



7450 Ethernet Service Switch
7750 Service Router
7950 Extensible Routing System
Virtualized Service Router
Releases up to 24.10.R2

Unicast Routing Protocols Advanced Configuration Guide for Classic CLI

3HE 20812 AAAC TQZZA
Edition: 01
March 2025

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.

© 2025 Nokia.

Table of contents

List of tables.....	5
List of figures.....	6
Preface.....	13
Advertising IPv4 NLRI with IPv6 Next-Hop.....	14
Associating Communities with Static and Aggregate Routes.....	30
BGP Add-Path.....	54
BGP Add-Path Policy Control.....	82
BGP Autonomous System Override.....	98
BGP Conditional Route Advertisement.....	112
BGP Convergence - Delayed Route Advertisement.....	124
BGP Default Route Origination.....	138
BGP Fast Reroute.....	151
BGP Fast Reroute Policy Control.....	165
BGP FlowSpec for IPv4 and IPv6.....	183
BGP FlowSpec Route Validation.....	202
BGP Graceful Restart and Long-Lived Graceful Restart.....	218
BGP Monitoring Protocol Basics.....	247
BGP Multipath.....	257

BGP Optimal Route Reflection for Hierarchical Networks.....	285
BGP Optimal Route Reflection for Non-Hierarchical Networks.....	302
BGP Prefix Limit per Address Family.....	317
BGP Remove-Private ASN.....	327
BGP Route Leaking.....	346
BGP Route Refresh.....	380
BGP Unresolved Route Leaking from Base Router to VPRN.....	391
BGP Weighted ECMP.....	410
Dynamic BGP Peers.....	426
EBGP Default Reject Policy.....	439
EBGP Route Resolution to a Static Route.....	448
Flexible Algorithm for IS-IS.....	463
IS-IS Link Bundling.....	482
Next-Hop Resolution for Labeled BGP Routes.....	496
Policy Chaining and Logical Expressions.....	527
Pop-Label for /32 Label-IPv4 BGP Routes.....	554
Route Policy Action to Suppress BGP Route Installation.....	566
Separate BGP RIBs for Labeled Routes.....	580

List of tables

Table 1: Supported address families for GR and LLGR in base router and in VPRN.....	218
Table 2: Helper actions during GR and LLGR.....	225
Table 3: BMP Message Types.....	248
Table 4: Supported address families for BGP prefix limit.....	317
Table 5: Status of the links A, B, C, and D.....	484
Table 6: Default preferences in route table.....	503
Table 7: Policy chaining versus policy logical expressions.....	527
Table 8: Boolean values for the policy actions.....	534
Table 9: Actions for the logical operators.....	535
Table 10: Mapping the final result of an expression to a policy action.....	535
Table 11: Assigned LP and communities for the import logical expressions.....	541
Table 12: Assigned LP and communities for the import logical expressions.....	544
Table 13: Assigned LP for the import logical expressions.....	551

List of figures

Figure 1: Capability value field format.....	14
Figure 2: Example topology with IPv6 interfaces.....	16
Figure 3: Loopback addresses and advertised IPv4, label-IPv4, and VPN-IPv4 routes.....	16
Figure 4: Example topology.....	31
Figure 5: CE connections for next-hops.....	33
Figure 6: CE-7 connectivity.....	45
Figure 7: CE-6 connectivity.....	48
Figure 8: RR advertises best path only – path A preferred over path B.....	55
Figure 9: Reconvergence after path failure (without add-path).....	56
Figure 10: Advertised paths when BGP add-path is enabled in PEs and RR.....	57
Figure 11: Reconvergence after path failure when BGP add-path is enabled.....	58
Figure 12: Example topology.....	62
Figure 13: Example topology with VPRNs.....	75
Figure 14: BGP add-paths before policy control.....	83
Figure 15: BGP add-paths after policy control.....	83
Figure 16: Example topology - IPv4.....	85
Figure 17: Example topology - VPN-IPv4.....	91
Figure 18: PE-2 detects AS-path loop and advertises the route to PE-3 as invalid.....	99
Figure 19: BGP AS override replaces the peer ASN in the AS-path with the local ASN.....	99
Figure 20: Example topology.....	100
Figure 21: PE-2 detects AS loop and advertises a route to PE-3 as invalid.....	103

Figure 22: No AS loop when BGP AS override is enabled for group "eBGP" on PE-2 and PE-4.....	105
Figure 23: Example topology with VPRN 1 on all PEs.....	106
Figure 24: AS loop when BGP AS override is not configured in VPRN 1 on PE-2.....	109
Figure 25: Routes advertised when BGP AS override is enabled in VPRN 1 on the PEs.....	109
Figure 26: Conditional BGP Route Advertisement - ISP Peering.....	112
Figure 27: Conditional BGP Route Advertisement Implementation Example.....	113
Figure 28: Example Topology.....	114
Figure 29: Default SR OS behavior when the BGP process restarts.....	125
Figure 30: BGP convergence tuning with delayed route advertisement.....	125
Figure 31: BGP convergence timers.....	126
Figure 32: BGP convergence states.....	127
Figure 33: Example topology.....	128
Figure 34: Example topology with IPv4 addresses.....	139
Figure 35: Example topology with IPv6 addresses.....	140
Figure 36: Core PIC.....	152
Figure 37: Edge PIC.....	152
Figure 38: BGP FRR topology.....	153
Figure 39: Community addition on PE-1 and PE-2.....	166
Figure 40: FRR policy on PE-3.....	167
Figure 41: Example topology - IPv4.....	168
Figure 42: Example topology - VPN-IPv4.....	175
Figure 43: Example topology.....	185
Figure 44: Example Topology with FlowSpec Route Server in AS 64496.....	204

Figure 45: Topology with FlowSpec Route Server in AS 64500.....	213
Figure 46: BGP GR capability.....	220
Figure 47: LLGR capability.....	223
Figure 48: GR and LLGR.....	225
Figure 49: Example topology.....	227
Figure 50: VPRN 1 and VPLS 2 in the example topology.....	228
Figure 51: BMP Operational Overview.....	248
Figure 52: Example topology.....	259
Figure 53: BGP multipath with eBGP limit 2.....	262
Figure 54: eBGP multipath with limit 2 and ECMP disabled.....	262
Figure 55: BGP multipath with iBGP limit 3 and ECMP limit 8.....	264
Figure 56: BGP multipath with limit 6 and eBGP preferred.....	266
Figure 57: BGP multipath with limit 6, eBGP equal to iBGP, and other path options identical.....	268
Figure 58: BGP multipath configured with restriction to the same neighbor AS.....	269
Figure 59: BGP multipath restricted to the same neighbor AS: AS paths with same length.....	270
Figure 60: BGP multipath restricted to the same neighbor AS: AS paths of different lengths.....	272
Figure 61: BGP multipath restricted to the same neighbor AS: AS paths of different lengths, AS path ignored.....	273
Figure 62: BGP multipath restricted to exact same AS. All AS paths are different.....	275
Figure 63: BGP multipath restricted to exact same AS. All AS paths are identical.....	276
Figure 64: BGP multipath for the IPv4 address family.....	278
Figure 65: BGP multipath for the label-IPv4 address family.....	279
Figure 66: BGP multipath for the label-IPv6 address family.....	280

Figure 67: Best IPv4 path originates from a non-multipath-eligible BGP neighbor.....	282
Figure 68: Two IPv4 paths from multipath-eligible BGP peers are used.....	284
Figure 69: Centralized route reflection.....	286
Figure 70: Centralized route reflection with ORR.....	287
Figure 71: Example hierarchical networking using OSPF.....	289
Figure 72: Suboptimal route reflection.....	296
Figure 73: Optimal route reflection.....	300
Figure 74: Centralized route reflection.....	303
Figure 75: Centralized route reflection with ORR.....	304
Figure 76: Example non-hierarchical networking using IS-IS.....	306
Figure 77: Suboptimal route reflection.....	312
Figure 78: Optimal route reflection.....	315
Figure 79: Post-import option.....	318
Figure 80: Example topology.....	319
Figure 81: Use case 1 topology.....	328
Figure 82: PE-2 adds its ASN and keeps all ASNs in the AS path (default).....	331
Figure 83: PE-2 adds its own ASN and removes all private ASNs.....	332
Figure 84: PE-2 adds its own ASN and replaces all private ASNs with its own ASN.....	333
Figure 85: Use case 2 topology.....	334
Figure 86: PE-2 adds its own private ASN and its public ASN (default).....	335
Figure 87: PE-2 adds only its own public ASN when local ASN is configured as private.....	336
Figure 88: PE-2 removes the private ASNs until the first public ASN.....	338
Figure 89: PE-2 replaces the private ASNs until the first public ASN.....	339

Figure 90: Use case 3 topology with private ASN 64513 on CE-1 and CE-6.....	339
Figure 91: PE-2 adds its public ASN to the AS path.....	341
Figure 92: PE-2 removes the private ASNs except peer ASN 64513.....	342
Figure 93: PE-2 replaces the private ASNs except peer ASN 64513.....	343
Figure 94: BGP route leaking process.....	347
Figure 95: Example topology.....	348
Figure 96: BGP IPv4 route leaking between VPRNs.....	349
Figure 97: BGP IPv4 route leaking from VPRN to GRT.....	359
Figure 98: BGP IPv4 route leaking from GRT to VPRN.....	364
Figure 99: BGP IPv6 route leaking between VPRNs.....	368
Figure 100: BGP IPv6 route leaking from GRT and VPRN to VPRN.....	372
Figure 101: Example topology.....	381
Figure 102: BGP route leaking process between BGP routing instances X and Y.....	391
Figure 103: Example topology.....	393
Figure 104: Leaked route 10.14.0.0/16 with next-hop resolved in VPRN 1 using IS-IS.....	396
Figure 105: Leaked route 10.24.0.0/16 with next-hop resolved in VPRN 2 using VPN-IP.....	401
Figure 106: Leaked route 10.34.0.0/16 with next-hop resolved in VPRN 2 using eBGP.....	405
Figure 107: Standard ECMP - Equal Bandwidth Links.....	411
Figure 108: Standard ECMP - Unequal Bandwidth Links.....	411
Figure 109: Link Bandwidth Extended Community Advertisement.....	412
Figure 110: Weighted ECMP - Unequal Bandwidth Links.....	413
Figure 111: Weighted ECMP - Link Aggregation Group.....	413
Figure 112: Standard ECMP - Unequal Bandwidth Links with eBGP.....	414

Figure 113: Weighted ECMP - Unequal Bandwidth Links with VPRN.....	414
Figure 114: Example Topology - BGP Weighted ECMP for IPv4 Family.....	416
Figure 115: Establishing dynamic BGP sessions.....	427
Figure 116: Dynamic BGP peers.....	428
Figure 117: Example topology with VPRN 1 in different ASs.....	435
Figure 118: Example topology.....	440
Figure 119: Advertised BGP and BGP-LU IPv4 routes.....	441
Figure 120: Advertised BGP and BGP-LU IPv6 routes.....	442
Figure 121: Example topology.....	449
Figure 122: BGP peering.....	450
Figure 123: IS-IS FAD sub-TLV.....	464
Figure 124: Application Identifier Bit Mask.....	465
Figure 125: Flexible Algorithm example in an SR-MPLS domain.....	467
Figure 126: Example topology.....	468
Figure 127: Example topology with modified IS-IS Level-1/2 capabilities.....	479
Figure 128: Link bundle schematic.....	482
Figure 129: Effect of single link failure on bundle group.....	483
Figure 130: Double link failure.....	484
Figure 131: Example topology.....	485
Figure 132: Link failure.....	491
Figure 133: Second link failure.....	494
Figure 134: Example topology.....	498
Figure 135: VPRN 1 in AS 64496.....	512

Figure 136: VPRN 2 in AS 64496 and in AS 64500.....	515
Figure 137: VPRN 3 - inter-AS VPRN model C.....	522
Figure 138: Example topology.....	535
Figure 139: Stitching RSVP/LDP tunnels to BGP tunnels.....	554
Figure 140: Example topology.....	556
Figure 141: Example topology.....	567
Figure 142: PE-1 exports BGP IPv4 and BGP-LU IPv4 routes to RR-2.....	569
Figure 143: RR-1 with separate labeled-IPv4 RIB implementation.....	581
Figure 144: Seamless MPLS - Separate labeled-IPv4 implementation.....	582
Figure 145: System architecture with separate RIBs for labeled-unicast and unlabeled routes.....	583
Figure 146: Example IPv4 topology.....	584
Figure 147: BGP sessions.....	585
Figure 148: PE-1 applies next-hop-self toward neighbor PE-2.....	589
Figure 149: Applying next-hop-self to unlabeled IP-4 routes to neighbor PE-2.....	592
Figure 150: PE-1 advertises prefixes 1.1.1.1/32 and 11.11.11.11/32.....	593
Figure 151: RR with labeled and unlabeled BGP sessions.....	597
Figure 152: Updates from unlabeled sessions not propagated to labeled sessions (default).....	598
Figure 153: RIB leaking from IPv4 BGP RIB to labeled-IPv4 BGP RIB.....	600

Preface

About This Guide

Each Advanced Configuration Guide is organized alphabetically and provides feature and configuration explanations, CLI descriptions, and overall solutions. The Advanced Configuration Guide chapters are written for and based on several Releases, up to 24.10.R2. The Applicability section in each chapter specifies on which release the configuration is based.

The Advanced Configuration Guides supplement 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.

