcow2' cache='none' />
    <source file='/var/lib/libvirt/images/cf1.qcow2' />
    <target dev='hdb' bus='virtio' />
  </disk>
  <serial type='pty'>
    <source path='/dev/pts/1' />
    <target port='0' />
    <alias name='serial0' />
  </serial>
  <console type='pty' tty='/dev/pts/1'>
    <source path='/dev/pts/1' />
    <target type='serial' port='0' />
    <alias name='serial0' />
  </console>
```

In this case, the configuration has a separate qcow2 file for cf1. This file can be used to back up SR OS files before an upgrade, as the cf3 qcow2 file gets replaced during the upgrade. The devices section also contains the network ports (PCI devices):

```
<interface type='bridge'>
  <source bridge='br0' />
  <model type='virtio' />
</interface>
<hostdev mode='subsystem' type='pci' managed='yes'>
  <source>
    <address domain='0x0000' bus='0x17' slot='0x00' function='0x0' />
  </source>
  <rom bar='off' />
</hostdev>
<hostdev mode='subsystem' type='pci' managed='yes'>
  <source>
    <address domain='0x0000' bus='0x17' slot='0x00' function='0x1' />
  </source>
  <rom bar='off' />
</hostdev>
<hostdev mode='subsystem' type='pci' managed='yes'>
  <source>
    <address domain='0x0000' bus='0xb3' slot='0x00' function='0x0' />
  </source>
  <rom bar='off' />
</hostdev>
<hostdev mode='subsystem' type='pci' managed='yes'>
  <source>
    <address domain='0x0000' bus='0xb3' slot='0x00' function='0x1' />
  </source>
  <rom bar='off' />
</hostdev>
<hostdev mode='subsystem' type='pci' managed='yes'>
  <source>
    <address domain='0x0000' bus='0x6a' slot='0x00' function='0x1' />
  </source>
  <rom bar='off' />
</hostdev>
```

The first network port is assigned to the SR OS management port, the next to port 1/1/1, then 1/1/2, and so on. The Mellanox ports are assigned to ports 1/1/1 to 1/1/3 using the PCI addresses obtained with the **lshw** command.

Dell server

Server model and motherboard

The server model can be identified with the **dmidecode** command:

```
[root@hyp56 ~]# dmidecode -t 2
# dmidecode 3.2
Getting SMBIOS data from sysfs.
SMBIOS 2.8 present.

Handle 0x0200, DMI type 2, 8 bytes
Base Board Information
    Manufacturer: Dell Inc.
    Product Name: 072T6D
```

```
Version: A06  
Serial Number: .DJ2XHH2.CN7793173D00E3.
```

This product name corresponds to a Dell PowerEdge R730

CPU

The **lscpu** command gives the number of sockets, cores, threads, and their numbering scheme.

With hyperthreading disabled:

```
[root@hyp56 ~]# lscpu  
Architecture:          x86_64  
CPU op-mode(s):      32-bit, 64-bit  
Byte Order:           Little Endian  
CPU(s):              44  
On-line CPU(s) list: 0-43  
Thread(s) per core:  1  
Core(s) per socket:  22  
Socket(s):           2  
NUMA node(s):        2  
Vendor ID:            GenuineIntel  
CPU family:          6  
Model:               79  
Model name:           Intel(R) Xeon(R) CPU E5-2699A v4 @ 2.40GHz  
Stepping:            1  
CPU MHz:             1200.000  
CPU max MHz:         3000.0000  
CPU min MHz:         1200.0000  
BogoMIPS:            4799.90  
Virtualization:      VT-x  
L1d cache:           32K  
L1i cache:           32K  
L2 cache:            256K  
L3 cache:            56320K  
NUMA node0 CPU(s):  0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42  
NUMA node1 CPU(s):  1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,41,43  
...
```

This system has two CPU sockets, each CPU has 22 cores, and hyperthreading is disabled, resulting in 44 vCPUs.

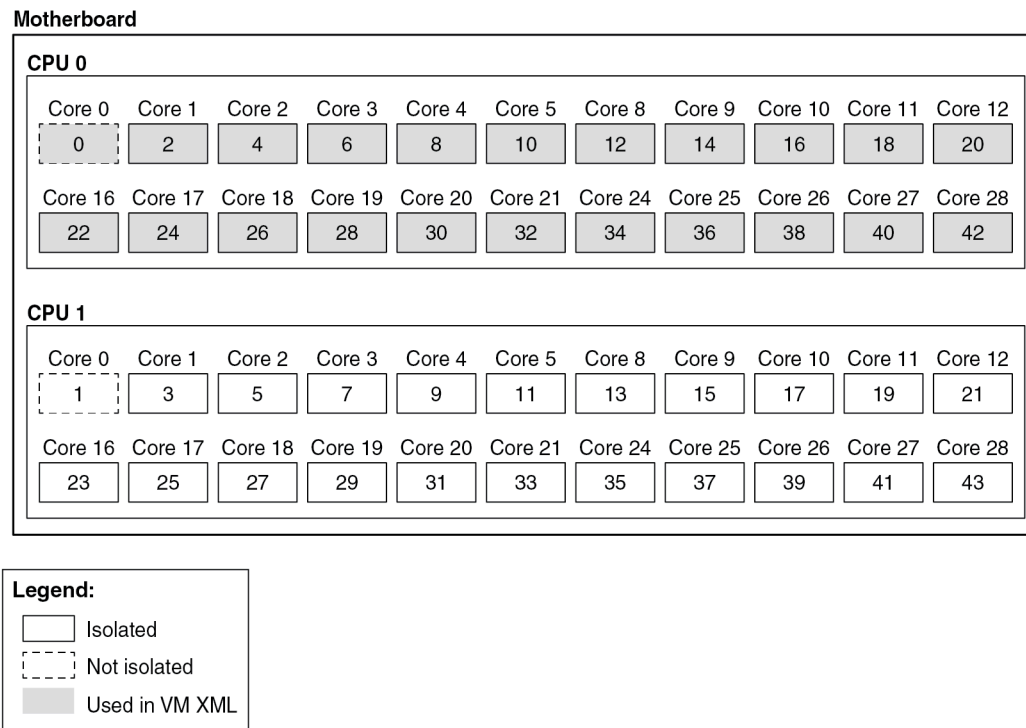
Use the **cat /proc/cpuinfo** command to confirm the CPU numbering scheme and see that all even number CPUs (listed as "processors" in the command output) are on NUMA node 0 (listed as "physical id" in the command output), while all odd number CPUs are on NUMA node 1. As a result, all even number CPUs can be assigned to one VSR VM, while all odd number CPUs must be assigned to a different VM.

```
[root@hyp56 ~]# cat /proc/cpuinfo | egrep "processor|physical id|core id"  
processor      : 0  
physical id   : 0  
core id       : 0  
processor      : 1  
physical id   : 1  
core id       : 0  
processor      : 2  
physical id   : 0  
core id       : 1  
processor      : 3  
physical id   : 1  
core id       : 1
```

```
processor      : 4
physical id   : 0
core id       : 2
processor      : 5
physical id   : 1
core id       : 2
processor      : 6
physical id   : 0
core id       : 3
processor      : 7
physical id   : 1
core id       : 3
...
```

Figure 377: Numbering scheme Dell server without hyperthreading shows the numbering scheme. Only NUMA node 0 (CPU 0) CPUs are used for VM functions in this example, but CPU isolation is also configured on NUMA node 1 for a possible identical future VSR VM:

Figure 377: Numbering scheme Dell server without hyperthreading



38115

With hyperthreading enabled:

```
[root@hyp56 admin]# lscpu
Architecture:      x86_64
CPU op-mode(s):   32-bit, 64-bit
Byte Order:       Little Endian
CPU(s):           88
On-line CPU(s) list: 0-87
Thread(s) per core: 2
Core(s) per socket: 22
Socket(s):        2
```