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

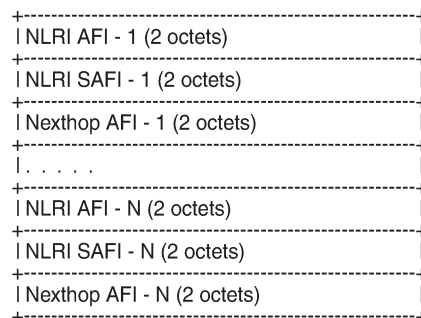
The information and configuration in this chapter are based on SR OS Release 24.3.R1. Advertising IPv4 Network Layer Reachability Information (NLRI) with IPv6 next-hop is supported in SR OS Release 19.5.R1 and later.

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 the format shown in [Figure 1: Capability value field format](#):

Figure 1: 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:

```
*A:PE-1# configure router bgp advertise-ipv6-next-hops ?
- advertise-ipv6-next-hops [vpn-ipv6] [label-ipv6] [evpn] [vpn-ipv4] [label-ipv4] [ipv4]
- no advertise-ipv6-next-hops

<vpn-ipv6>          : keyword - provision support of the specific family
<label-ipv6>       : keyword - provision support of the specific family
<evpn>             : keyword - provision support of the specific family
<vpn-ipv4>         : keyword - provision support of the specific family
<label-ipv4>       : keyword - provision support of the specific family
<ipv4>             : keyword - provision support of the specific family
```

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:

```
*A:PE-1# configure router bgp extended-nh-encoding ?
- extended-nh-encoding [label-ipv4] [vpn-ipv4] [ipv4]
- no extended-nh-encoding

<label-ipv4>       : keyword - provision support of the specific family
<vpn-ipv4>         : keyword - provision support of the specific family
<ipv4>             : keyword - provision support of the specific family
```

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:PE-1# show router bgp neighbor 2001:db8::12:2 | match "Capability" post-lines 1
Local Capability      : RtRefresh MPBGP 4byte ASN ExtNhEncoding
Remote Capability     : RtRefresh MPBGP 4byte ASN ExtNhEncoding
---snip---
```

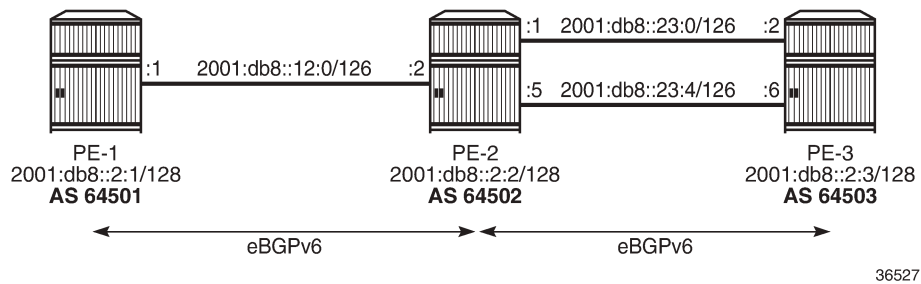
When **next-hop-self** applies to the BGP session and the neighbor address is IPv6, an IPv4-unicast route that is advertised or re-advertised gets the following as next-hop:

- 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 2: Example topology with IPv6 interfaces](#) shows the example topology with three nodes with IPv6-only interfaces in different Autonomous Systems (ASs).

Figure 2: Example topology with IPv6 interfaces

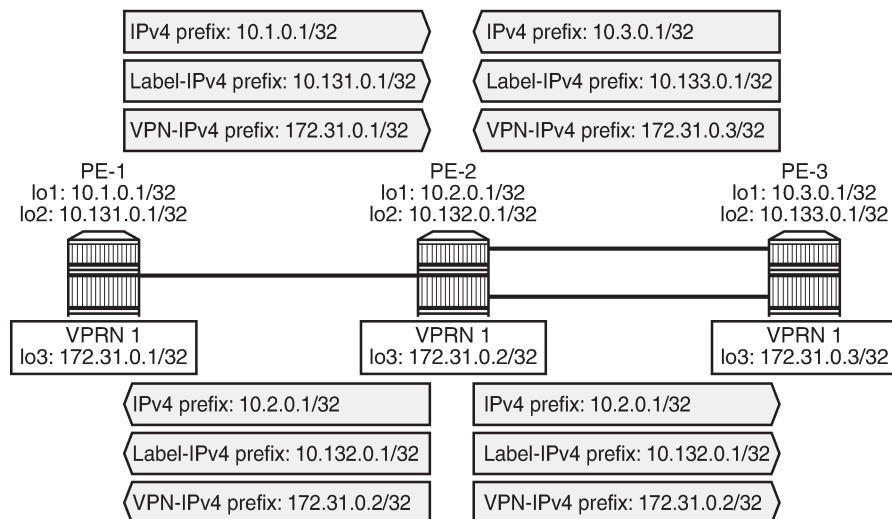


The initial configuration includes:

- Cards, MDAs, ports
- Router interfaces with IPv6 addresses

In the example, IPv4, label-IPv4, and VPN-IPv4 routes are 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 is 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 is advertised as a VPN-IPv4 route on PE-1. PE-2 and PE-3 have similar loopback interfaces. [Figure 3: Loopback addresses and advertised IPv4, label-IPv4, and VPN-IPv4 routes](#) shows the loopback addresses and the advertised routes.

Figure 3: Loopback addresses and advertised IPv4, label-IPv4, and VPN-IPv4 routes



36528

On PE-2, the interface configuration is as follows:

```
# on PE-2:
configure
router "Base"
interface "int-PE-2-PE-1"
port 1/1/c2/1
ipv6
address 2001:db8::12:2/126
```



```

        exit
        no shutdown
    exit
    interface "int-PE-2-PE-3-0"
        port 1/1/c1/1
        ipv6
            address 2001:db8::23:1/126
        exit
        no shutdown
    exit
    interface "int-PE-2-PE-3-4"
        port 1/1/c3/1
        ipv6
            address 2001:db8::23:5/126
        exit
        no shutdown
    exit
    interface "lo1"
        address 10.2.0.1/32
        loopback
        ipv6
            address 2001:db8::10:2:0:1/128
        exit
        no shutdown
    exit
    interface "lo2"
        address 10.132.0.1/32
        loopback
        ipv6
            address 2001:db8::10:132:0:1/128
        exit
        no shutdown
    exit
    interface "system"
        address 192.0.2.2/32
        ipv6
            address 2001:db8::2:2/128
        exit
        no shutdown
    exit
exit
exit

```

The interface configuration on PE-1 and on PE-3 is similar.

On PE-2, the VPRN configuration is as follows:

```

# on PE-2:
configure
service
    vprn 1 name "VPRN 1" customer 1 create
        ecmp 2
        interface "lo3" create
            address 172.31.0.2/32
            ipv6
                address 2001:db8::172:31:0:2/128
            exit
            loopback
        exit
    bgp-ipvpn
        mpls
            auto-bind-tunnel
            resolution-filter
        exit

```

```

        resolution filter
        exit
        route-distinguisher 64502:1
        vrf-target target:1:1
        no shutdown
    exit
    exit
    no shutdown
exit
exit
exit
exit

```

The VPRN configuration on PE-1 and on PE-3 is similar.

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"
    autonomous-system 64502
    ecmp 2
    bgp
      multi-path
        maximum-paths 2 ebgp 2
      exit
      enable-inter-as-vpn      # inter-AS VPN model B
      group "eBGP-IPv6"
        family ipv4 ipv6 vpn-ipv4 vpn-ipv6 label-ipv4 label-ipv6
        loop-detect discard-route
        next-hop-self
        import "import-1:1-3:3"
        export "export-10.2" "export-10.132"
        local-as 64502
        split-horizon
        extended-nh-encoding ipv4 vpn-ipv4 label-ipv4
        advertise-ipv6-next-hops ipv4 vpn-ipv4 vpn-ipv6 label-ipv4 label-ipv6
        neighbor 2001:db8::12:1
          peer-as 64501
        exit
        neighbor 2001:db8::23:2
          peer-as 64503
        exit
        neighbor 2001:db8::23:6
          peer-as 64503
        exit
        no shutdown
      exit
    exit
    no shutdown
  exit
exit
exit
exit

```

where configurations that apply for all neighbors are moved to the BGP group to which they belong. The BGP configuration on PE-1 and on PE-3 is similar.

Import and export policies tailor the information that the BGP neighbors exchange. On PE-2, the policies are configured as follows:

```

# on PE-2:
configure

```

```
router "Base"
  policy-options
  begin
    prefix-list "10.2.0.0/16"
      prefix 10.2.0.0/16 longer
    exit
    prefix-list "10.132.0.0/16"
      prefix 10.132.0.0/16 longer
    exit
    prefix-list "2001:db8::10:2:0:0/96"
      prefix 2001:db8::10:2:0:0/96 longer
    exit
    prefix-list "2001:db8::10:132:0:0/96"
      prefix 2001:db8::10:132:0:0/96 longer
    exit
    community "1:1"
      members "1:1"
    exit
    community "2:2"
      members "2:2"
    exit
    community "3:3"
      members "3:3"
    exit
    policy-statement "export-10.2"
      entry 10
        from
          prefix-list "10.2.0.0/16"
        exit
        to
          protocol bgp
        exit
        action accept
          community add "2:2"
        exit
      exit
      entry 20
        from
          prefix-list "2001:db8::10:2:0:0/96"
        exit
        to
          protocol bgp
        exit
        action accept
          community add "2:2"
        exit
      exit
    exit
    policy-statement "export-10.132"
      entry 10
        from
          prefix-list "10.132.0.0/16"
        exit
        to
          protocol bgp-label
        exit
        action accept
          community add "2:2"
        exit
      exit
      entry 20
        from
          prefix-list "2001:db8::10:132:0:0/96"
        exit
```

```

        to
          protocol bgp-label
        exit
        action accept
          community add "2:2"
        exit
      exit
    exit
  policy-statement "import-1:1-3:3"
    entry 10
      from
        community "1:1"
      exit
      action accept
    exit
  entry 20
    from
      community "3:3"
    exit
    action accept
  exit
exit
commit
exit
exit
exit

```

The policies on PE-1 and on PE-3 are 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: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. Inst          64502    33   0 00h05m25s 2/2/1 (IPv4)
                   27   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:PE-1# show router bgp routes
=====

```

```

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 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:PE-2# show router bgp routes vpn-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 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                                     524286
u*>i 64503:1:172.31.0.3/32                    None      None
      2001:db8::23:2                          None      0
      64503                                     524283
u*>i 64503:1:172.31.0.3/32                    None      None
      2001:db8::23:6                          None      0
      64503                                     524283
-----
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:PE-3# show router bgp routes label-ipv4
=====

```

```

BGP Router ID:192.0.2.3      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 LABEL-IPV4 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                               524286
u*>i  10.131.0.1/32                          None       None
      2001:db8::23:5                          None       0
      64502 64501                               524286
u*>i  10.132.0.1/32                          None       None
      2001:db8::23:1                          None       0
      64502                                     524287
u*>i  10.132.0.1/32                          None       None
      2001:db8::23:5                          None       0
      64502                                     524287
-----
Routes : 4
=====

```

The route table on PE-3 includes BGP IPv4 and label-IPv4 routes with IPv6 next-hops, as follows:

```

*A: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     00h05m02s    170
      2001:db8::23:1                               0
10.1.0.1/32                                       Remote BGP     00h05m02s    170
      2001:db8::23:5                               0
10.2.0.1/32                                       Remote BGP     00h05m02s    170
      2001:db8::23:1                               0
10.2.0.1/32                                       Remote BGP     00h05m02s    170
      2001:db8::23:5                               0
10.3.0.1/32                                       Local  Local    00h07m49s    0
      lo1                                           0
10.131.0.1/32                                     Remote BGP_LABEL 00h05m02s    170
      2001:db8::23:1                               0
10.131.0.1/32                                     Remote BGP_LABEL 00h05m02s    170
      2001:db8::23:5                               0
10.132.0.1/32                                     Remote BGP_LABEL 00h05m02s    170
      2001:db8::23:1                               0
10.132.0.1/32                                     Remote BGP_LABEL 00h05m02s    170
      2001:db8::23:5                               0
10.133.0.1/32                                       Local  Local    00h07m49s    0
      lo2                                           0
192.0.2.3/32                                       Local  Local    00h07m49s    0
      system                                         0
-----

```

```
No. of Routes: 11
---snip---
=====
```

The tunnel table on PE-3 shows four BGP tunnels with IPv6 next-hops, as follows:

```
*A: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
-----
---snip---
=====
```

The route table for VPRN 1 on PE-3 includes BGP VPN-IPv4 routes with IPv6 next-hops, as follows:

```
*A: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 00h04m45s 170
  2001:db8::23:1          0
172.31.0.1/32          Remote BGP VPN 00h04m45s 170
  2001:db8::23:5          0
172.31.0.2/32          Remote BGP VPN 00h04m45s 170
  2001:db8::23:1          0
172.31.0.2/32          Remote BGP VPN 00h04m45s 170
  2001:db8::23:5          0
172.31.0.3/32          Local  Local  00h04m49s 0
  lo3                      0
-----
No. of Routes: 5
---snip---
=====
```

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:PE-3# traceroute router 1 no-dns 172.31.0.1 source 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 3.70 ms 3.99 ms 3.23 ms
```

However, the following traceroute from lo1 on PE-3 to lo1 on PE-1 fails:

```
*A:PE-3# traceroute no-dns 10.1.0.1 source 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 * * *
6 0.0.0.0 * * *
7 0.0.0.0 * * *
8 0.0.0.0 * * *
9 0.0.0.0 * * *
10 0.0.0.0 * * *
11 0.0.0.0 * * *
12 0.0.0.0 * ^C

```

traceroute aborted by user

In an IPv6-only network, the IPv4 interfaces are down, as follows:

```
*A:PE-2# show router interface
```

```
=====
Interface Table (Router: Base)
=====
```

Interface-Name IP-Address	Adm	Opr(v4/v6)	Mode	Port/SapId PfxState
int-PE-2-PE-1 2001:db8::12:2/126 fe80::60e:1ff:fe01:b/64	Up	Down/Up	Network	1/1/c2/1 PREFERRED PREFERRED
int-PE-2-PE-3-0 2001:db8::23:1/126 fe80::60e:1ff:fe01:1/64	Up	Down/Up	Network	1/1/c1/1 PREFERRED PREFERRED
int-PE-2-PE-3-4 2001:db8::23:5/126 fe80::60e:1ff:fe01:15/64	Up	Down/Up	Network	1/1/c3/1 PREFERRED PREFERRED
lo1 10.2.0.1/32 2001:db8::10:2:0:1/128 fe80::202:feff:fe00:0/64	Up	Up/Up	Network	loopback n/a PREFERRED PREFERRED
lo2 10.132.0.1/32 2001:db8::10:132:0:1/128 fe80::202:feff:fe00:0/64	Up	Up/Up	Network	loopback n/a PREFERRED PREFERRED
system 192.0.2.2/32 2001:db8::2:2/128	Up	Up/Up	Network	system n/a 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"
      ipv6
        forward-ipv4-packets
      exit
    exit
  interface "int-PE-2-PE-3-0"
    ipv6
      forward-ipv4-packets
    exit

```



```

    exit
    interface "int-PE-2-PE-3-4"
      ipv6
      forward-ipv4-packets
    exit
  exit
exit
exit
exit

```

The configuration on PE-1 and on PE-3 is similar.

The connectivity between the lo1 and lo2 interfaces can now be verified from PE-3, as follows:

```

*A:PE-3# traceroute no-dns 10.1.0.1 source 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    2.58 ms  2.89 ms  2.93 ms
 2 10.1.0.1    3.94 ms  4.03 ms  4.03 ms

```

```

*A:PE-3# traceroute no-dns 10.2.0.1 source 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    3.44 ms  3.08 ms  2.88 ms

```

```

*A:PE-3# traceroute no-dns 10.131.0.1 source 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    2.52 ms  3.03 ms  2.54 ms
 2 10.131.0.1  4.24 ms  3.59 ms  3.80 ms

```

```

*A:PE-3# traceroute no-dns 10.132.0.1 source 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  2.77 ms  2.98 ms  2.63 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.

PE-1 does not accept IPv4, VPN-IPv4 and label-IPv4 BGP routes that have an IPv6 next-hop, when **extended-nh-encoding** is not configured on PE-1 for the BGP neighbor on PE-2, as follows:

```

*A:PE-1# show router bgp neighbor 2001:db8::12:2 | match "Capability" post-lines 1
Local Capability      : RtRefresh MPBGP 4byte ASN
Remote Capability    : RtRefresh MPBGP 4byte ASN ExtNhEncoding
---snip---

*A:PE-2# show router bgp neighbor 2001:db8::12:1 | match "Capability" post-lines 1
Local Capability      : RtRefresh MPBGP 4byte ASN ExtNhEncoding
Remote Capability    : RtRefresh MPBGP 4byte ASN
---snip---

```

This is verified as follows:

```

*A: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. Inst      64502      27    0 00h03m20s 2/0/1 (IPv4)
                21    0          2/2/1 (IPv6)
                2/0/1 (VpnIPv4)
                2/2/1 (VpnIPv6)
                2/0/1 (Lbl-IPv4)
                2/2/1 (Lbl-IPv6)
-----
```

PE-1 does not install the corresponding routes in its route tables.

- When **extended-nh-encoding** is configured on PE-1:
 - the PE-1 route table has BGP routes to lo1 on PE-2, lo1 on PE-3, lo2 on PE-2, and lo2 on PE-3, via an IPv6 next-hop as follows:

```
*A:PE-1# show router route-table ipv4
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
  Next Hop[Interface Name]         Metric
-----
10.1.0.1/32                        Local  Local  00h08m10s  0
   lo1                               0
10.2.0.1/32                        Remote BGP    00h05m13s 170
   2001:db8::12:2                    0
10.3.0.1/32                        Remote BGP    00h04m45s 170
   2001:db8::12:2                    0
10.131.0.1/32                      Local  Local  00h08m10s  0
   lo2                               0
10.132.0.1/32                      Remote BGP_LABEL 00h05m13s 170
   2001:db8::12:2                    0
10.133.0.1/32                      Remote BGP_LABEL 00h04m45s 170
   2001:db8::12:2                    0
192.0.2.1/32                       Local  Local  00h08m10s  0
   system                             0
-----
No. of Routes: 7
---snip---
```

- the VPRN 1 route table on PE-1 has BGP routes to lo3 on PE-2 and lo3 on PE-3, via an IPv6 next-hop as follows:

```
*A:PE-1# show router 1 route-table ipv4
=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
  Next Hop[Interface Name]         Metric
-----
172.31.0.1/32                      Local  Local  00h05m08s  0
   lo3                               0
172.31.0.2/32                      Remote BGP VPN 00h04m45s 170
```

```

    2001:db8::12:2                                0
172.31.0.3/32                                     Remote BGP VPN 00h04m12s 170
    2001:db8::12:2                                0
-----
No. of Routes: 3
---snip---
=====

```

- the tunnel table on PE-1 has tunnels to lo2 on PE-2 and lo2 on PE-3, as follows:

```

*A:PE-1# show router tunnel-table

=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner      Encap TunnelId  Pref  Nexthop          Metric
  Color
-----
10.132.0.1/32        bgp        MPLS  262145    12    2001:db8::12:2 1000
10.133.0.1/32        bgp        MPLS  262146    12    2001:db8::12:2 1000
-----
---snip---
=====

```

- When **extended-nh-encoding** is not configured on PE-1:
 - those BGP routes and tunnels are missing, as follows:

```

*A:PE-1# show router route-table ipv4

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]          Type  Proto  Age      Pref
  Next Hop[Interface Name]  Metric
-----
10.1.0.1/32                 Local Local  00h20m09s  0
    lo1                      0
10.131.0.1/32               Local Local  00h20m09s  0
    lo2                      0
192.0.2.1/32                Local Local  00h20m09s  0
    system                    0
-----
No. of Routes: 3
---snip---
=====

```

```

*A:PE-1# show router 1 route-table ipv4

=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]          Type  Proto  Age      Pref
  Next Hop[Interface Name]  Metric
-----
172.31.0.1/32               Local Local  00h17m07s  0
    lo3                      0
-----
No. of Routes: 1
---snip---

```

```

=====
*A:PE-1# show router tunnel-table
=====
IPv4 Tunnel Table (Router: Base)
=====
Destination          Owner    Encap TunnelId  Pref  Nexthop      Metric
  Color
-----
No Matching Entries.
---snip---
=====

```

- and there is no connectivity from and to PE-1. This is verified as follows:

```

# on PE-1, towards PE-2 and PE-3

*A:PE-1# ping 10.2.0.1 source 10.1.0.1 rapid
PING 10.2.0.1 56 data bytes
.....
---- 10.2.0.1 PING Statistics ----
5 packets transmitted, 0 packets received, 100% packet loss
*A:PE-1# ping 10.3.0.1 source 10.1.0.1 rapid
PING 10.3.0.1 56 data bytes
.....
---- 10.3.0.1 PING Statistics ----
5 packets transmitted, 0 packets received, 100% packet loss
*A:PE-1# ping 10.132.0.1 source 10.131.0.1 rapid
PING 10.132.0.1 56 data bytes
.....
---- 10.132.0.1 PING Statistics ----
5 packets transmitted, 0 packets received, 100% packet loss
*A:PE-1# ping 10.133.0.1 source 10.131.0.1 rapid
PING 10.133.0.1 56 data bytes
.....
---- 10.133.0.1 PING Statistics ----
5 packets transmitted, 0 packets received, 100% packet loss
*A:PE-1# ping router 1 172.31.0.2 source 172.31.0.1 rapid
PING 172.31.0.2 56 data bytes
.....
---- 172.31.0.2 PING Statistics ----
5 packets transmitted, 0 packets received, 100% packet loss
*A:PE-1# ping router 1 172.31.0.3 source 172.31.0.1 rapid
PING 172.31.0.3 56 data bytes
.....
---- 172.31.0.3 PING Statistics ----
5 packets transmitted, 0 packets received, 100% packet loss

```

```

# on PE-3, towards PE-1 (similar on PE-2 towards PE-1):

*A:PE-3# ping 10.1.0.1 source 10.3.0.1 rapid
PING 10.1.0.1 56 data bytes
.....
---- 10.1.0.1 PING Statistics ----
5 packets transmitted, 0 packets received, 100% packet loss
---snip---
*A:PE-3# ping 10.131.0.1 source 10.133.0.1 rapid
PING 10.131.0.1 56 data bytes
.....
---- 10.131.0.1 PING Statistics ----

```

```
5 packets transmitted, 0 packets received, 100% packet loss
---snip---
*A:PE-3# ping router 1 172.31.0.1 source 172.31.0.3 rapid
PING 172.31.0.1 56 data bytes
.....
---- 172.31.0.1 PING Statistics ----
5 packets transmitted, 0 packets received, 100% packet loss
---snip---
```

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 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 additional information about that route prefix. A number of 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.

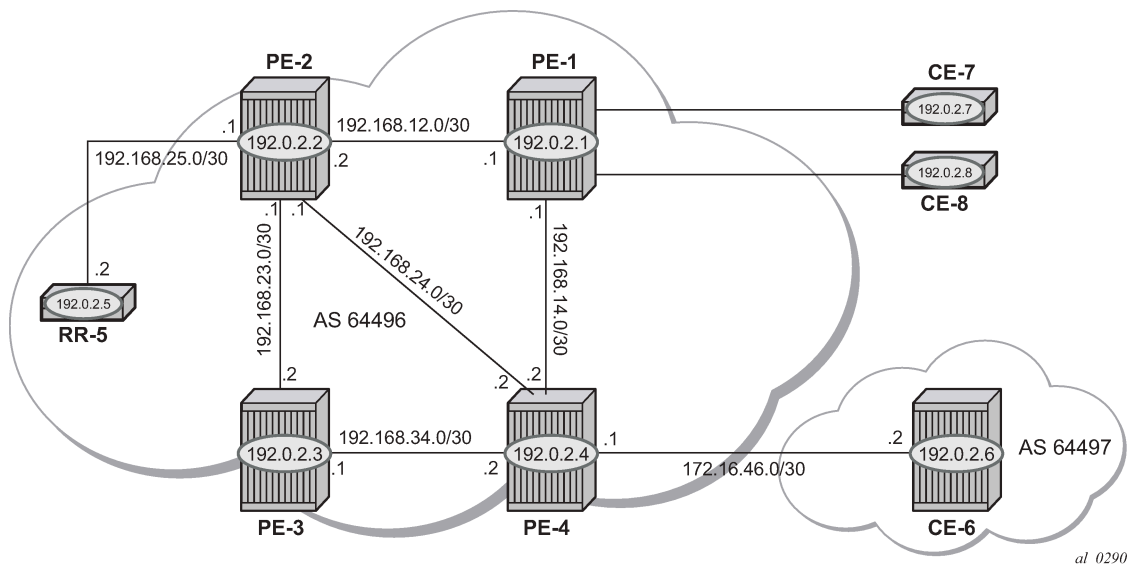
A BGP community is a 32-bit value that is written as two 16-bit numbers separated by a colon. The first number usually represents the autonomous system (AS) number that defines or originates the community while the second is set by the network administrator.

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 4: 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 4: 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 each PE.