```

NUMA node(s):      2
Vendor ID:         GenuineIntel
CPU family:        6
Model:             79
Model name:        Intel(R) Xeon(R) CPU E5-2699A v4 @ 2.40GHz
Stepping:          1
CPU MHz:           1199.853
CPU max MHz:       3000.0000
CPU min MHz:       1200.0000
BogoMIPS:          4800.22
Virtualization:    VT-x
L1d cache:         32K
L1i cache:         32K
L2 cache:          256K
L3 cache:          56320K
NUMA node0 CPU(s):
 0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,58,60,62,64,
66,68,70,72,74,76,78,80,82,84,86
NUMA node1 CPU(s):
 1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,41,43,45,47,49,51,53,55,57,59,61,63,65,
67,69,71,73,75,77,79,81,83,85,87
...

```

The system has 88 CPUs across two sockets, with all even-number CPUs belonging to NUMA node 0 and all odd-number CPUs belonging to NUMA node 1. The numbering scheme is such that the additional threads start at number 44 and continue with even-number CPUs on NUMA 0 and odd-number CPUs on NUMA 1:

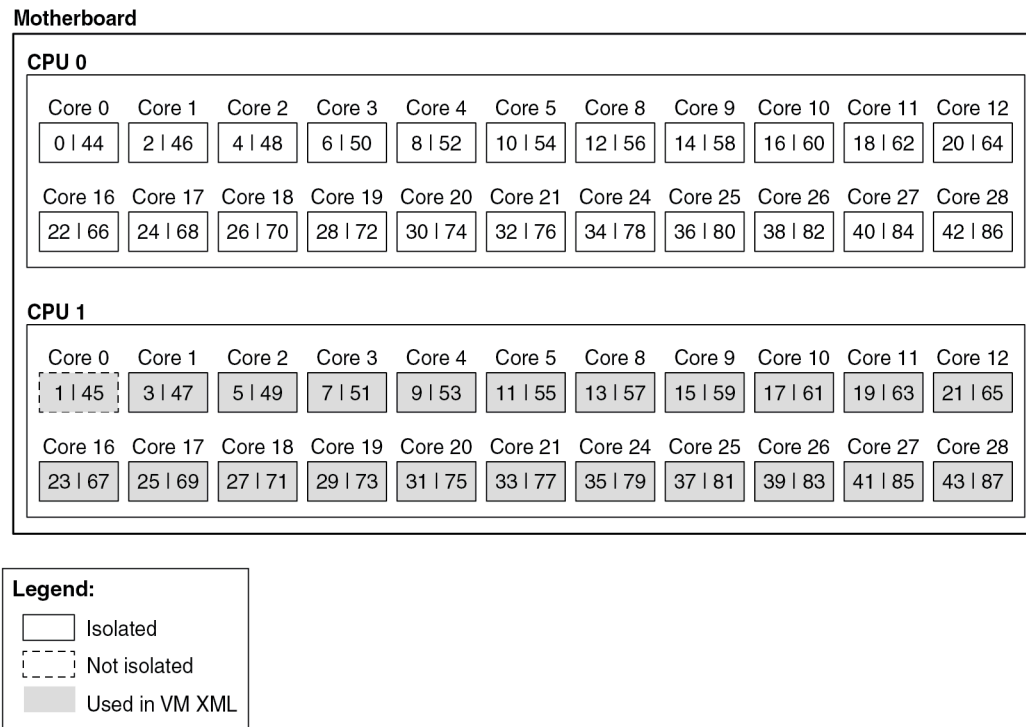
```

[root@hyp56 admin]# cat /proc/cpuinfo | egrep "processor|physical id|core id"
processor      : 0
physical id   : 0
core id       : 0
processor      : 1
physical id   : 1
core id       : 0
...
processor      : 44
physical id   : 0
core id       : 0
processor      : 45
physical id   : 1
core id       : 0
...
processor      : 86
physical id   : 0
core id       : 28
processor      : 87
physical id   : 1
core id       : 28

```

Figure 378: Numbering scheme Dell server with hyperthreading shows this numbering scheme. Only NUMA node 1 CPUs are used for the example VM with hyperthreading enabled. NUMA node 0 CPUs are isolated for future use with another identical VSR VM.

Figure 378: Numbering scheme Dell server with hyperthreading



38116

Network ports

This server has the following network ports:

```
[root@hyp56 ~]# lshw -c network -businfo
Bus info          Device          Class           Description
-----
pci@0000:04:00.0  p6p1            network        MT27700 Family [ConnectX-4]
pci@0000:04:00.1  p6p2            network        MT27700 Family [ConnectX-4]
pci@0000:01:00.0  em1             network        Ethernet Controller 10-Gigabit X540-AT2
pci@0000:01:00.1  em2             network        Ethernet Controller 10-Gigabit X540-AT2
pci@0000:06:00.0  p5p1            network        Ethernet Controller X710 for 10GbE SFP+
pci@0000:06:00.1  p5p2            network        Ethernet Controller X710 for 10GbE SFP+
pci@0000:06:00.2  p5p3            network        Ethernet Controller X710 for 10GbE SFP+
pci@0000:06:00.3  p5p4            network        Ethernet Controller X710 for 10GbE SFP+
pci@0000:09:00.0  em3             network        I350 Gigabit Network Connection
pci@0000:09:00.1  em4             network        I350 Gigabit Network Connection
pci@0000:81:00.0  network        network        Ethernet 10G 2P X520 Adapter
pci@0000:81:00.1  network        network        Ethernet 10G 2P X520 Adapter
pci@0000:83:00.0  p4p1            network        MT27700 Family [ConnectX-4]
pci@0000:83:00.1  p4p2            network        MT27700 Family [ConnectX-4]
pci@0000:83:00.2  p4p1_0          network        MT27700 Family [ConnectX-4 Virtual Function]
pci@0000:83:00.4  p4p2_0          network        MT27700 Family [ConnectX-4 Virtual Function]
pci@0000:85:00.0  p3p1            network        Ethernet 10G 2P X520 Adapter
pci@0000:85:00.1  p3p2            network        Ethernet 10G 2P X520 Adapter
pci@0000:87:00.0  p2p1            network        Ethernet 10G 2P X520 Adapter
pci@0000:87:00.1  p2p2            network        Ethernet 10G 2P X520 Adapter
```

```
br-mgmt      network      Ethernet interface
virbr0-nic   network      Ethernet interface
virbr0       network      Ethernet interface
vnet0        network      Ethernet interface
```

Because this server has two NUMA nodes, ensure that the NICs allocated to a VSR VM are on the same NUMA node as the VM's CPUs. Carefully reviewing the Mellanox ConnectX-4 NIC 0000:04:00 ports shows that this NIC is on NUMA node 0:

```
[root@hyp56 ~]# lspci -v -s 04:00.0 | egrep "Ethernet|NUMA"
04:00.0 Ethernet controller: Mellanox Technologies MT27700 Family [ConnectX-4]
Flags: bus master, fast devsel, latency 0, IRQ 149, NUMA node 0
```

However, the Mellanox ConnectX-4 NIC 0000:83:00 is on NUMA node 1:

```
[root@hyp56 ~]# lspci -v -s 83:00.0 | egrep "Ethernet|NUMA"
83:00.0 Ethernet controller: Mellanox Technologies MT27700 Family [ConnectX-4]
Flags: bus master, fast devsel, latency 0, IRQ 456, NUMA node 1
```

The Mellanox NIC on NUMA node 1 has two VFs that can be used for SR-IOV.

Kernel parameters

The following kernel parameters include those required for CPU isolation based on the installed CPU's numbering scheme and a sufficient number of hugepages to accommodate the memory usage of all VSR VMs. With hyperthreading disabled:

```
[root@hyp56 ~]# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.10.0-1160.66.1.el7.x86_64 root=/dev/mapper/centos-root ro crashkernel=
auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet pci=realloc pcie_aspm=off iommu=
pt intel_iommu=on nopat hugepagesz=1G default_hugepagesz=1G hugepages=100 kvm_intel.ple_gap=0
isolcpus=2-43 selinux=0 audit=0 LANG=en_US.UTF-8
```

On this server, all CPUs are isolated except for the first CPU on each socket, CPUs 0 and 1. These two CPUs will be used for the emulatorpin in the XML file. Having 100 hugepages on a two-NUMA node results in 50 hugepages per NUMA node and therefore 50GB available to a VSR VM.

With hyperthreading enabled, the **isolcpus** parameter must be:

```
isolcpus=2-43,46-87
```

XML file for hyperthreading disabled and NUMA node 0

The XML file header contains the VM name:

```
<domain type='kvm'>
  <name>vsr56-1</name>
```

The UUID is removed in order to generate a new UUID. The new UUID can be obtained with the **virsh dumpxml** command after running the VM and then pasted into the XML file to preserve it.

```
<!-- UUID: remove to auto-generate a new UUID -->
```


Configure hugepages and allocate 64 GB of RAM to the VSR VM. Here, the nodeset is always set to 0:

```
<memory unit="G">64</memory>
<memoryBacking>
  <hugepages>
    <page size="1" unit="G" nodeset="0"/>
  </hugepages>
  <nosharepages/>
</memoryBacking>
```

Configure the numatune setting to ensure all the RAM is allocated from NUMA node 0, the NUMA used for all resources on this VM:

```
<numatune>
  <memory mode='strict' nodeset='0' />
</numatune>
```

Configure CPU features and CPU mode. Host-model with fallback=allow is recommended:

```
<features>
  <acpi/>
</features>
<cpu mode='host-model'>
  <model fallback='allow' />
</cpu>
```

Configure the CPU pinning, with the emulatorpin set to a non-isolated CPU on the same NUMA node as the VM:

```
<vcpu placement='static'>21</vcpu>
<cputune>
  <vcpupin vcpu='0' cpuset='2' />
  <vcpupin vcpu='1' cpuset='4' />
  <vcpupin vcpu='2' cpuset='6' />
  <vcpupin vcpu='3' cpuset='8' />
  <vcpupin vcpu='4' cpuset='10' />
  <vcpupin vcpu='5' cpuset='12' />
  <vcpupin vcpu='6' cpuset='14' />
  <vcpupin vcpu='7' cpuset='16' />
  <vcpupin vcpu='8' cpuset='18' />
  <vcpupin vcpu='9' cpuset='20' />
  <vcpupin vcpu='10' cpuset='22' />
  <vcpupin vcpu='11' cpuset='24' />
  <vcpupin vcpu='12' cpuset='26' />
  <vcpupin vcpu='13' cpuset='28' />
  <vcpupin vcpu='14' cpuset='30' />
  <vcpupin vcpu='15' cpuset='32' />
  <vcpupin vcpu='16' cpuset='34' />
  <vcpupin vcpu='17' cpuset='36' />
  <vcpupin vcpu='18' cpuset='38' />
  <vcpupin vcpu='19' cpuset='40' />
  <vcpupin vcpu='20' cpuset='42' />
  <emulatorpin cpuset='0' />
</cputune>
```

Configure the OS features:

```
<os>
  <type arch='x86_64' machine='pc'>hvm</type>
  <boot dev='hd' />
</os>
```

```
<smbios mode='sysinfo' />
</os>
```

Configure SMBIOS. The **control-cpu-cores** and **vsr-deployment-model=high-packet-touch** parameters must only be enabled when required for the specific VSR deployment (see the *VSR VNF Installation and Setup Guide* and the *SR OS Software Release Notes*):

```
<sysinfo type='smbios'>
  <system>
    <entry name='product'>TiMOS:
      address=172.16.224.192/24@active \
      static-route=172.16.0.0/8@172.16.224.1 \
      license-file=ftp://user:password@172.16.224.10/license/VSR_license.txt \
      primary-config=ftp://user:pass@172.16.29.91/kvm/hyp56/vsr56-1.cfg \
      chassis=vsr-i \
      slot=A \
        card=cpm-v \
      slot=1 \
        card=iom-v \
          mda/1=m20-v \
          mda/2=isa-ms-v \
      system-base-mac=fa:ac:ff:ff:54:00 \
      <!-- control-cpu-cores=2 \ -->
      <!-- vsr-deployment-model=high-packet-touch \ -->
    </entry>
  </system>
</sysinfo>
```

Configure the clock:

```
<clock offset='utc'>
  <timer name='pit' tickpolicy='delay' />
  <timer name='rtc' tickpolicy='catchup' />
  <timer name='hpet' present='no' />
</clock>
```

The devices configuration section includes disks, network interfaces, and console ports. By default, a VSR is configured with cf3:

```
<devices>
  <emulator>/usr/libexec/qemu-kvm</emulator>
  <disk type='file' device='disk'>
    <driver name='qemu' type='qcow2' cache='none' />
    <source file='/var/lib/libvirt/images/vsr56-1.qcow2' />
    <target dev='hda' bus='virtio' />
  </disk>
```

Configure network interfaces starting with a management port attached to a Linux bridge and two PCI-PT network interfaces on NUMA node 0:

```
<interface type='bridge'>
  <source bridge='br-mgmt' />
  <model type='virtio' />
  <target dev='vsr1-mgmt' />
</interface>

<hostdev mode='subsystem' type='pci' managed='yes'>
  <source>
    <address domain='0x0000' bus='0x04' slot='0x00' function='0x0' />
  </source>
```

```
<rom bar='off' />
</hostdev>

<hostdev mode='subsystem' type='pci' managed='yes'>
  <source>
    <address domain='0x0000' bus='0x04' slot='0x00' function='0x1' />
  </source>
  <rom bar='off' />
</hostdev>
```

Configure the console port, that is accessible using the **virsh console <vm>** command:

```
<console type='pty' tty='/dev/pts/1'>
  <source path='/dev/pts/1' />
  <target type='serial' port='0' />
  <alias name='serial0' />
</console>
```

The end of the XML file includes the required **seclabel** configuration:

```
</devices>
<seclabel type='none' />
</domain>
```

The VSR's boot messages can be checked to confirm the correct CPU, memory, and NIC assignments; for example:

```
...
KVM based vcpu
Running in a KVM/QEMU virtual machine
ACPI: found 21 cores, 21 enabled
1 virtio net device is detected
2 MLX5 devices detected (2 pci passthrough, 0 SR-IOV virtual function)
...
```

The VSR automatically allocates CPUs between different task types:

```
A:v5r56-1-kvm# show card 1 virtual fp

=====
Card 1 Virtual Forwarding Plane Statistics
=====
Task                vCPUs    Average      Maximum
                   vCPUs    Utilization  Utilization
-----
NIC                  1         0.00 %      0.00 %
Worker              17         0.03 %      0.03 %
Scheduler            1         0.00 %      0.00 %
=====
```

Verify the CPU pinning with the **virsh vcpuinfo** command and ensure that no other VMs are sharing the VSR's resources:

```
[root@hyp56 ~]# virsh vcpuinfo vsr56-1
VCPU:          0
CPU:           2
State:         running
CPU time:      30.6s
CPU Affinity:  --y-----
```

```

VCPU:      1
CPU:       4
State:     running
CPU Affinity:  ----y-----

VCPU:      2
CPU:       6
State:     running
CPU Affinity:  -----y-----

...
VCPU:      20
CPU:       42
State:     running
CPU Affinity:  -----y-
    
```

XML file for hyperthreading enabled and NUMA node 1

The XML file header contains the VM name:

```

<domain type='kvm'>
  <name>vsr56-1</name>
    
```

The UUID is removed in order to generate a new UUID. The new UUID can be obtained with the **virsh dumpxml** command after running the VM and then pasted into the XML file to preserve it.

```

<!-- UUID: remove to auto-generate a new UUID -->
    
```

Define hugepages and allocate 64 GB of RAM to the VSR VM. Here, the nodeset is always set to 0:

```

<memoryBacking>
  <hugepages>
    <page size="1" unit="G" nodeset="0"/>
  </hugepages>
  <nosharepages/>
</memoryBacking>
    
```

Configure the numatune setting to ensure that all the RAM is allocated from NUMA node 1, the NUMA used for all resources on this VM:

```

<numatune>
  <memory mode='strict' nodeset='1' />
</numatune>
    
```

Configure the CPU features and CPU mode. The **host-model** with **fallback=allow** is recommended. Because hyperthreading is enabled, the following topology is required:

```

<features>
  <acpi/>
</features>
<cpu mode='host-model'>
  <model fallback='allow' />
  <!-- topology is required when hyperthreading is enabled -->
  <topology sockets='1' cores='21' threads='2' />
</cpu>
    
```

Configure the CPU pinning, with the `emulatorpin` set to non-isolated CPUs on the same NUMA as the VM:

```
<vcpu placement='static'>42</vcpu>
<cputune>
  <vcpupin vcpu='0' cpuset='3' />
  <vcpupin vcpu='1' cpuset='47' />
  <vcpupin vcpu='2' cpuset='5' />
  <vcpupin vcpu='3' cpuset='49' />
  <vcpupin vcpu='4' cpuset='7' />
  <vcpupin vcpu='5' cpuset='51' />
  <vcpupin vcpu='6' cpuset='9' />
  <vcpupin vcpu='7' cpuset='53' />
  <vcpupin vcpu='8' cpuset='11' />
  <vcpupin vcpu='9' cpuset='55' />
  <vcpupin vcpu='10' cpuset='13' />
  <vcpupin vcpu='11' cpuset='57' />
  <vcpupin vcpu='12' cpuset='15' />
  <vcpupin vcpu='13' cpuset='59' />
  <vcpupin vcpu='14' cpuset='17' />
  <vcpupin vcpu='15' cpuset='61' />
  <vcpupin vcpu='16' cpuset='19' />
  <vcpupin vcpu='17' cpuset='63' />
  <vcpupin vcpu='18' cpuset='21' />
  <vcpupin vcpu='19' cpuset='65' />
  <vcpupin vcpu='20' cpuset='23' />
  <vcpupin vcpu='21' cpuset='67' />
  <vcpupin vcpu='22' cpuset='25' />
  <vcpupin vcpu='23' cpuset='69' />
  <vcpupin vcpu='24' cpuset='27' />
  <vcpupin vcpu='25' cpuset='71' />
  <vcpupin vcpu='26' cpuset='29' />
  <vcpupin vcpu='27' cpuset='73' />
  <vcpupin vcpu='28' cpuset='31' />
  <vcpupin vcpu='29' cpuset='75' />
  <vcpupin vcpu='30' cpuset='33' />
  <vcpupin vcpu='31' cpuset='77' />
  <vcpupin vcpu='32' cpuset='35' />
  <vcpupin vcpu='33' cpuset='79' />
  <vcpupin vcpu='34' cpuset='37' />
  <vcpupin vcpu='35' cpuset='81' />
  <vcpupin vcpu='36' cpuset='39' />
  <vcpupin vcpu='37' cpuset='83' />
  <vcpupin vcpu='38' cpuset='41' />
  <vcpupin vcpu='39' cpuset='85' />
  <vcpupin vcpu='40' cpuset='43' />
  <vcpupin vcpu='41' cpuset='87' />
  <emulatorpin cpuset='1,45' />
</cputune>
```

Configure the OS features:

```
<os>
  <type arch='x86_64' machine='pc'>hvm</type>
  <boot dev='hd' />
  <smbios mode='sysinfo' />
</os>
```

Configure SMBIOS. Only enable the **control-cpu-cores** and **vsr-deployment-model=high-packet-touch** parameters when required for the specific VSR deployment (see the *VSR VNF Installation and Setup*

Guide). In this example, configure **vsr-deployment-model=high-packet-touch** and additional Worker tasks will be present to take advantage of the additional threads.