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
  autonomous-system 64496
  bgp
    group "internal"
      family vpn-ipv4
      peer-as 64496
      neighbor 192.0.2.5
    exit
  exit
```

The IP addresses can be derived from [Figure 4: Example topology](#).

The BGP configuration for RR-5 is as follows:

```
# on RR-5:
configure
router
  autonomous-system 64496
  bgp
    cluster 0.0.0.1
    group "RR-clients"
      family vpn-ipv4
      peer-as 64496
      neighbor 192.0.2.1
    exit
    neighbor 192.0.2.2
    exit
    neighbor 192.0.2.3
    exit
    neighbor 192.0.2.4
    exit
  exit
```

The following BGP summary on RR-5 shows that BGP sessions with each PE are established for the VPN-IPv4 address family:

```
*A: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)
```



```

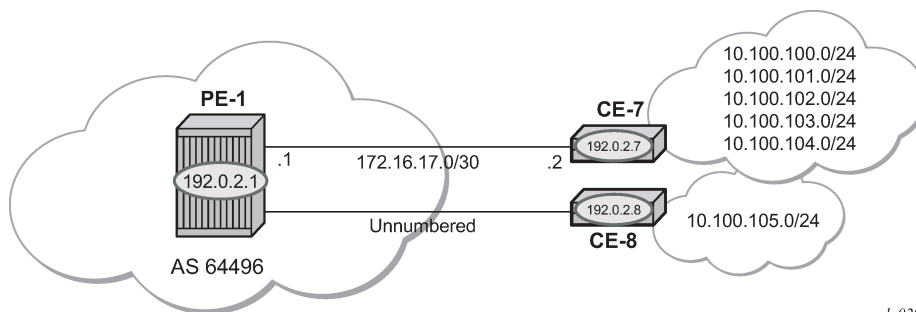
3 0
192.0.2.2
Def. Instance 64496 3 0 00h00m11s 0/0/0 (VpnIPv4)
3 0
192.0.2.3
Def. Instance 64496 3 0 00h00m11s 0/0/0 (VpnIPv4)
3 0
192.0.2.4
Def. Instance 64496 3 0 00h00m11s 0/0/0 (VpnIPv4)
3 0
-----

```

VPRN: IPv4

Figure 5: CE connections for next-hops shows the Customer Edge (CE) routers connected to PE-1.

Figure 5: CE connections for next-hops



al_0291

The VPRN configuration for PE-1 is as follows:

```

# on PE-1:
configure
service
  vprn 1 name "VPRN 1" customer 1 create
  route-distinguisher 64496:1
  auto-bind-tunnel
  resolution-filter
  ldp
  exit
  resolution filter
  exit
  vrf-target target:64496:1
  interface "int-PE-1-CE-7" create
  address 172.16.17.1/30
  sap 1/2/1:1.0 create
  exit
  interface "loop1" create
  address 192.0.2.100/32
  loopback
  exit
  interface "int-PE-1-CE-8" create
  unnumbered "loop1"
  sap 1/2/2:1.0 create
  exit
exit

```

```
no shutdown
```

For unnumbered interfaces, an IP address is borrowed from a loopback interface, see chapter "Unnumbered Interfaces in RSVP-TE and LDP" in *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Advanced Configuration Guide for Classic CLI*.

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
router
  policy-options
  begin
  policy-statement "BGP-VPN-accept"
  entry 10
  from
  protocol bgp-vpn
  exit
  action accept
  exit
  exit
  exit
  commit
  exit
```

The configuration of the VPRN service on PE-4 is as follows:

```
# on PE-4:
configure
service
  vprn 1 name "VPRN 1" customer 1 create
  autonomous-system 64496
  route-distinguisher 64496:1
  auto-bind-tunnel
  resolution-filter
  ldp
  exit
  resolution filter
  exit
  vrf-target target:64496:1
  interface "int-PE-4-CE-6" create
  address 172.16.46.1/30
  sap 1/2/1:1 create
  exit
  exit
  bgp
  group "VPRN1-external"
  export "BGP-VPN-accept"
  peer-as 64497
  neighbor 172.16.46.2
  exit
  exit
  exit
  no shutdown
```

Static routes with communities

A static route has a number of next-hop options: direct connected IP address, black-hole, indirect IP address, and interface-name.

[Figure 5: 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 a number of 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 (0xFFFFFFFF01), but the CLI requires the keyword **no-export**.

```
# on PE-1:
configure
  service
    vprn 1
      static-route-entry 10.100.100.0/24
        next-hop 172.16.17.2
          community no-export
          no shutdown
        exit
```

- 10.100.101.0/24 must be advertised with a community of 64496:101

```
static-route-entry 10.100.101.0/24
  next-hop 172.16.17.2
    community 64496:101
    no shutdown
  exit
```

- 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 (0xFFFFFFFF02), but the CLI requires the keyword **no-advertise**.

```
static-route-entry 10.100.102.0/24
  next-hop 172.16.17.2
    community no-advertise
    no shutdown
  exit
```

- 10.100.103.0/24 must be advertised with a community of 64496:103 and a route tag of 10.

```
static-route-entry 10.100.103.0/24
  next-hop 172.16.17.2
  community 64496:103
  tag 10
  no shutdown
exit
exit
```

- 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.

```
static-route-entry 10.100.104.0/24
  indirect 192.0.2.7
  community 64496:104
  no shutdown
exit
exit
static-route-entry 192.0.2.7/32
  next-hop 172.16.17.2
  community no-advertise
  no shutdown
exit
exit
```

- 10.100.105.0/24 must be advertised with a community of 64496:105. It is reachable via the unnumbered interface to CE-8.

```
static-route-entry 10.100.105.0/24
  next-hop "int-PE-1-CE-8"
  community 64496:105
  no shutdown
exit
exit
```

On PE-1, static routes are configured that match the static routes from [Figure 5: 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: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 Nextthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
i	64496:1:10.100.100.0/24 192.0.2.1 No As-Path	100 None	None n/a 524283
i	64496:1:10.100.101.0/24 192.0.2.1 No As-Path	100 None	None n/a 524283
i	64496:1:10.100.103.0/24 192.0.2.1 No As-Path	100 None	None n/a 524283
i	64496:1:10.100.104.0/24 192.0.2.1 No As-Path	100 None	None n/a 524283
i	64496:1:10.100.105.0/24 192.0.2.1 No As-Path	100 None	None n/a 524283
i	64496:1:172.16.17.0/30 192.0.2.1 No As-Path	100 None	None n/a 524283
i	64496:1:192.0.2.100/32 192.0.2.1 No As-Path	100 None	None n/a 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: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
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      : 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: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: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: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: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
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
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   : 64496:103 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: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
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
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   : 64496:104 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: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
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     : 64496:105 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:PE-4# show router 1 route-table protocol bgp-vpn
=====
Route Table (Service: 1)
=====
```

```

Dest Prefix[Flags]
  Next Hop[Interface Name]
-----
10.100.100.0/24
  192.0.2.1 (tunneled)      Remote BGP VPN 01h54m30s 170
                             0
10.100.101.0/24
  192.0.2.1 (tunneled)      Remote BGP VPN 01h46m55s 170
                             0
10.100.103.0/24
  192.0.2.1 (tunneled)      Remote BGP VPN 01h37m47s 170
                             0
10.100.104.0/24
  192.0.2.1 (tunneled)      Remote BGP VPN 01h30m18s 170
                             0
10.100.105.0/24
  192.0.2.1 (tunneled)      Remote BGP VPN 01h26m58s 170
                             0
172.16.17.0/30
  192.0.2.1 (tunneled)      Remote BGP VPN 01h54m30s 170
                             0
192.0.2.100/32
  192.0.2.1 (tunneled)      Remote BGP VPN 01h54m30s 170
                             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:CE-6# show router route-table protocol bgp

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]
  Next Hop[Interface Name]
-----
10.100.101.0/24
  172.16.46.1      Remote BGP 00h04m31s 170
                   0
10.100.103.0/24
  172.16.46.1      Remote BGP 00h04m31s 170
                   0
10.100.104.0/24
  172.16.46.1      Remote BGP 00h04m31s 170
                   0
10.100.105.0/24
  172.16.46.1      Remote BGP 00h04m31s 170
                   0
172.16.17.0/30
  172.16.46.1      Remote BGP 00h04m31s 170
                   0
192.0.2.100/32
  172.16.46.1      Remote BGP 00h04m31s 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 prefix 10.100.100.0/24 is not received from PE-4 because it is a member of the **no-export** community.

```

*A: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 with the correct community 64496:101.

```
*A: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 Nexthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
u*>i	10.100.101.0/24 172.16.46.1 64496	None None	None 0 -

```
-----
Routes : 1
=====
```

Static route 10.100.103.0/24 is received with the correct community 64496:103, as follows:

```
*A: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 Nexthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
u*>i	10.100.103.0/24 172.16.46.1 64496	None None	None 0 -

```
-----
Routes : 1
=====
```

Static route 10.100.104.0/24 is received with the correct community 64496:104, as follows:

```
*A: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 with the correct community 64496:105.

```
*A: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.

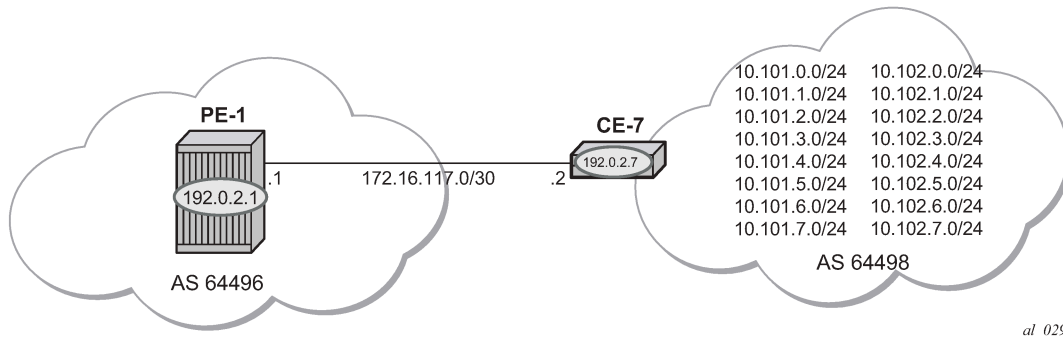
This is due to the fact that the third octet in the range 0 to 7 can be represented by the 8 bits 00000000 to 00000111. The first 5 bits of this octet are common, along with the previous 2 octets, giving a prefix where the first 21 bits are common. Therefore, the aggregate can be written as 10.101.0.0/21.

In order to illustrate the configuration of an aggregate, consider following.

Figure 6: 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 6: CE-7 connectivity



al_0292

Instead of advertising all these prefixes out of the VPRN towards 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 a VPRN on PE-1, including the external BGP configuration is as follows:

```
# on PE-1:
configure
  service
    vprn 2 name "VPRN 2" customer 1 create
      autonomous-system 64496
      route-distinguisher 64496:2
      auto-bind-tunnel
      resolution-filter
        ldp
      exit
      resolution filter
    exit
  vrf-target target:64496:2
  interface "int-PE-1-CE-7_2nd" create
    address 172.16.117.1/30
    sap 1/2/1:2.0 create
  exit
  exit
  bgp
    group "external"
      peer-as 64498
      neighbor 172.16.117.2
    exit
  exit
  no shutdown
  exit
  no shutdown
exit
```

The BGP neighbor relationship shows the following:

```
*A:PE-1# show router 2 bgp neighbor
```

```

=====
BGP Neighbor
=====
-----
Peer          : 172.16.117.2
Description   : (Not Specified)
Group         : external
-----
Peer AS       : 64498           Peer Port      : 50409
Peer Address  : 172.16.117.2
Local AS      : 64496           Local Port     : 179
Local Address : 172.16.117.1
Peer Type     : External       Dynamic Peer   : No
State        : Established     Last State     : Established
Last Event    : rcvOpen
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    : None
Preference    : 170           Num of Update Flaps : 0
Input Queue   : 0             Output Queue     : 0
Input Messages : 7           Output Messages  : 7
Input Octets  : 247           Output Octets    : 232
Input Updates : 1             Output Updates   : 1
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    : None Specified - Default Accept
Export Policy    : None Specified - Default Accept
---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: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 Nexthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
u*>i	10.101.0.0/24 172.16.117.2 64498	None None	None 0 -
u*>i	10.101.1.0/24 172.16.117.2 64498	None None	None 0 -
u*>i	10.101.2.0/24 172.16.117.2 64498	None None	None 0 -
u*>i	10.101.3.0/24 172.16.117.2 64498	None None	None 0 -
u*>i	10.101.4.0/24 172.16.117.2 64498	None None	None 0 -
u*>i	10.101.5.0/24 172.16.117.2 64498	None None	None 0 -
u*>i	10.101.6.0/24 172.16.117.2 64498	None None	None 0 -
u*>i	10.101.7.0/24 172.16.117.2 64498	None None	None 0 -
u*>i	10.102.0.0/24 172.16.117.2 64498	None None	None 0 -
u*>i	10.102.1.0/24 172.16.117.2 64498	None None	None 0 -
u*>i	10.102.2.0/24 172.16.117.2 64498	None None	None 0 -
u*>i	10.102.3.0/24 172.16.117.2 64498	None None	None 0 -
u*>i	10.102.4.0/24 172.16.117.2 64498	None None	None 0 -
u*>i	10.102.5.0/24 172.16.117.2 64498	None None	None 0 -
u*>i	10.102.6.0/24 172.16.117.2 64498	None None	None 0 -
u*>i	10.102.7.0/24 172.16.117.2 64498	None None	None 0 -

Routes : 16

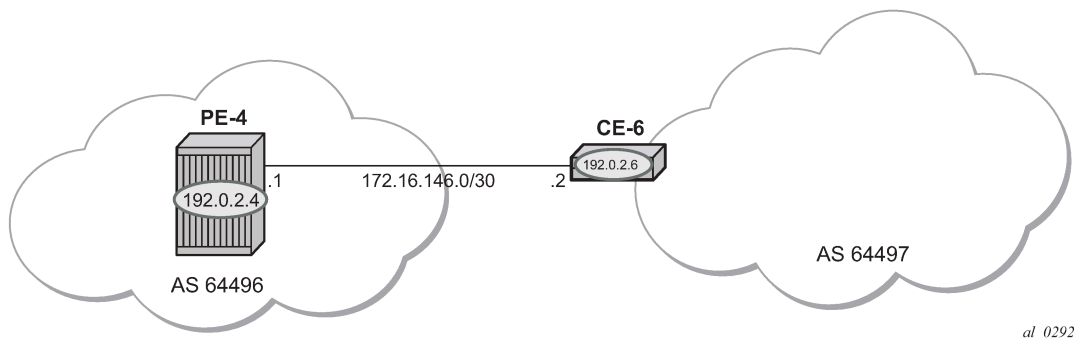
=====

PE-4 also has a VPRN 2 instance configured, so that it will receive the imported BGP routes. The service configuration for PE-4 is as follows:

```
# on PE-4:
configure
service
  vprn 2 name "VPRN 2" customer 1 create
  autonomous-system 64496
  route-distinguisher 64496:2
  auto-bind-tunnel
  resolution-filter
  ldp
  exit
  resolution filter
  exit
  vrf-target target:64496:2
  interface "int-PE-4-CE-6_2nd" create
  address 172.16.146.1/30
  sap 1/2/1:2 create
  exit
  exit
  bgp
  group "VPRN2-external"
  peer-as 64497
  neighbor 172.16.146.2
  exit
  no shutdown
  exit
  no shutdown
  exit
```