```
<sysinfo type='smbios'>
  <system>
    <entry name='product'>TiMOS:
      address=172.16.224.192/24@active \
      static-route=172.16.0.0/8@172.16.224.1 \
      license-file=ftp://user:password@172.16.224.10/license/VSR_license.txt \
      primary-config=ftp://user:pass@172.16.29.91/kvm/hyp56/vsr56-1.cfg \
      chassis=vsr-i \
      slot=A \
        card=cpm-v \
      slot=1 \
        card=iom-v \
          mda/1=m20-v \
          mda/2=isa-ms-v \
        system-base-mac=fa:ac:ff:ff:54:00 \
        vsr-deployment-model=high-packet-touch \
      </entry>
    </system>
  </sysinfo>
```

Configure the clock settings:

```
<clock offset='utc'>
  <timer name='pit' tickpolicy='delay' />
  <timer name='rtc' tickpolicy='catchup' />
  <timer name='hpet' present='no' />
</clock>
```

The devices configuration section includes disks, network interfaces, and console ports. By default, a VSR is configured with cf3:

```
<devices>
  <emulator>/usr/libexec/qemu-kvm</emulator>
  <disk type='file' device='disk'>
    <driver name='qemu' type='qcow2' cache='none' />
    <source file='/var/lib/libvirt/images/vsr56-1.qcow2' />
    <target dev='hda' bus='virtio' />
  </disk>
```

Configure network interfaces starting with a management port attached to a Linux bridge, and two SR-IOV network interfaces on NUMA node 1. To configure VLAN IDs on SR-IOV interfaces, the tags are defined in the XML file and the SR OS outer tagging is set to null:

```
<interface type='bridge'>
  <source bridge='br-mgmt' />
  <model type='virtio' />
  <target dev='vsr1-mgmt' />
</interface>

<interface type='hostdev' managed='yes'>
  <mac address="00:50:56:00:54:01" />
  <source>
    <address type='pci' domain='0x0000' bus='0x83' slot='0x00' function='0x2' />
  </source>
  <vlan>
    <tag id='100' />
  </vlan>
```

```
<target dev='vsr1_port_1/1/1' />
</interface>

<interface type='hostdev' managed='yes'>
  <mac address="00:50:56:00:54:02" />
  <source>
    <address type='pci' domain='0x0000' bus='0x83' slot='0x00' function='0x4' />
  </source>
  <vlan>
    <tag id='200' />
  </vlan>
  <target dev='vsr1_port_1/1/2' />
</interface>
```

Configure the console port, that is accessible using the **virsh console <vm>** command:

```
<console type='pty' tty='/dev/pts/1'>
  <source path='/dev/pts/1' />
  <target type='serial' port='0' />
  <alias name='serial0' />
</console>
```

The end of the XML file includes the required **seclabel** configuration:

```
</devices>
  <seclabel type='none' />
</domain>
```

The VSR's boot messages can be checked to confirm the correct CPU, memory, and NIC assignments; for example:

```
...
KVM based vcpu
Running in a KVM/QEMU virtual machine
ACPI: found 42 cores, 42 enabled
1 virtio net device is detected
2 MLX5 devices detected (0 pci passthrough, 2 SR-IOV virtual function)
...
```

The VSR automatically allocates CPUs between different task types, with more tasks than in the non-hyperthreaded case when hyperthreading gives a performance advantage:

```
A:v56-1-kvm# show card 1 virtual fp

=====
Card 1 Virtual Forwarding Plane Statistics
=====
Task                vCPUs    Average      Maximum
                   vCPUs    Utilization  Utilization
-----
NIC                  1         0.00 %      0.00 %
Worker              34         0.04 %      0.04 %
Scheduler            1         0.00 %      0.00 %
=====
A:v56-1-kvm#
```

Verify the CPU pinning with the **virsh vcpuinfo** command and ensure that no other VMs are sharing the VSR's resources:

```
[root@hyp56 ~]# virsh vcpuinfo vsr56-1
VCPU:      0
CPU:       3
State:     running
CPU time:  45.5s
CPU Affinity: ---y-----
-----

VCPU:      1
CPU:       47
State:     running
CPU time:  5.8s
CPU Affinity: -----y-----
-----

VCPU:      2
CPU:       5
State:     running
CPU time:  5.8s
CPU Affinity: -----y-----
-----

VCPU:      3
CPU:       49
State:     running
CPU time:  5.8s
CPU Affinity: -----y-----
-----

...
VCPU:      40
CPU:       43
State:     running
CPU time:  47.7s
CPU Affinity: -----y-----
-----

VCPU:      41
CPU:       87
State:     running
CPU time:  48.1s
CPU Affinity: -----y
-----
```

HPE server

Server model and motherboard

The server model can be identified with the **dmidecode** command:

```
[root@hyp62 ~]# dmidecode -t 2
# dmidecode 3.2
Getting SMBIOS data from sysfs.
SMBIOS 2.8 present.

Handle 0x0028, DMI type 2, 17 bytes
```



```
Base Board Information
  Manufacturer: HP
  Product Name: ProLiant DL380 Gen9
  Version: Not Specified
  Serial Number: MXQ724142B
  ...
```

This server's product name is HPE ProLiant DL380 Gen9.

CPU

The `lscpu` command gives the number of sockets, cores and threads and their numbering scheme. With hyperthreading disabled:

```
[root@hyp62 ~]# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
CPU(s):                44
On-line CPU(s) list:   0-43
Thread(s) per core:    1
Core(s) per socket:    22
Socket(s):              2
NUMA node(s):          2
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  79
Model name:             Intel(R) Xeon(R) CPU E5-2699A v4 @ 2.40GHz
Stepping:               1
CPU MHz:                1200.000
CPU max MHz:           2400.0000
CPU min MHz:           1200.0000
BogoMIPS:               4794.53
Virtualization:        VT-x
L1d cache:              32K
L1i cache:              32K
L2 cache:               256K
L3 cache:               56320K
NUMA node0 CPU(s):     0-10,22-32
NUMA node1 CPU(s):     11-21,33-43
...
```

The server has 44 cores with CPUs with number 0 through 10 and 22 through 32 on NUMA node 0, and CPUs with number 11 through 21 and 33 through 43 on NUMA node 1. Note the offset numbering scheme.

```
processor      : 0
physical id    : 0
core id        : 0
processor      : 1
physical id    : 0
core id        : 2
processor      : 2
physical id    : 0
core id        : 4
processor      : 3
physical id    : 0
core id        : 8
processor      : 4
physical id    : 0
core id        : 10
```

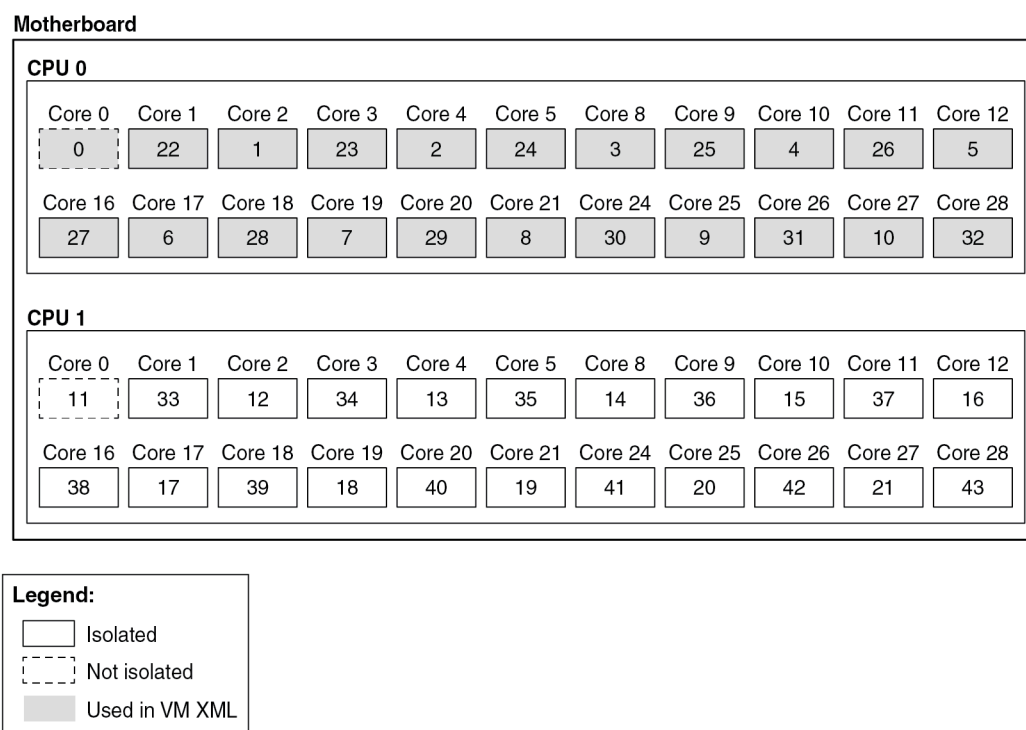
```

...
processor      : 22
physical id    : 0
core id       : 1
processor      : 23
physical id    : 0
core id       : 3
...

```

Figure 379: Numbering scheme HPE server without hyperthreading shows this numbering scheme. Only NUMA node 0 CPUs are used for VM functions in the example XML file with hyperthreading disabled. NUMA node 1 CPUs are also isolated for future use with an identical VM.

Figure 379: Numbering scheme HPE server without hyperthreading



38117

With hyperthreading enabled:

```

[admin@hyp62 ~]$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
CPU(s):                 88
On-line CPU(s) list:   0-87
Thread(s) per core:    2
Core(s) per socket:    22
Socket(s):              2
NUMA node(s):          2
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  79
Model name:             Intel(R) Xeon(R) CPU E5-2699A v4 @ 2.40GHz

```

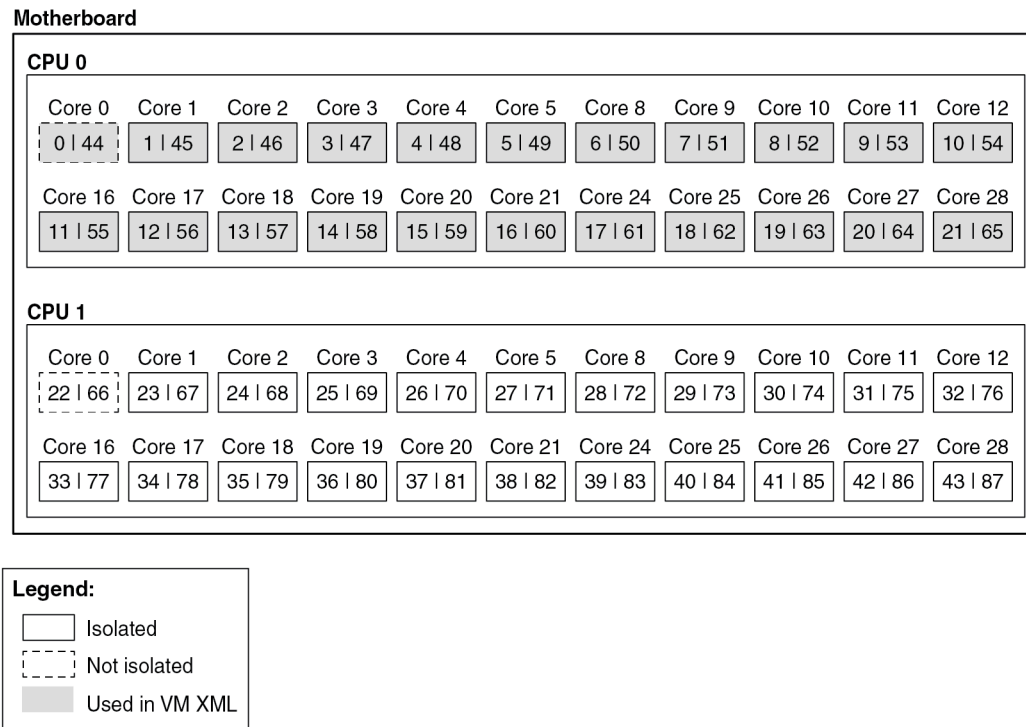
```
Stepping:          1
CPU MHz:           1200.000
CPU max MHz:      2400.0000
CPU min MHz:      1200.0000
BogoMIPS:         4794.48
Virtualization:   VT-x
L1d cache:        32K
L1i cache:        32K
L2 cache:         256K
L3 cache:         56320K
NUMA node0 CPU(s): 0-21,44-65
NUMA node1 CPU(s): 22-43,66-87
...
```

The server has 88 cores with CPUs with number 0 through 21 and 44 through 65 on NUMA node 0, and CPUs with number 22 through 43 and 66 through 87 on NUMA node 1.

```
[admin@hyp62 ~]$ cat /proc/cpuinfo | egrep "processor|physical id|core id"
processor          : 0
physical id       : 0
core id           : 0
processor          : 1
physical id       : 0
core id           : 1
...
processor          : 44
physical id       : 0
core id           : 0
processor          : 45
physical id       : 0
core id           : 1
...
```

Figure 380: Numbering scheme HPE server with hyperthreading shows the numbering scheme. Only NUMA node 0 CPUs are used for VM functions in the hyperthreading example XML file. NUMA node 1 CPUs are isolated for future use with an identical VM.

Figure 380: Numbering scheme HPE server with hyperthreading



38118

Network ports

The server has the following network ports:

```
[root@hyp62 ~]# lshw -c network -businfo
Bus info          Device           Class            Description
-----
pci@0000:05:00.0  ens3f0           network          Ethernet Controller X710 for 10GbE SFP+
pci@0000:05:00.1  ens3f1           network          Ethernet Controller X710 for 10GbE SFP+
pci@0000:05:00.2  ens3f2           network          Ethernet Controller X710 for 10GbE SFP+
pci@0000:05:00.3  ens3f3           network          Ethernet Controller X710 for 10GbE SFP+
pci@0000:05:02.0  pci@0000:05:02.0 network          Ethernet Virtual Function 700 Series
pci@0000:04:00.0  eno49            network          82599ES 10-Gigabit SFI/SFP+ Network
Connection
pci@0000:04:00.1  eno50            network          82599ES 10-Gigabit SFI/SFP+ Network
Connection
pci@0000:02:00.0  eno1             network          NetXtreme BCM5719 Gigabit Ethernet PCIe
pci@0000:02:00.1  eno2             network          NetXtreme BCM5719 Gigabit Ethernet PCIe
pci@0000:02:00.2  eno3             network          NetXtreme BCM5719 Gigabit Ethernet PCIe
pci@0000:02:00.3  eno4             network          NetXtreme BCM5719 Gigabit Ethernet PCIe
pci@0000:81:00.0  ens6f0           network          82599ES 10-Gigabit SFI/SFP+ Network
Connection
pci@0000:81:00.1  ens6f1           network          82599ES 10-Gigabit SFI/SFP+ Network
Connection
                Mgmt-br           network          Ethernet interface
                svc-rl-tap1        network          Ethernet interface
                svc-rl-tap2        network          Ethernet interface
```

```

br-pe2-pe3      network      Ethernet interface
br-pe1-pe3      network      Ethernet interface
br-pe5-int      network      Ethernet interface
br-pe5-ext      network      Ethernet interface
br-pe3-pe4      network      Ethernet interface
br-ce1-pe1      network      Ethernet interface
alubr0          network      Ethernet interface
br-pe2-pe4      network      Ethernet interface
ovs-system      network      Ethernet interface
eno49.9         network      Ethernet interface
br-pe1-pe4      network      Ethernet interface
SIM-Mgmt-bridge network      Ethernet interface
br-pe4-pe5      network      Ethernet interface
br-ce2-pe2      network      Ethernet interface
br-pe3-pe5      network      Ethernet interface
br-mgmt         network      Ethernet interface
br-unallocated network      Ethernet interface
br-ce3-pe3      network      Ethernet interface
virbr0-nic      network      Ethernet interface
virbr0          network      Ethernet interface
vsr1-mgmt       network      Ethernet interface
svc-pat-tap     network      Ethernet interface
br-ce4-pe4      network      Ethernet interface
br-ce2-int      network      Ethernet interface
br-pe1-pe2      network      Ethernet interface
br-ce2-ext      network      Ethernet interface

```

Verify that all network ports assigned to the VSR VM belong to NUMA node 0:

```

[root@hyp62 ~]# lspci -v -s 05:00.0 | egrep "Ethernet|NUMA"
libkmod: kmod_config_parse: /etc/modprobe.d/blacklist.conf line 1: ignoring bad line starting
with 'iavf'
05:00.0 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 01)
Subsystem: Intel Corporation Ethernet Converged Network Adapter X710-4
Flags: bus master, fast devsel, latency 0, IRQ 16, NUMA node 0
[admin@hyp62 ~]$ lspci -v -s 05:00.1 | egrep "Ethernet|NUMA"
libkmod: kmod_config_parse: /etc/modprobe.d/blacklist.conf line 1: ignoring bad line starting
with 'iavf'
05:00.1 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 01)
Subsystem: Intel Corporation Ethernet Converged Network Adapter X710
Flags: bus master, fast devsel, latency 0, IRQ 16, NUMA node 0

```

All ports on the same PCI slot belong to the same NUMA node.

Kernel parameters

The following kernel parameters include those required for CPU isolation based on the installed CPU's numbering scheme and a sufficient number of hugepages to accommodate the memory usage of all VSR VMs. With hyperthreading disabled:

```

[root@hyp62 ~]# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.10.0-1160.71.1.el7.x86_64 root=/dev/mapper/centos-root ro crashkernel=
auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet pci=realloc pcie_aspm=off iommu=pt
nopat intel_iommu=on hugepagesz=1G hugepages=128 default_hugepagesz=1G selinux=0 audit=0 kvm_
intel.ple_gap=0 isolcpus=1-10,12-43

```

On this server, all CPUs are isolated except for the first CPU on each socket, CPUs 0 and 11. These two CPUs will be used for emulatorpin in the XML file. 100 hugepages on a two-NUMA node results in 50 hugepages per NUMA node and therefore 50GB available to a VSR VM.

With hyperthreading enabled the **isolcpus** parameter must be:

```
isolcpus=1-21,23-43,45-65,67-87
```

XML file for hyperthreading disabled and NUMA node 0

The XML file header contains the VM name:

```
<domain type='kvm'>  
  <name>vsr62-1</name>
```

The UUID is removed in order to generate a new UUID. The new UUID can be obtained with the **virsh dumpxml** command after running the VM and then pasted into the XML file to preserve it. Here the UUID is set, as a license is already tied to this UUID:

```
<uuid>66fdf83e-fd6b-4ce7-9ee1-ca16080e073f</uuid>
```

Define the hugepages and allocate 64 GB of RAM to the VSR VM. Here, the nodeset is always set to 0:

```
<memory unit="G">64</memory>  
<memoryBacking>  
  <hugepages>  
    <page size="1" unit="G" nodeset="0"/>  
  </hugepages>  
  <nosharepages/>  
</memoryBacking>
```

Configure the numatune setting to ensure all the RAM is allocated from NUMA node 0, the NUMA used for all resources on this VM:

```
<numatune>  
  <memory mode='strict' nodeset='0' />  
</numatune>
```

Configure the CPU features and the CPU mode; **host-model** with **fallback=allow** is recommended:

```
<features>  
  <acpi/>  
</features>  
<cpu mode='host-model'>  
  <model fallback='allow' />  
</cpu>
```

Configure the CPU pinning, with the emulatorpin set to a non-isolated CPU on the same NUMA as the VM:

```
<vcpu placement='static'>21</vcpu>  
<cputune>  
  <vcpupin vcpu='0' cpuset='22' />  
  <vcpupin vcpu='1' cpuset='1' />  
  <vcpupin vcpu='2' cpuset='23' />  
  <vcpupin vcpu='3' cpuset='2' />  
  <vcpupin vcpu='4' cpuset='24' />  
  <vcpupin vcpu='5' cpuset='3' />  
  <vcpupin vcpu='6' cpuset='25' />  
  <vcpupin vcpu='7' cpuset='4' />  
  <vcpupin vcpu='8' cpuset='26' />
```

```

<vcupin vcpu='9' cpuset='5' />
<vcupin vcpu='10' cpuset='27' />
<vcupin vcpu='11' cpuset='6' />
<vcupin vcpu='12' cpuset='28' />
<vcupin vcpu='13' cpuset='7' />
<vcupin vcpu='14' cpuset='29' />
<vcupin vcpu='15' cpuset='8' />
<vcupin vcpu='16' cpuset='30' />
<vcupin vcpu='17' cpuset='9' />
<vcupin vcpu='18' cpuset='31' />
<vcupin vcpu='19' cpuset='10' />
<vcupin vcpu='20' cpuset='32' />
<emulatorpin cpuset='0' />
</cputune>

```

Configure the OS features:

```

<os>
  <type arch='x86_64' machine='pc'>hvm</type>
  <boot dev='hd' />
  <smbios mode='sysinfo' />
</os>

```

Configure SMBIOS. The **control-cpu-cores** and **vsr-deployment-model=high-packet-touch** parameters must only be enabled when required for the specific VSR deployment (see the *VSR VNF Installation and Setup Guide*):

```

<sysinfo type='smbios'>
  <system>
    <entry name='product'>TiMOS:
      address=138.120.224.187/24@active \
      static-route=138.0.0.0/8@138.120.224.1 \
      static-route=135.0.0.0/8@138.120.224.1 \
      system-base-mac=ba:db:ee:f4:f3:3d \
      license-file=ftp://ftpuser:3LSaccess@135.121.29.91/license/VSR-I_license_22.txt \
      primary-config=ftp://anonymous:pass@135.121.29.91/kvm/hyp62/vsr62-1.cfg \
      chassis=vsr-i \
      slot=A \
        card=cpm-v \
      slot=1 \
        card=iom-v \
          mda/1=m20-v \
          mda/2=isa-bb-v \
        system-base-mac=fa:ac:ff:ff:10:00 \
      <!-- control-cpu-cores=2 \ -->
      <!-- vsr-deployment-model=high-packet-touch \ -->
    </entry>
  </system>
</sysinfo>

```

Configure the clock:

```

<clock offset='utc'>
  <timer name='pit' tickpolicy='delay' />
  <timer name='rtc' tickpolicy='catchup' />
  <timer name='hpet' present='no' />
</clock>

```

The devices configuration section includes disks, network interfaces and console ports. By default, a VSR is configured with cf3:

```
<devices>
  <emulator>/usr/libexec/qemu-kvm</emulator>
  <disk type='file' device='disk'>
    <driver name='qemu' type='qcow2' cache='none' />
    <source file='/var/lib/libvirt/images/vsr62-1.qcow2' />
    <target dev='hda' bus='virtio' />
  </disk>
```

Configure the network interfaces starting with a management port attached to a Linux bridge and two PCI-PT network interfaces on NUMA node 0:

```
<interface type='bridge'>
  <source bridge='br-mgmt' />
  <model type='virtio' />
  <target dev='vsr1-mgmt' />
</interface>

<hostdev mode='subsystem' type='pci' managed='yes'>
  <source>
    <address domain='0x0000' bus='0x05' slot='0x00' function='0x0' />
  </source>
  <rom bar='off' />
</hostdev>

<hostdev mode='subsystem' type='pci' managed='yes'>
  <source>
    <address domain='0x0000' bus='0x05' slot='0x00' function='0x1' />
  </source>
  <rom bar='off' />
</hostdev>
```

The console port is accessible using the **virsh console <vm>** command:

```
<console type='pty' tty='/dev/pts/1'>
  <source path='/dev/pts/1' />
  <target type='serial' port='0' />
  <alias name='serial0' />
</console>
```

The end of the XML file includes the required **seclabel** configuration:

```
</devices>
<!-- Seclabel: required -->
<seclabel type='none' />
</domain>
```

The VSR's boot messages can be checked to confirm the correct CPU, memory, and NIC assignments; for example:

```
...
KVM based vcpu
Running in a KVM/QEMU virtual machine
ACPI: found 21 cores, 21 enabled
1 virtio net device is detected
2 i40e devices are detected
...
```


The VSR automatically allocates CPUs between different task types:

```
A:vsr62-1# show card 1 virtual fp

=====
Card 1 Virtual Forwarding Plane Statistics
=====
Task                vCPUs      Average      Maximum
                   vCPUs      Utilization  Utilization
-----
NIC                  1           0.00 %      0.00 %
Worker              17           0.03 %      0.03 %
Scheduler            1           0.00 %      0.00 %
=====
```

Verify the CPU pinning with the **virsh vcpuinfo** command and ensure that no other VMs are sharing the VSR's resources:

```
[root@hyp62 ~]# virsh vcpuinfo vsr62-1
VCPU:          0
CPU:           22
State:         running
CPU time:      40.6s
CPU Affinity:  -----y-----

VCPU:          1
CPU:           1
State:         running
CPU time:      9.8s
CPU Affinity:  -y-----

VCPU:          2
CPU:           23
State:         running
CPU time:      145.2s
CPU Affinity:  -----y-----

VCPU:          3
CPU:           2
State:         running
CPU time:      153.4s
CPU Affinity:  --y-----
...
```

XML file for hyperthreading enabled and NUMA node 0

The XML file header contains the VM name:

```
<domain type='kvm'>
  <name>vsr62-1</name>
```

The UUID is removed in order to generate a new UUID. The new UUID can be obtained with the **virsh dumpxml** command after running the VM and then pasted into the XML file to preserve it. Here, the UUID is set, as a license is already tied to this UUID:

```
<uuid>66fdf83e-fd6b-4ce7-9ee1-ca16080e073f</uuid>
```

Define the hugepages and the allocate 64 GB of RAM to the VSR VM. Here, the nodeset is always set to 0:

```
<memory unit="G">64</memory>
<memoryBacking>
  <hugepages>
    <page size="1" unit="G" nodeset="0"/>
  </hugepages>
  <nosharepages/>
</memoryBacking>
```

Configure the numatune setting to ensure all the RAM is allocated from NUMA node 0, the NUMA used for all resources on this VM:

```
<numatune>
  <memory mode='strict' nodeset='0' />
</numatune>
```

Configure the CPU features and CPU mode; **host-model** with **fallback=allow** is recommended. Because hyperthreading is enabled, the topology is required:

```
<features>
  <acpi/>
</features>
<cpu mode='host-model'>
  <model fallback='allow' />
  <topology sockets='1' cores='21' threads='2' />
</cpu>
```

Configure the CPU pinning, with emulatorpin set to a non-isolated CPU on the same NUMA as the VM:

```
<vcpu placement='static'>42</vcpu>
<cputune>
  <vcpupin vcpu='0' cpuset='1' />
  <vcpupin vcpu='1' cpuset='45' />
  <vcpupin vcpu='2' cpuset='2' />
  <vcpupin vcpu='3' cpuset='46' />
  <vcpupin vcpu='4' cpuset='3' />
  <vcpupin vcpu='5' cpuset='47' />
  <vcpupin vcpu='6' cpuset='4' />
  <vcpupin vcpu='7' cpuset='48' />
  <vcpupin vcpu='8' cpuset='5' />
  <vcpupin vcpu='9' cpuset='49' />
  <vcpupin vcpu='10' cpuset='6' />
  <vcpupin vcpu='11' cpuset='50' />
  <vcpupin vcpu='12' cpuset='7' />
  <vcpupin vcpu='13' cpuset='51' />
  <vcpupin vcpu='14' cpuset='8' />
  <vcpupin vcpu='15' cpuset='52' />
  <vcpupin vcpu='16' cpuset='9' />
  <vcpupin vcpu='17' cpuset='53' />
  <vcpupin vcpu='18' cpuset='10' />
  <vcpupin vcpu='19' cpuset='54' />
  <vcpupin vcpu='20' cpuset='11' />
  <vcpupin vcpu='21' cpuset='55' />
  <vcpupin vcpu='22' cpuset='12' />
  <vcpupin vcpu='23' cpuset='56' />
  <vcpupin vcpu='24' cpuset='13' />
  <vcpupin vcpu='25' cpuset='57' />
  <vcpupin vcpu='26' cpuset='14' />
```

```
<vcupin vcpu='27' cpuset='58' />
<vcupin vcpu='28' cpuset='15' />
<vcupin vcpu='29' cpuset='59' />
<vcupin vcpu='30' cpuset='16' />
<vcupin vcpu='31' cpuset='60' />
<vcupin vcpu='32' cpuset='17' />
<vcupin vcpu='33' cpuset='61' />
<vcupin vcpu='34' cpuset='18' />
<vcupin vcpu='35' cpuset='62' />
<vcupin vcpu='36' cpuset='19' />
<vcupin vcpu='37' cpuset='63' />
<vcupin vcpu='38' cpuset='20' />
<vcupin vcpu='39' cpuset='64' />
<vcupin vcpu='40' cpuset='21' />
<vcupin vcpu='41' cpuset='65' />
<emulatorpin cpuset='0,44' />
</cputune>
```

Configure the OS features:

```
<os>
<type arch='x86_64' machine='pc'>hvm</type>
<boot dev='hd' />
<smbios mode='sysinfo' />
</os>
```

Configure SMBIOS. The **control-cpu-cores** and **vsr-deployment-model=high-packet-touch** parameters must only be enabled when required for the specific VSR deployment (see the *VSR VNF Installation and Setup Guide* and the *SR OS Software Release Notes*). In this example, **vsr-deployment-model=high-packet-touch** is configured and additional worker tasks will be present to take advantage of the additional threads.