Figure 7: 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 7: CE-6 connectivity



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: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 2 name "IES 2" customer 1 create
    interface "int-CE-6-PE-4_2nd" create
      address 172.16.146.2/30
      sap 1/1/1:2 create
    exit
  exit
no shutdown

```

The BGP configuration of CE-6 is as follows:

```

# on CE-6:
configure
  router

```

```

    bgp
      group "external-toVPRN2onPE-4"
        peer-as 64496
        neighbor 172.16.146.1
        exit
      exit
    no shutdown
  
```

The BGP neighbor state for PE-4 is as follows:

```
*A: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      : 49549
Peer Address  : 172.16.146.2
Local AS      : 64496           Local Port     : 179
Local Address : 172.16.146.1
Peer Type     : External        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     : 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    : None Specified - Default Accept
Export Policy    : None Specified - Default Accept
---snip---
  
```

```
-----
Neighbors shown : 1
=====
* indicates that the corresponding row element may have been truncated.
```

In order 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 2
      aggregate 10.101.0.0/21 community 64496:101
      aggregate 10.102.0.0/21 community no-export
    exit
```

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
  router
    policy-options
      begin
        policy-statement "PE-4-VPN-Agg"
          entry 10
            from
              protocol aggregate
            exit
            action accept
          exit
        exit
      exit
    commit
```

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 2
      bgp
        group "VPRN2-external"
          export "PE-4-VPN-Agg"
        exit
```

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: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
Nexthop       : 172.16.146.1
Path Id       : None
From          : 172.16.146.1
Res. Protocol : LOCAL                Res. Metric   : 0
Res. Nexthop  : 172.16.146.1
Local Pref.   : None
Aggregator AS : 64496                 Interface Name : int-CE-6-PE-4_2nd
Atomic Aggr.  : Atomic               Aggregator    : 192.0.2.4
AIGP Metric   : None                 MED           : None
Connector     : None                 IGP Cost      : 0
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: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 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:

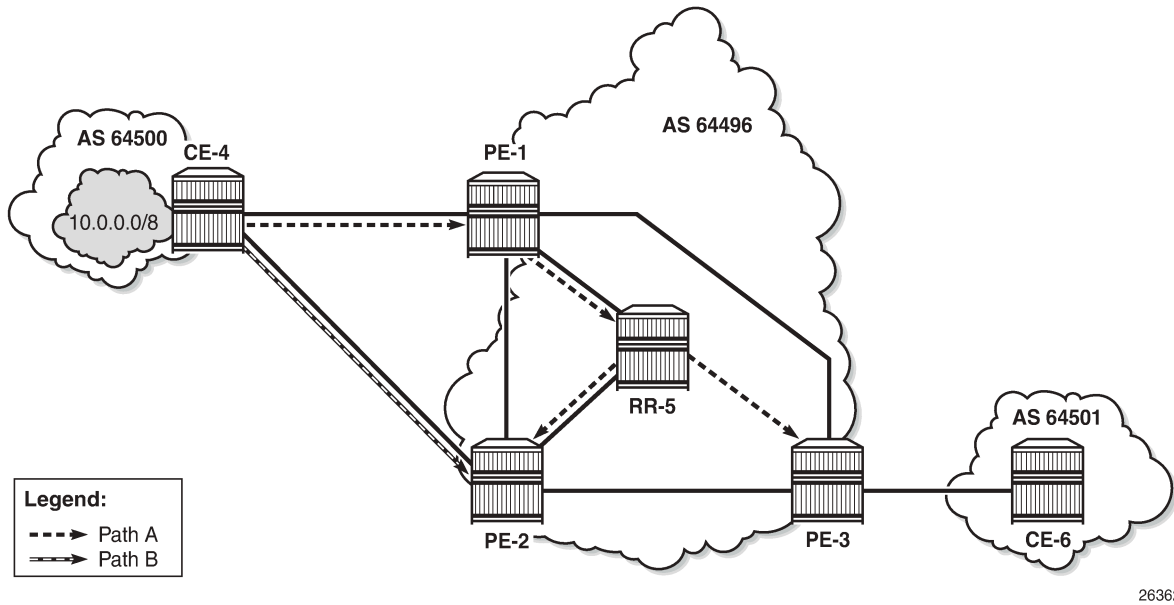
- Faster convergence after failure
- Enhanced load-sharing
- Reduced routing churn

These benefits are described in the following sections.

Faster convergence after failure

[Figure 8: 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 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 8: RR advertises best path only – path A preferred over path B

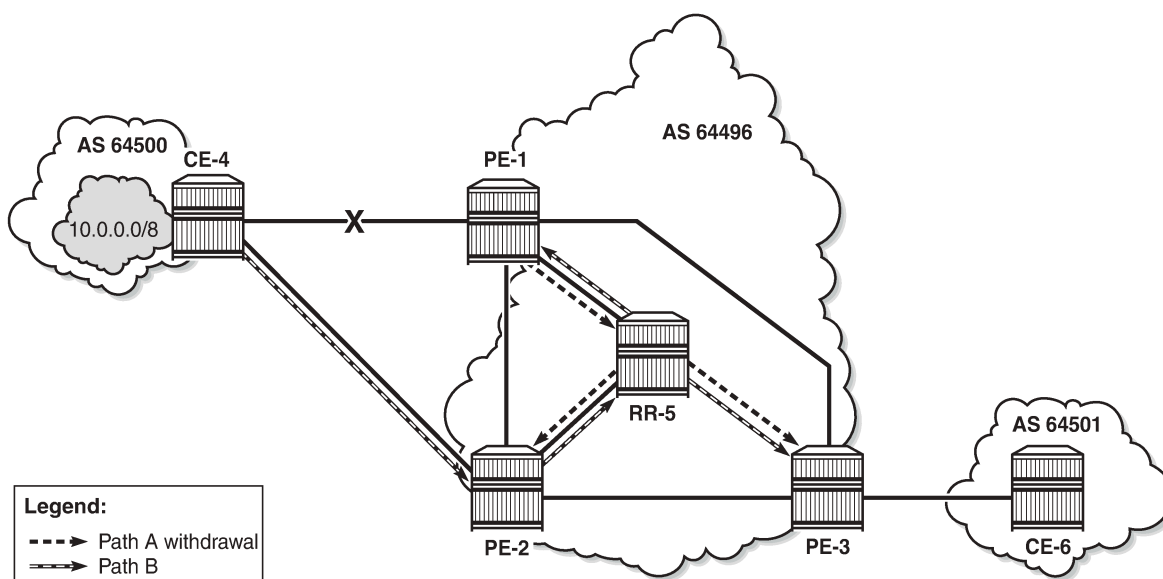


When the link between CE-4 and PE-1 fails, the following steps take place for reconvergence:

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 9: Reconvergence after path failure (without add-path) shows the BGP updates sent to withdraw path A and advertise path B.

Figure 9: Reconvergence after path failure (without add-path)



26363

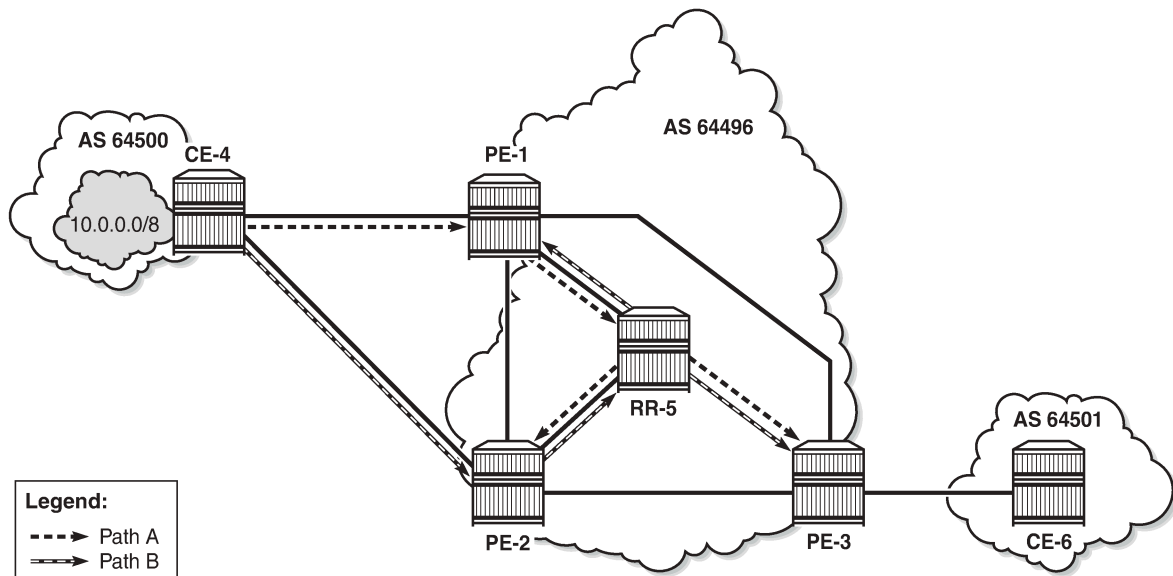
If the propagation time of a BGP update message between RR-5 and any of its clients is X , the convergence time is four times X , plus processing, transmission, and queuing delays.

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 10: 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 10: Advertised paths when BGP add-path is enabled in PEs and RR

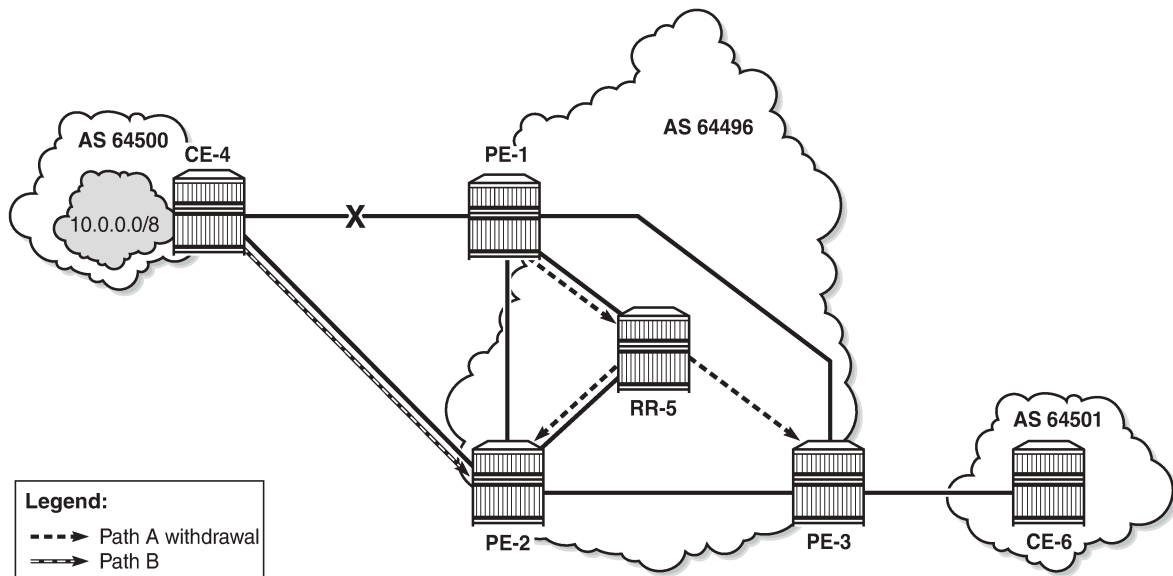


26364

Figure 11: [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 11: 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:

```
*A:PE-1>config>router>bgp# add-paths ?
- add-paths
- no add-paths

[no] evpn          - Configure evpn ADD-PATH limits
[no] ipv4          - Configure ipv4 ADD-PATH limits
[no] ipv6          - Configure ipv6 ADD-PATH limits
[no] label-ipv4    - Configure label-ipv4 ADD-PATH limits
[no] label-ipv6    - Configure label-ipv6 ADD-PATH limits
[no] mcast-vpn-ipv4 - Configure mcast-vpn-ipv4 ADD-PATH limits
[no] mcast-vpn-ipv6 - Configure mcast-vpn-ipv6 ADD-PATH limits
[no] mvpn-ipv4     - Configure mvpn-ipv4 ADD-PATH limits
[no] mvpn-ipv6     - Configure mvpn-ipv6 ADD-PATH limits
[no] vpn-ipv4      - Configure vpn-ipv4 ADD-PATH limits
[no] vpn-ipv6      - Configure vpn-ipv6 ADD-PATH limits
```

Up to 16 paths are configurable per address family per peer (send-limit):

```
*A:PE-1>config>router>bgp>add-paths# ipv4 ?
- ipv4 send <send-limit>
- ipv4 send <send-limit> receive [none]
- no ipv4

<send-limit>      : [1..16]|none|multipaths
```

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-limit> 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:
debug
  router "Base"
    bgp
      open
    exit
```

```
58 2022/05/04 08:04:37.417 UTC MINOR: DEBUG #2001 Base BGP
"BGP: OPEN
Peer 1: 192.0.2.5 - Send (Passive) BGP OPEN: Version 4
AS Num 64496: Holdtime 90: BGP_ID 192.0.2.1: Opt Length 26 (Ext0pt 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:
debug
  router "Base"
    bgp
      update
    exit
```

RR-5 received the following BGP update for prefix 10.0.4.0/24 with path ID.