```
<sysinfo type='smbios'>
<system>
<entry name='product'>TiMOS:
address=138.120.224.187/24@active \
static-route=138.0.0.0/8@138.120.224.1 \
static-route=135.0.0.0/8@138.120.224.1 \
system-base-mac=ba:db:ee:f4:f3:3d \
license-file=ftp://ftpuser:3L$access@135.121.29.91/license/VSR-I_license_22.txt \
primary-config=ftp://anonymous:pass@135.121.29.91/kvm/hyp62/vsr62-1.cfg \
chassis=vsr-i \
slot=A \
card=cpm-v \
slot=1 \
card=iom-v \
mda/1=m20-v \
mda/2=isa-bb-v \
system-base-mac=fa:ac:ff:ff:10:00 \
<!-- control-cpu-cores=2 \ -->
vsr-deployment-model=high-packet-touch \
</entry>
</system>
</sysinfo>
```

Configure the clock settings:

```
<clock offset='utc'>
<timer name='pit' tickpolicy='delay' />
<timer name='rtc' tickpolicy='catchup' />
<timer name='hpet' present='no' />
```

```
</clock>
```

The devices configuration section includes disks, network interfaces, and console ports. By default, a VSR is configured with cf3:

```
<devices>
  <emulator>/usr/libexec/qemu-kvm</emulator>
  <disk type='file' device='disk'>
    <driver name='qemu' type='qcow2' cache='none' />
    <source file='/var/lib/libvirt/images/vsr62-1.qcow2' />
    <target dev='hda' bus='virtio' />
  </disk>
```

Configure network interfaces starting with a management port attached to a Linux bridge, and two PCI-PT network interfaces on NUMA node 0:

```
<interface type='bridge'>
  <source bridge='br-mgmt' />
  <model type='virtio' />
  <target dev='vsr1-mgmt' />
</interface>

<hostdev mode='subsystem' type='pci' managed='yes'>
  <source>
    <address domain='0x0000' bus='0x05' slot='0x00' function='0x0' />
  </source>
  <rom bar='off' />
</hostdev>

<hostdev mode='subsystem' type='pci' managed='yes'>
  <source>
    <address domain='0x0000' bus='0x05' slot='0x00' function='0x1' />
  </source>
  <rom bar='off' />
</hostdev>
```

Configure the console port accessible using the **virsh console <vm>** command:

```
<console type='pty' tty='/dev/pts/1'>
  <source path='/dev/pts/1' />
  <target type='serial' port='0' />
  <alias name='serial0' />
</console>
```

The end of XML file includes the required **seclabel** configuration:

```
</devices>
<!-- Seclabel: required -->
<seclabel type='none' />
</domain>
```

The VSR's boot messages can be checked to confirm correct CPU, memory and NIC assignments; for example:

```
...
KVM based vcpu
Running in a KVM/QEMU virtual machine
ACPI: found 42 cores, 42 enabled
1 virtio net device is detected
```

```
2 i40e devices are detected
...
```

The VSR automatically allocates CPUs between different task types:

```
A:vsr62-1# show card 1 virtual fp
```

```
=====
Card 1 Virtual Forwarding Plane Statistics
=====
```

Task	vCPUs	Average Utilization	Maximum Utilization
NIC	1	0.00 %	0.00 %
Worker	34	0.03 %	0.04 %
Scheduler	1	0.00 %	0.00 %

```
=====
```

Verify the CPU pinning with the **virsh vcpuinfo** command and ensure that no other VMs are sharing the VSR's resources:

```
[root@hyp62 ~]# virsh vcpuinfo vsr62-1
```

```
VCPU:      0
CPU:       1
State:     running
CPU time:  40.5s
CPU Affinity: -y-----
-----

VCPU:      1
CPU:       45
State:     running
CPU time:  8.3s
CPU Affinity: -----y-----
-----

VCPU:      2
CPU:       2
State:     running
CPU time:  8.3s
CPU Affinity: --y-----
-----

VCPU:      3
CPU:       46
State:     running
CPU time:  8.5s
CPU Affinity: -----y-----
-----
...
```

Complete XML file example

The following example XML file is provided in an easy to read and modify format with multiple mutually exclusive options that can be commented out.

```
<domain type='kvm'>
<!-- VM name -->
```

```

<name>vsr56-1</name>

<!-- UUID: remove to auto-generate a new UUID -->
<!-- Example UUID, do not use -->
<!-- <uuid>ab9711d2-f725-4e27-8a52-ffe1873c102f</uuid> -->

<!-- VM memory allocation depends on role and required scaling, see User Guide and Release
Notes -->
<memory unit="G">64</memory>

<!-- Hugepages -->
<memoryBacking>
  <hugepages>
    <page size="1" unit="G" nodeset="0"/>
  </hugepages>
  <nosharepages/>
</memoryBacking>

<!-- Numatune: specifying NUMA node is required for systems with multiple NUMA nodes -->
<numatune>
  <memory mode='strict' nodeset='0' />
</numatune>

<!-- CPU features: ACPI is required for hyperthreading -->
<features>
  <acpi/>
</features>

<!-- CPU mode: mode='host-model' with fallback='allow' is recommended -->
<cpu mode='host-model'>
  <model fallback='allow' />
  <!-- topology is required when hyperthreading is enabled -->
  <!-- <topology sockets='1' cores='21' threads='1' /> -->
  <!-- <topology sockets='1' cores='21' threads='2' /> -->
</cpu>

<!-- CPU pinning: 'emulatorpin' is set to non-isolated CPU not allocated to VSR VM, but are on
the same CPU/NUMA -->

<!-- NUMA 0 hypertheadng disabled -->
<vcpu placement='static'>21</vcpu>
<cputune>
  <vcpupin vcpu='0' cpuset='2' />
  <vcpupin vcpu='1' cpuset='4' />
  <vcpupin vcpu='2' cpuset='6' />
  <vcpupin vcpu='3' cpuset='8' />
  <vcpupin vcpu='4' cpuset='10' />
  <vcpupin vcpu='5' cpuset='12' />
  <vcpupin vcpu='6' cpuset='14' />
  <vcpupin vcpu='7' cpuset='16' />
  <vcpupin vcpu='8' cpuset='18' />
  <vcpupin vcpu='9' cpuset='20' />
  <vcpupin vcpu='10' cpuset='22' />
  <vcpupin vcpu='11' cpuset='24' />
  <vcpupin vcpu='12' cpuset='26' />
  <vcpupin vcpu='13' cpuset='28' />
  <vcpupin vcpu='14' cpuset='30' />
  <vcpupin vcpu='15' cpuset='32' />
  <vcpupin vcpu='16' cpuset='34' />
  <vcpupin vcpu='17' cpuset='36' />
  <vcpupin vcpu='18' cpuset='38' />
  <vcpupin vcpu='19' cpuset='40' />
  <vcpupin vcpu='20' cpuset='42' />
  <emulatorpin cpuset='0' />

```

```

</cputune>

<!-- NUMA 1 hyperthreading disabled
<vcpu placement='static'>21</vcpu>
<cputune>
  <vcpupin vcpu='0' cpuset='3' />
  <vcpupin vcpu='1' cpuset='5' />
  <vcpupin vcpu='2' cpuset='7' />
  <vcpupin vcpu='3' cpuset='9' />
  <vcpupin vcpu='4' cpuset='11' />
  <vcpupin vcpu='5' cpuset='13' />
  <vcpupin vcpu='6' cpuset='15' />
  <vcpupin vcpu='7' cpuset='17' />
  <vcpupin vcpu='8' cpuset='19' />
  <vcpupin vcpu='9' cpuset='21' />
  <vcpupin vcpu='10' cpuset='23' />
  <vcpupin vcpu='11' cpuset='25' />
  <vcpupin vcpu='12' cpuset='27' />
  <vcpupin vcpu='13' cpuset='29' />
  <vcpupin vcpu='14' cpuset='31' />
  <vcpupin vcpu='15' cpuset='33' />
  <vcpupin vcpu='16' cpuset='35' />
  <vcpupin vcpu='17' cpuset='37' />
  <vcpupin vcpu='18' cpuset='39' />
  <vcpupin vcpu='19' cpuset='41' />
  <vcpupin vcpu='20' cpuset='43' />
  <emulatorpin cpuset='1' />
</cputune>
-->