```
50 2022/05/04 08:05:07.380 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 = 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 8  
"
```

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.

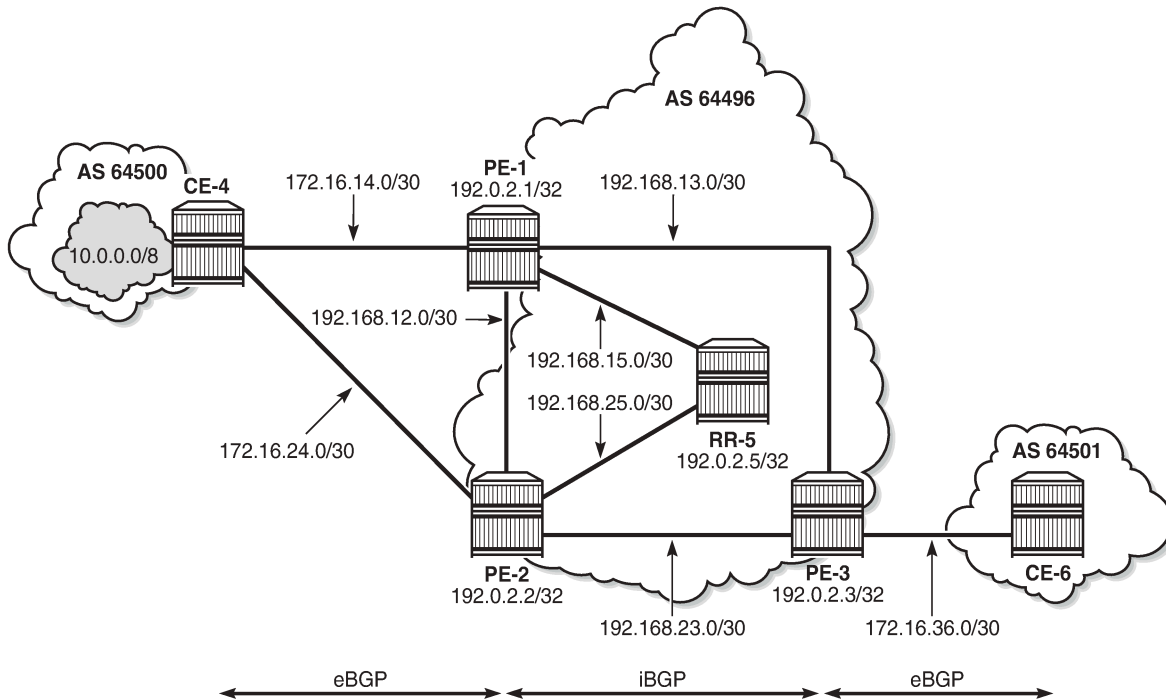
Configuration

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 12: 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 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 12: 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
router Base
  autonomous-system 64500
  policy-options
    begin
    prefix-list "10.0.4.0/24"
      prefix 10.0.4.0/24 exact
    exit
  policy-statement "export-bgp"
    entry 10
      from
        prefix-list "10.0.4.0/24"
    exit
```

```

        action accept
        exit
    exit
exit
commit
exit
bgp
    rapid-withdrawal
    split-horizon
    group "EBGP"
        export "export-bgp"
        peer-as 64496
        neighbor 172.16.14.1
        exit
        neighbor 172.16.24.1
        exit
    exit
exit

```

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
    router Base
        autonomous-system 64496
        policy-options
            begin
                policy-statement "import-bgp-LP200"
                    default-action accept
                    local-preference 200
                exit
            exit
        commit
    exit
    bgp
        rapid-withdrawal
        split-horizon
        group "EBGP"
            import "import-bgp-LP200"
            peer-as 64500
            neighbor 172.16.14.2
            exit
        exit
        group "IBGP"
            next-hop-self
            peer-as 64496
            neighbor 192.0.2.5
            exit
        exit
    exit

```

The BGP configuration on PE-2 and PE-3 is similar, but there is no import policy.

The BGP configuration on RR-5 is as follows:

```

# on RR-5:
configure
    router Base
        autonomous-system 64496
    bgp
        rapid-withdrawal
        split-horizon

```

```

group "IBGP"
  cluster 192.0.2.5
  peer-as 64496
  neighbor 192.0.2.1
  exit
  neighbor 192.0.2.2
  exit
  neighbor 192.0.2.3
  exit
exit

```

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: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
      192.0.2.1      200      None
      64500          None      10
      -
-----
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"
      shutdown

```

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:
28 2022/05/04 08:00:38.222 UTC 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:
29 2022/05/04 08:00:38.223 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 = 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:
31 2022/05/04 08:00:57.380 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 = 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:
32 2022/05/04 08:01:00.618 UTC 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: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
	192.0.2.2	None	10
	64500		-

```
-----
Routes : 1
=====
```

The configuration is restored as follows:

```
# on CE-4:
configure
router Base
interface "int-CE-4-PE-1"
no shutdown
```

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: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" and no limit on the receive path limit, which is the default setting, as follows:

```
# on PE-1 and PE-2:
configure
router Base
bgp
group "EBGP"
add-paths
ipv4 send 2 receive
exit
exit
group "IBGP"
add-paths
ipv4 send 2 receive
exit
exit
```

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: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 none** option implies that the add-path receive capability is not negotiated.

```
# on RR-5:
configure
  router Base
    bgp
      group "IBGP"
        neighbor 192.0.2.1
          add-paths
            ipv4 send 2 receive none
        exit
      exit
    exit
  group "IBGP"
    neighbor 192.0.2.2
      add-paths
        ipv4 send 2 receive
    exit
  exit
```

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: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: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: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 none** 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: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: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: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
    exit
```

```
# on RR-5:
configure
router Base
  bgp
    group "IBGP"
      neighbor 192.0.2.3
      add-paths
      ipv4 send 2 receive
    exit
```

```
*A: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              14         10
      64500                   -
*i    10.0.4.0/24                100        None
      192.0.2.2              15         10
      64500                   -
-----
Routes : 2
=====
```

BGP add-path is enabled, but BGP FRR or ECMP are disabled. The routing table on PE-3 only contains one entry for prefix 10.0.4.0/24:

```
*A: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    00h00m29s  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"
      shutdown

```

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: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
     192.0.2.2                15         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: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   00h00m10s 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
```

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: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	Nexthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
u*>i	10.0.4.0/24	192.0.2.1 64500	200 20	None 10 -
ub*i	10.0.4.0/24	192.0.2.2 64500	15 10	100 None -

```
=====
```

```
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:PE-3# show router route-table 10.0.4.0/24
```

```
=====
```

```
Route Table (Router: Base)
```

```
=====
```

Dest Prefix[Flags] Next Hop[Interface Name]	Type	Proto	Age Metric	Pref
10.0.4.0/24 [B] 192.168.13.1	Remote	BGP	00h00m49s 10	170

```
-----
```

```
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:PE-3# show router route-table 10.0.4.0/24 alternative
```

```
=====
```

```
Route Table (Router: Base)
```

```
=====
```

Dest Prefix[Flags] Next Hop[Interface Name] Alt-NextHop	Type	Proto	Age Metric Alt- Metric	Pref
10.0.4.0/24 192.168.13.1	Remote	BGP	00h00m49s 10	170
10.0.4.0/24 (Backup) 192.168.23.1	Remote	BGP	00h00m49s 10	170

```
-----
```

```
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 as follows:

```
# on all PEs:
configure
  router Base
    bgp
      no backup-path
```


Add-path and ECMP

On PE-1, the import policy is removed to have paths with equal cost:

```
# on PE-1:
configure
router Base
  bgp
    group "EBGP"
    no import
```

ECMP is enabled on all PEs with a value of two, as follows:

```
# on all PEs:
configure
router Base
  ecmp 2
```

On all PEs, BGP multipath is configured with the maximum number of paths equal to two in the **bgp** context, as follows:

```
# on all PEs:
configure
router Base
  bgp
    multi-path
    maximum-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: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              15         10
      64500                   -
-----
Routes : 2
```

```

=====
*A: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      00h00m54s  170
    192.168.13.1                      10
10.0.4.0/24                        Remote BGP      00h00m54s  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
=====

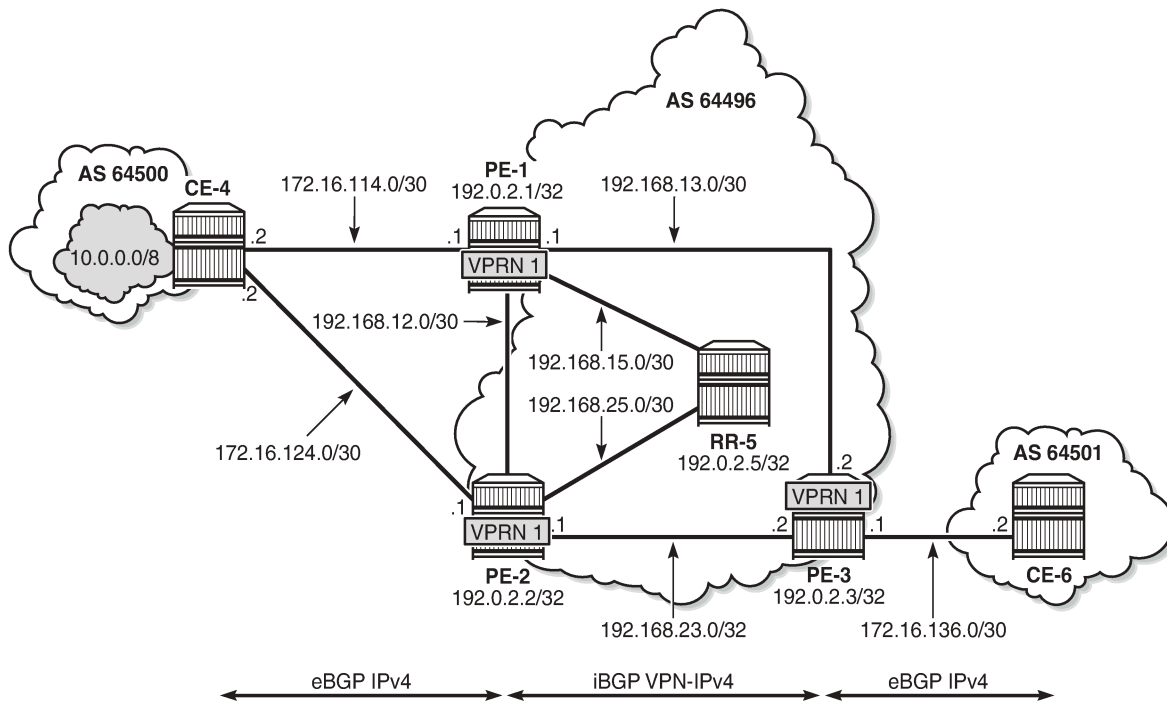
```

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 13: 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 13: 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 **enable-bgp-vpn-backup** 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
router Base
policy-options
begin
policy-statement "export-bgp"
entry 10
from
protocol bgp-vpn
exit
to
protocol bgp
exit
action accept
exit
exit
policy-statement "import-bgp-LP200"
default-action accept
local-preference 200
exit
exit
commit
exit
exit
service
```

```

vprn 1 name "VPRN 1" customer 1 create
  autonomous-system 64496
  enable-bgp-vpn-backup ipv4 # BGP FRR
  interface "int-PE-1-CE-4_VPRN1" create
    address 172.16.114.1/30
    sap 1/1/3:1 create
    exit
  exit
  bgp-ipvpn
    mpls
      auto-bind-tunnel
        resolution any
      exit
      route-distinguisher 64496:1
      vrf-target target:64496:1
      no shutdown
    exit
  exit
  bgp
    split-horizon
    group "EBGP_1"
      next-hop-self
      import "import-bgp-LP200"
      export "export-bgp"
      peer-as 64500
      neighbor 172.16.114.2
    exit
  exit
  export-inactive-bgp # BGP best-external in VPRN
  no shutdown

```

The import policy sets the LP to 200 for the routes received from CE-4. The configuration on PE-2 is similar, but without import policy. Therefore, the path via PE-1 will be preferred over the path via PE-2.