<!-- NUMA 0 hyperthreading enabled
<vcpu placement='static'>42</vcpu>
<cputune>
  <vcpupin vcpu='0' cpuset='2' />
  <vcpupin vcpu='1' cpuset='46' />
  <vcpupin vcpu='2' cpuset='4' />
  <vcpupin vcpu='3' cpuset='48' />
  <vcpupin vcpu='4' cpuset='6' />
  <vcpupin vcpu='5' cpuset='50' />
  <vcpupin vcpu='6' cpuset='8' />
  <vcpupin vcpu='7' cpuset='52' />
  <vcpupin vcpu='8' cpuset='10' />
  <vcpupin vcpu='9' cpuset='54' />
  <vcpupin vcpu='10' cpuset='12' />
  <vcpupin vcpu='11' cpuset='56' />
  <vcpupin vcpu='12' cpuset='14' />
  <vcpupin vcpu='13' cpuset='58' />
  <vcpupin vcpu='14' cpuset='16' />
  <vcpupin vcpu='15' cpuset='60' />
  <vcpupin vcpu='16' cpuset='18' />
  <vcpupin vcpu='17' cpuset='62' />
  <vcpupin vcpu='18' cpuset='20' />
  <vcpupin vcpu='19' cpuset='64' />
  <vcpupin vcpu='20' cpuset='22' />
  <vcpupin vcpu='21' cpuset='66' />
  <vcpupin vcpu='22' cpuset='24' />
  <vcpupin vcpu='23' cpuset='68' />
  <vcpupin vcpu='24' cpuset='26' />
  <vcpupin vcpu='25' cpuset='70' />
  <vcpupin vcpu='26' cpuset='28' />
  <vcpupin vcpu='27' cpuset='72' />
  <vcpupin vcpu='28' cpuset='30' />
  <vcpupin vcpu='29' cpuset='74' />
  <vcpupin vcpu='30' cpuset='32' />

```

```
<vcupin vcpu='31' cpuset='76' />
<vcupin vcpu='32' cpuset='34' />
<vcupin vcpu='33' cpuset='78' />
<vcupin vcpu='34' cpuset='36' />
<vcupin vcpu='35' cpuset='80' />
<vcupin vcpu='36' cpuset='38' />
<vcupin vcpu='37' cpuset='82' />
<vcupin vcpu='38' cpuset='40' />
<vcupin vcpu='39' cpuset='84' />
<vcupin vcpu='40' cpuset='42' />
<vcupin vcpu='41' cpuset='86' />
<emulatorpin cpuset='0,44' />
</cputune>
-->

<!-- NUMA 1 hyperthreading enabled
<vcpu placement='static'>42</vcpu>
<cputune>
  <vcupin vcpu='0' cpuset='3' />
  <vcupin vcpu='1' cpuset='47' />
  <vcupin vcpu='2' cpuset='5' />
  <vcupin vcpu='3' cpuset='49' />
  <vcupin vcpu='4' cpuset='7' />
  <vcupin vcpu='5' cpuset='51' />
  <vcupin vcpu='6' cpuset='9' />
  <vcupin vcpu='7' cpuset='53' />
  <vcupin vcpu='8' cpuset='11' />
  <vcupin vcpu='9' cpuset='55' />
  <vcupin vcpu='10' cpuset='13' />
  <vcupin vcpu='11' cpuset='57' />
  <vcupin vcpu='12' cpuset='15' />
  <vcupin vcpu='13' cpuset='59' />
  <vcupin vcpu='14' cpuset='17' />
  <vcupin vcpu='15' cpuset='61' />
  <vcupin vcpu='16' cpuset='19' />
  <vcupin vcpu='17' cpuset='63' />
  <vcupin vcpu='18' cpuset='21' />
  <vcupin vcpu='19' cpuset='65' />
  <vcupin vcpu='20' cpuset='23' />
  <vcupin vcpu='21' cpuset='67' />
  <vcupin vcpu='22' cpuset='25' />
  <vcupin vcpu='23' cpuset='69' />
  <vcupin vcpu='24' cpuset='27' />
  <vcupin vcpu='25' cpuset='71' />
  <vcupin vcpu='26' cpuset='29' />
  <vcupin vcpu='27' cpuset='73' />
  <vcupin vcpu='28' cpuset='31' />
  <vcupin vcpu='29' cpuset='75' />
  <vcupin vcpu='30' cpuset='33' />
  <vcupin vcpu='31' cpuset='77' />
  <vcupin vcpu='32' cpuset='35' />
  <vcupin vcpu='33' cpuset='79' />
  <vcupin vcpu='34' cpuset='37' />
  <vcupin vcpu='35' cpuset='81' />
  <vcupin vcpu='36' cpuset='39' />
  <vcupin vcpu='37' cpuset='83' />
  <vcupin vcpu='38' cpuset='41' />
  <vcupin vcpu='39' cpuset='85' />
  <vcupin vcpu='40' cpuset='43' />
  <vcupin vcpu='41' cpuset='87' />
  <emulatorpin cpuset='1,45' />
</cputune>
-->
```



```

<!-- OS features -->
<os>
  <type arch='x86_64' machine='pc'>hvm</type>
  <boot dev='hd' />
  <smbios mode='sysinfo' />
</os>

<!-- SMBIOS configuration -->
<sysinfo type='smbios'>
  <system>
    <entry name='product'>TiMOS:
      address=138.120.224.192/24@active \
      static-route=138.0.0.0/8@138.120.224.1 \
      static-route=135.0.0.0/8@138.120.224.1 \
      license-file=ftp://ftuser:3LSaccess@172.16.29.91/license/VSR-I_license_22.txt \
      primary-config=ftp://anonymous:pass@172.16.29.91/kvm/hyp56/vsr56-1.cfg \
      chassis=vsr-i \
      slot=A \
        card=cpm-v \
      slot=1 \
        card=iom-v \
          mda/1=m20-v \
          mda/2=isa-ms-v \
        system-base-mac=fa:ac:ff:ff:10:00 \
      <!-- control-cpu-cores=2 \ -->
      <!-- vsr-deployment-model=high-packet-touch \ -->
    </entry>
  </system>
</sysinfo>

<!-- Clock features -->
<clock offset='utc'>
  <timer name='pit' tickpolicy='delay' />
  <timer name='rtc' tickpolicy='catchup' />
  <timer name='hpet' present='no' />
</clock>

<!-- Devices -->
<devices>
  <emulator>/usr/libexec/qemu-kvm</emulator>

<!-- CF3 QCOW2 -->
<disk type='file' device='disk'>
  <driver name='qemu' type='qcow2' cache='none' />
  <source file='/var/lib/libvirt/images/vsr56-1.qcow2' />
  <target dev='hda' bus='virtio' />
</disk>

<!-- Network interfaces -->

<!-- Management network: Linux bridge -->
<interface type='bridge'>
  <source bridge='br-mgmt' />
  <model type='virtio' />
  <target dev='vsr1-mgmt' />
</interface>

<!-- Linux bridge port 1 -->
<interface type='bridge'>
  <source bridge='br-mgmt' />
  <model type='virtio' />
  <target dev='vsr1-br1' />
</interface>
-->

```

```

<!-- PCI Passthrough port 1 -->
<hostdev mode='subsystem' type='pci' managed='yes'>
  <source>
    <address domain='0x0000' bus='0x04' slot='0x00' function='0x0' />
  </source>
  <rom bar='off' />
</hostdev>

<!-- SR-IOV port 1
<interface type='hostdev' managed='yes'>
  <mac address="00:50:56:00:54:01"/>
  <source>
    <address type='pci' domain='0x0000' bus='0x83' slot='0x00' function='0x2' />
  </source>
  <vlan>
    <tag id='1001' />
  </vlan>
  <target dev='vsr1_port_1/1/1' />
</interface>
-->

<!-- PCI Passthrough port 2 -->
<hostdev mode='subsystem' type='pci' managed='yes'>
  <source>
    <address domain='0x0000' bus='0x04' slot='0x00' function='0x1' />
  </source>
  <rom bar='off' />
</hostdev>

<!-- SR-IOV port 2
<interface type='hostdev' managed='yes'>
  <mac address="00:50:56:00:54:02"/>
  <source>
    <address type='pci' domain='0x0000' bus='0x83' slot='0x00' function='0x4' />
  </source>
  <vlan>
    <tag id='200' />
  </vlan>
  <target dev='vsr1_port_1/1/2' />
</interface>
-->

<!-- Console redirected to virsh console -->
<console type='pty' tty='/dev/pts/1'>
  <source path='/dev/pts/1' />
  <target type='serial' port='0' />
  <alias name='serial0' />
</console>

<!-- Console port redirected to TCP socket, port number must be unique per hypervisor
<console type='tcp'>
  <source mode='bind' host='0.0.0.0' service='2501' />
  <protocol type='telnet' />
  <target type='virtio' port='0' />
</console>
-->

</devices>

<!-- Seclabel: required -->
<seclabel type='none' />
</domain>

```

Conclusion

VSR VM configuration requires specific CPU pinning and proper assignment of dedicated system resources to achieve high performance and stability. The examples in this guide demonstrate various CPU numbering schemes and provisioning scenarios and provide an easier starting point when creating a new VSR VM.

Customer document and product support



Customer documentation

[Customer documentation welcome page](#)



Technical support

[Product support portal](#)



Documentation feedback

[Customer documentation feedback](#)