The **export-inactive-bgp** 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 this example, the **export-inactive-bgp** 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
  router Base
    policy-options
      begin
      prefix-list "172.31.0.0/16"
        prefix 172.31.0.0/16 longer
      exit
      policy-statement "export_172.31.0.0/16"
        entry 10
          from
            prefix-list "172.31.0.0/16"
          exit
          action accept
        exit
      exit
    exit
  commit
exit
exit

```

```

service
  vprn 1 name "VPRN 1" customer 1 create
  autonomous-system 64500
  route-distinguisher 64500:1
  interface "int-CE-4-PE-1_VPRN1" create
    address 172.16.114.2/30
    sap 1/1/1:1 create
    exit
  exit
  interface "int-CE-4-PE-2_VPRN1" create
    address 172.16.124.2/30
    sap 1/1/2:1 create
    exit
  exit
  interface "test_connectedNW" create
    address 172.31.0.1/16
    loopback
  exit
  bgp
    split-horizon
    group "EBGP_1"
      export "export_172.31.0.0/16"
      peer-as 64496
      neighbor 172.16.114.1
      exit
      neighbor 172.16.124.1
      exit
    exit
  exit
  no shutdown

```

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, as follows:

```

# on PE-1, PE-2, PE-3:
configure
  router Base
    bgp
      group "IBGP"
        family ipv4 vpn-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 all PEs:
configure
  router Base
    bgp
      group "IBGP"
        add-paths
          vpn-ipv4 send 2 receive

```

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"
  neighbor 192.0.2.1
    add-paths
      vpn-ipv4 send 2 receive
    exit
  exit
  neighbor 192.0.2.2
    add-paths
      vpn-ipv4 send 2 receive
    exit
  exit
  neighbor 192.0.2.3
    add-paths
      vpn-ipv4 send 2 receive
    exit
  exit

```

The BGP configuration for group "IBGP" on PE-1 is as follows:

```

*A:PE-1# configure router bgp group "IBGP"
*A:PE-1>config>router>bgp>group# info
-----
      family ipv4 vpn-ipv4
      next-hop-self
      peer-as 64496
      add-paths
        ipv4 send 2 receive
        vpn-ipv4 send 2 receive
      exit
      neighbor 192.0.2.5
      exit
-----

```

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: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                             3          10
      64500                                    524284
ub*i  64496:1:172.31.0.0/16                    100        None
192.0.2.2                                15         10
64500                                    524283
-----
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: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 00h00m32s    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: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 00h00m32s    170
  192.0.2.1 (tunneled)              10
172.31.0.0/16 (Backup)             Remote BGP VPN 00h00m32s    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"
    no enable-bgp-vpn-backup
```

Add-path for family VPN-IPv4 with ECMP

The import policy is removed 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
  service
    vprn "VPRN 1"
      bgp
        group "EBGP_1"
          no import
```

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: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                             3           10
      64500                                   524284
u*>i  64496:1:172.31.0.0/16                   100         None
      192.0.2.2                             15          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:PE-3# show router 1 route-table 172.31.0.0/16
```

```
=====
Route Table (Service: 1)
=====
```



```
Dest Prefix[Flags]
Next Hop[Interface Name]      Type   Proto   Age           Pref
-----
172.31.0.0/16                 Remote BGP VPN 00h00m48s 170
192.0.2.1 (tunneled)         10
172.31.0.0/16                 Remote BGP VPN 00h00m48s 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.

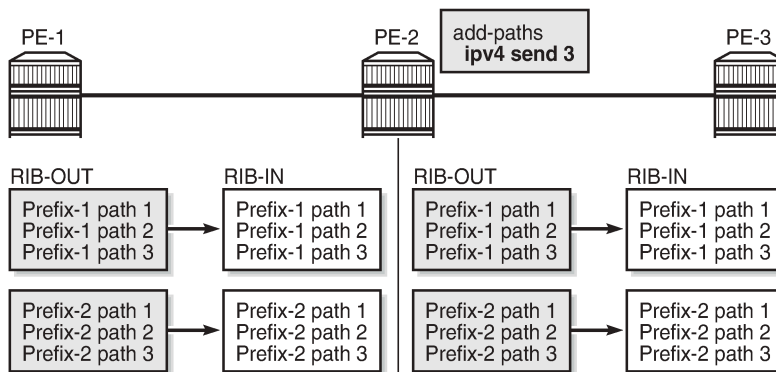
The BGP add-path policy control feature extends the functionality of BGP add-path, which was able to control the number of advertised paths per prefix per address family. This meant that all prefixes that belonged to an address family (such as IPv4, IPv6, and so on) were subject to the same sending limit imposed by the **send** *<send-limit>* 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 14: BGP add-paths before policy control](#) shows a topology for BGP add-paths before policy control.

Figure 14: BGP add-paths before policy control

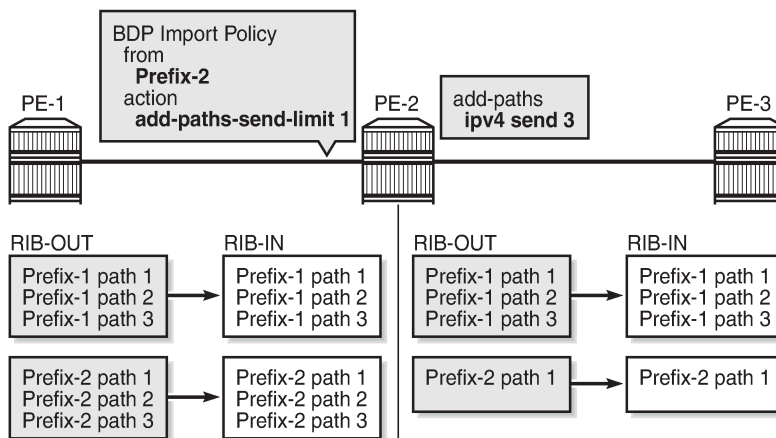


26774

In [Figure 14: 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 15: BGP add-paths after policy control](#) shows a topology for BGP add-paths after policy control.

Figure 15: BGP add-paths after policy control



26775

In [Figure 15: 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** 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:

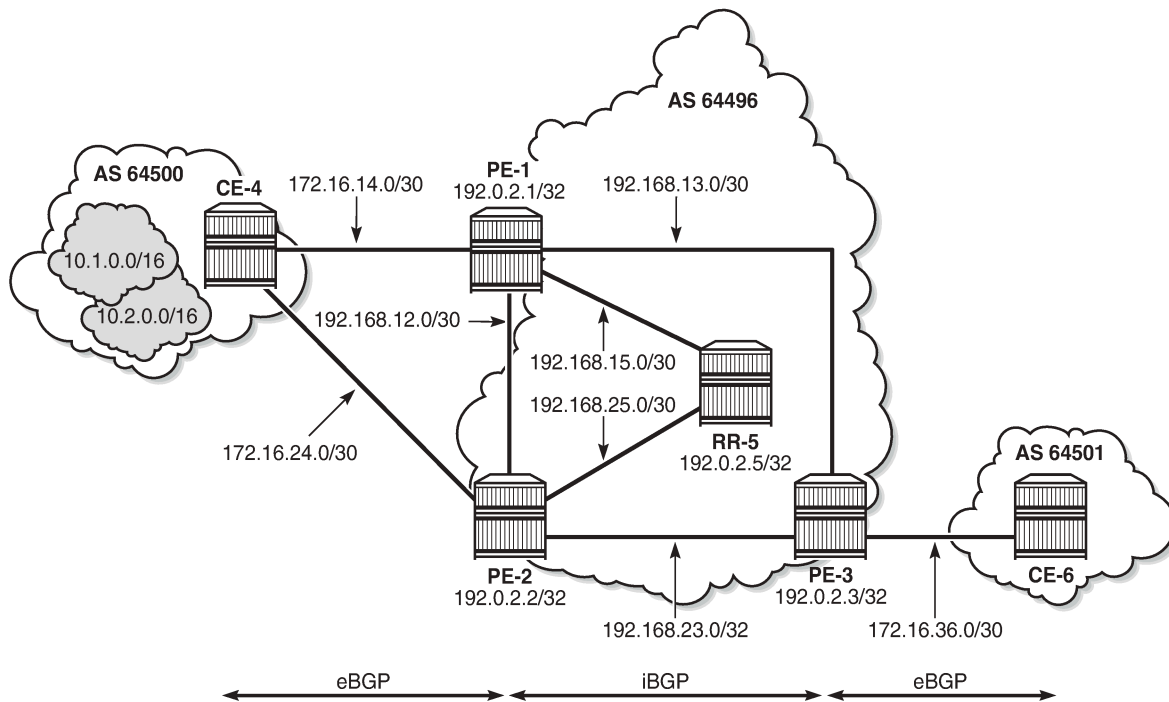
- BGP add-path for address family IPv4 without policy control
- BGP add-path for address family IPv4 with policy control
- BGP add-path for address family VPN-IPv4 with policy control

Example topology

[Figure 16: 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 16: 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
  autonomous-system 64500
  bgp
    rapid-withdrawal
    split-horizon
    group "eBGP"
      export "export-bgp"
      peer-as 64496
      neighbor 172.16.14.1
    exit
  neighbor 172.16.24.1
```

```

        exit
    exit
    no shutdown
exit
policy-options
  begin
  prefix-list "10.1.0.0/16"
    prefix 10.1.0.0/16 longer
  exit
  prefix-list "10.2.0.0/16"
    prefix 10.2.0.0/16 longer
  exit
  policy-statement "export-bgp"
    entry 10
      from
        prefix-list "10.1.0.0/16"
      exit
      action accept
      exit
    exit
    entry 20
      from
        prefix-list "10.2.0.0/16"
      exit
      action accept
      exit
    exit
  exit
exit
commit
exit
interface "int-loopback-1"
  address 10.1.1.1/16
  loopback
  no shutdown
exit
interface "int-loopback-2"
  address 10.2.1.1/16
  loopback
  no shutdown
exit

```

The BGP configuration on CE-6 is similar, except for the export policy.

PE-1 peers with CE-4 in AS 64500 and RR-5 in AS 64496. The BGP configuration on PE-1 is as follows:

```

# on PE-1:
configure
  router
    autonomous-system 64496
    bgp
      rapid-withdrawal
      split-horizon
      group "eBGP"
        peer-as 64500
        neighbor 172.16.14.2
      exit
    exit
    group "iBGP"
      next-hop-self
      peer-as 64496
      add-paths
        ipv4 send 2 receive
      exit
    neighbor 192.0.2.5

```

```

        exit
    exit
    no shutdown
exit

```

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
  autonomous-system 64500
  bgp
    rapid-withdrawal
    split-horizon
    group "iBGP"
      cluster 5.5.5.5
      peer-as 64496
      add-paths
        ipv4 send 2 receive
    exit
    neighbor 192.0.2.1
    exit
    neighbor 192.0.2.2
    exit
    neighbor 192.0.2.3
    exit
  exit
  no shutdown
exit

```

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: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 2023/01/25 17:04:01.502 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 9
    10.2.0.0/16 Path-ID 12
"

```

```

6 2023/01/25 17:03:34.502 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 3
    10.2.0.0/16 Path-ID 4
"

```

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
  ecmp 2
  bgp
    multi-path
      maximum-paths 2
  exit

```

```

*A: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              9          10
      64500                   -
u*>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              12         10
      64500                   -
u*>i  10.2.0.0/16              100        None
      192.0.2.2              4          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 10.2.0/16 to one:

```

# on RR-5
configure
router
  policy-options
  begin
  prefix-list "10.2.0.0/16"
    prefix 10.2.0.0/16 longer
  exit
  policy-statement "import-add-path"
  entry 10
    from
      prefix-list "10.2.0.0/16"
    exit
    action accept
    add-paths-send-limit 1
  exit
  exit
  commit
exit
bgp
  group "iBGP"
    import "import-add-path"
  exit

```

RR-5 sends the following withdrawal message to PE-3:

```
1 2023/01/25 17:07:53.502 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 = 7
    10.2.0.0/16 Path-ID 4
  Total Path Attr Length = 0
"
```

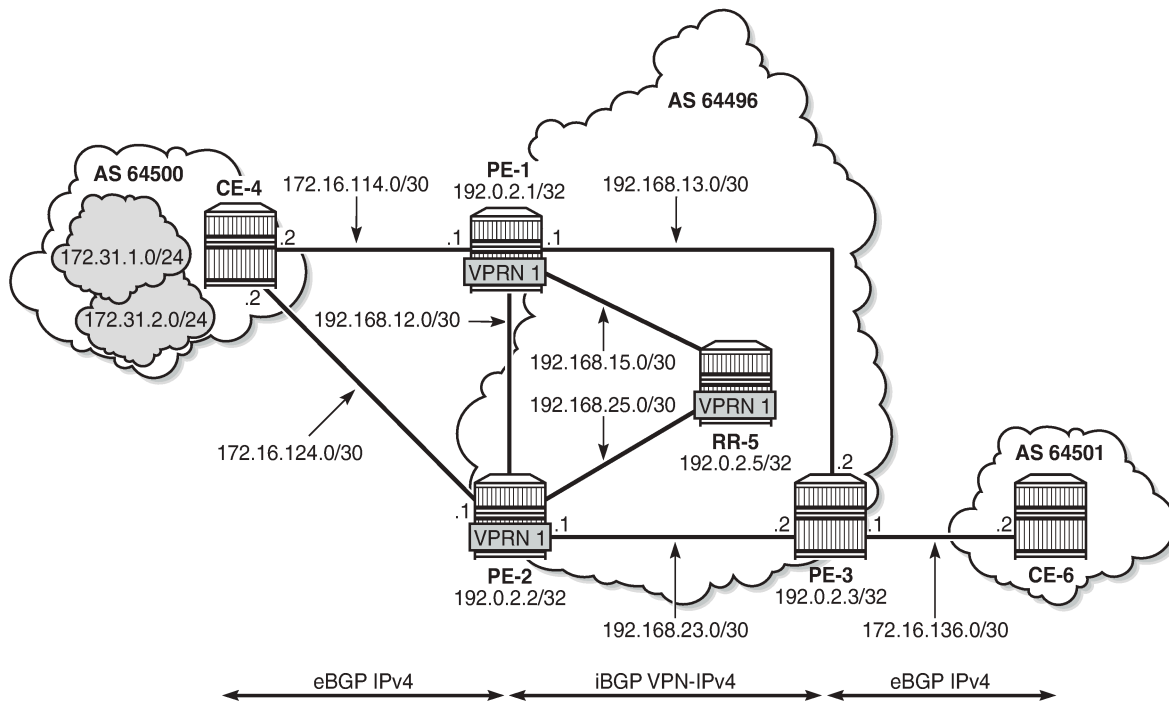
PE-3 deletes the route with Path-ID 12 for prefix 10.2.0.0/16:

```
*A: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                9          10
      64500                      -
u*>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                12         10
      64500                      -
-----
Routes : 3
=====
```

BGP add-path for address family VPN-IPv4 with policy control

[Figure 17: 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 17: 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 1 name "VPRN 1" customer 1 create
  autonomous-system 64496
  route-distinguisher 64496:1
  auto-bind-tunnel
  resolution any
  exit
  vrf-target target:64496:1
  interface "int-PE-1-CE-4-VPRN1" create
  address 172.16.114.1/30
  sap 1/1/c1/2:1 create
  exit
  exit
  bgp
  split-horizon
  group "eBGP-1"
  peer-as 64500
  neighbor 172.16.114.2
  exit
  exit
  no shutdown
  exit
  no shutdown
```

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
  service
    vprn 1 name "VPRN 1" customer 1 create
      autonomous-system 64500
      route-distinguisher 64500:1
      interface "int-CE-4-PE-1-VPRN1" create
        address 172.16.114.2/30
        sap 1/1/c1/1:1 create
        exit
      exit
      interface "int-CE-4-PE-2-VPRN1" create
        address 172.16.124.2/30
        sap 1/1/c1/2:1 create
        exit
      exit
      interface "loopback1-VPRN1" create
        address 172.31.1.1/24
        loopback
      exit
      interface "loopback2-VPRN1" create
        address 172.31.2.1/24
        loopback
      exit
    bgp
      split-horizon
      group "eBGP-1"
        export "export-VPRN1"
        peer-as 64496
        neighbor 172.16.114.1
        exit
        neighbor 172.16.124.1
        exit
      exit
    no shutdown
  exit
no shutdown
```

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
  router
    policy-options
      begin
      prefix-list "172.31.0.0/16"
        prefix 172.31.0.0/16 longer
      exit
      policy-statement "export-VPRN1"
        entry 10
          from
            prefix-list "172.31.0.0/16"
          exit
          action accept
          exit
        exit
      exit
    commit
```

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
  bgp
    group "iBGP"
      family ipv4 vpn-ipv4
```

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
router
  bgp
    group "iBGP"
      add-paths
        vpn-ipv4 send 2 receive
    exit
```

The BGP configuration on PE-1 is as follows:

```
# on PE-1:
configure
router
  bgp
    rapid-withdrawal
    split-horizon
    group "eBGP"
      peer-as 64500
      neighbor 172.16.14.2
    exit
  group "iBGP"
    family ipv4 vpn-ipv4
    next-hop-self
    peer-as 64496
    add-paths
      ipv4 send 2 receive
      vpn-ipv4 send 2 receive
    exit
  neighbor 192.0.2.5
  exit
  no shutdown
exit
```

With add-path enabled for address family VPN-IPv4, PE-1 and PE-2 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 advertises 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: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                               19         10
     64500                                    524284
u*>i 64496:1:172.31.1.0/24                    100        None
     192.0.2.2                               4          10
     64500                                    524284
u*>i 64496:1:172.31.2.0/24                    100        None
     192.0.2.1                               18         10
     64500                                    524284
u*>i 64496:1:172.31.2.0/24                    100        None
     192.0.2.2                               5          10
     64500                                    524284
-----
Routes : 4
=====

```

All routes are used because of the ECMP setting in VPRN 1:

```

# on PE-3:
configure
  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
  router
    policy-options
      begin
        prefix-list "172.31.2.0/24"
          prefix 172.31.2.0/24 longer
        exit
        policy-statement "import-add-path"
          entry 20
            from
              prefix-list "172.31.2.0/24"
            exit
            action accept
              add-paths-send-limit 1
            exit
          exit
        exit
      exit
    exit
  commit

```

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
  bgp
    group "iBGP"
      import "import-add-path"
```

At this point, PE-3 still has two paths for each of the prefixes:

```
*A: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                             19         10
      64500                                   524284
u*>i  64496:1:172.31.1.0/24                   100        None
      192.0.2.2                             4          10
      64500                                   524284
u*>i  64496:1:172.31.2.0/24                   100        None
      192.0.2.1                             18         10
      64500                                   524284
u*>i  64496:1:172.31.2.0/24                   100        None
      192.0.2.2                             5          10
      64500                                   524284
-----
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
  bgp
    vpn-apply-import
```

Upon application of this configuration, RR-5 sends the following withdrawal to PE-3:

```
43 2023/01/25 17:12:57.502 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 5
"
```

PE-3 now has a single route for prefix 172.31.2.0/24 in its BGP routing table:

```
*A: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                               19         10
     64500                                    524284
u*>i 64496:1:172.31.1.0/24                   100        None
     192.0.2.2                               4          10
     64500                                    524284
u*>i 64496:1:172.31.2.0/24                   100        None
     192.0.2.1                               18         10
     64500                                    524284
-----
Routes : 3
=====
```

PE-3 has installed a single route for prefix 172.31.2.0/24 in its VPRN route table:

```
*A: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 00h02m43s 170
     192.0.2.1 (tunneled)   10
172.31.1.0/24               Remote BGP VPN 00h02m43s 170
     192.0.2.2 (tunneled)   10
172.31.2.0/24               Remote BGP VPN 00h01m14s 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

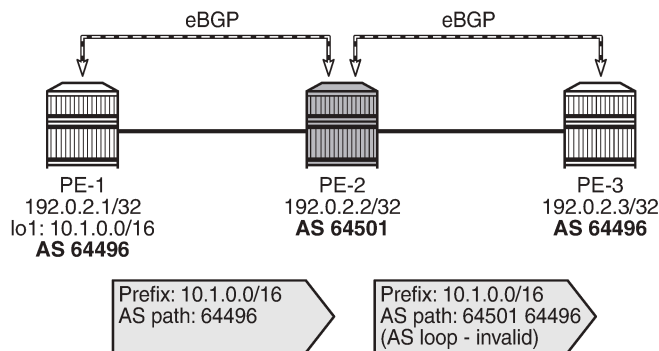
The information and configuration in this chapter are based on SR OS Release 20.5.R1. In SR OS Releases earlier than 19.7.R1, BGP Autonomous System (AS) override is only supported in VPRN BGP instances; BGP AS override in the base router is supported in SR OS Release 19.7.R1 and later.

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 18: 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 18: PE-2 detects AS-path loop and advertises the route to PE-3 as invalid



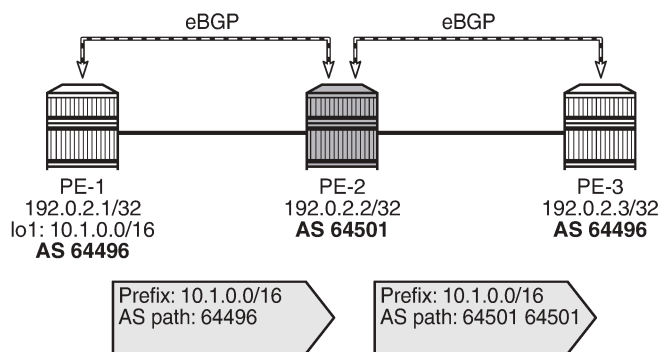
36187

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 19: 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 19: 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|discard-route|ignore-loop|off}
```

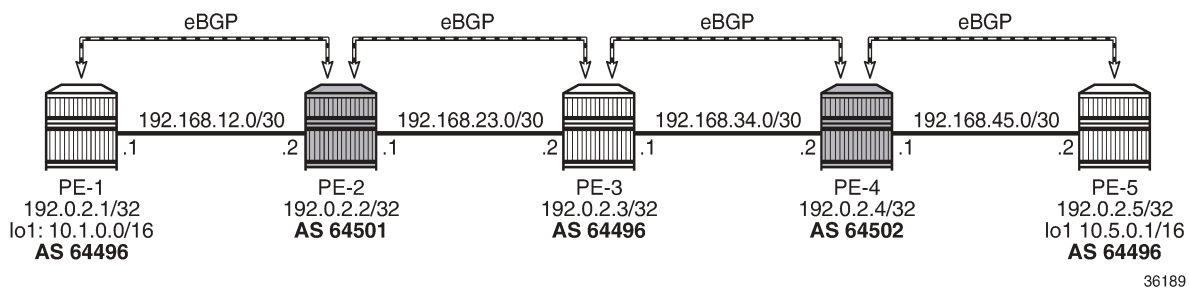
```
configure service vprn bgp / group / neighbor loop-detect {drop-peer|discard-route|ignore-loop|off}
```

With the **ignore-loop** parameter configured, the BGP routes are ignored when having an AS-loop flag but BGP peering remains established.

Configuration

[Figure 20: 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 20: Example topology



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
router Base
  autonomous-system 64501
  bgp
    split-horizon
    group "eBGP"
      family ipv4
        neighbor 192.168.23.2
          peer-as 64496
        exit
        neighbor 192.168.12.1
          peer-as 64496
        exit
      exit
    exit
  no shutdown
exit
```

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:PE-2# show router bgp neighbor 192.168.12.1 detail | match "AS Override"
Multihop          : 0 (Default)      AS Override      : Disabled
```

```
*A: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
  router Base
    autonomous-system 64496
    policy-options
      begin
      prefix-list "10.1.0.0/16"
        prefix 10.1.0.0/16 longer
      exit
      policy-statement "export-prefix_10.1"
        entry 10
          from
            prefix-list "10.1.0.0/16"
          exit
          action accept
        exit
      exit
    exit
  commit
exit
bgp
  split-horizon
  group "eBGP"
    family ipv4
    peer-as 64501
    neighbor 192.168.12.2
      export "export-prefix_10.1"
    exit
  exit
  no shutdown
exit
```

PE-2 receives the BGP route from PE-1 with AS-path 64496, as follows:

```
*A:PE-2# show router bgp neighbor 192.168.12.1 received-routes
=====
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)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  10.1.0.0/16                None       None
      192.168.12.1           None       0
      64496                  -
-----
Routes : 1
=====

```

PE-2 detects that the ASN 64496 in the AS-path equals the ASN of the peer AS of PE-3, so an AS loop is detected and PE-2 advertises this route to PE-3 as an invalid route:

```

*A:PE-2# show router bgp neighbor 192.168.23.2 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                Label
-----
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: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: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

```

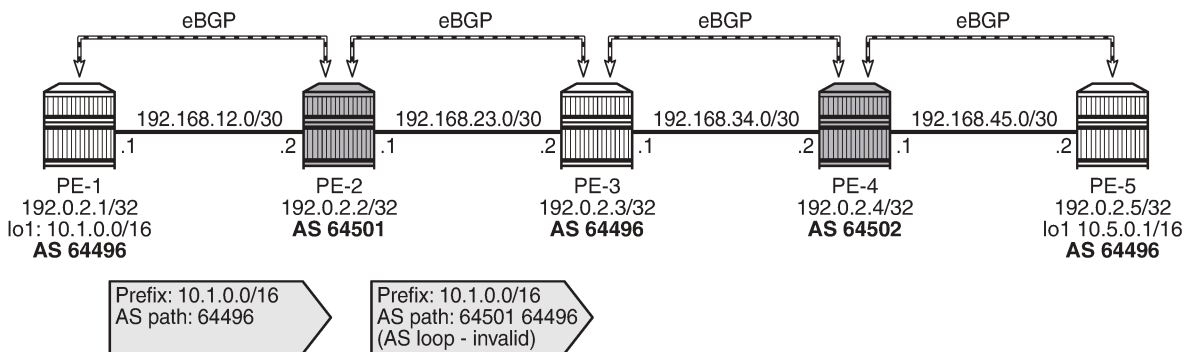
```

Origin codes : i - IGP, e - EGP, ? - incomplete

=====
BGP IPv4 Routes
=====
Flag Network LocalPref MED
Nexthop (Router) Path-Id IGP Cost
As-Path Label
-----
No Matching Entries Found.
=====
    
```

Figure 21: 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 21: 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
    exit
  exit
    
```

With this configuration, BGP AS override is configured for both BGP neighbors, as follows:

```

*A:PE-2# show router bgp neighbor 192.168.12.1 detail | match "AS Override"
Multihop           : 0 (Default)      AS Override       : Enabled

*A: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: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)                        Path-Id    IGP Cost
  As-Path                                  Label
-----
u*>i 10.1.0.0/16                            None       None
      192.168.12.1                          None       0
      64496                                  -
-----
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: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                               LocalPref  MED
  Nexthop (Router)                        Path-Id    IGP Cost
  As-Path                                  Label
-----
u*>i 10.1.0.0/16                            None       None
      192.168.23.1                          None       0
      64501 64501                              -
-----
Routes : 1
=====

```

PE-4 receives the following BGP route:

```

*A: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                               LocalPref MED
  Nextthop (Router)                       Path-Id   IGP Cost
  As-Path                                            Label
-----
u*>i 10.1.0.0/16                            None     None
     192.168.34.1                          None     0
     64496 64501 64501                    -
-----
Routes : 1
=====
    
```

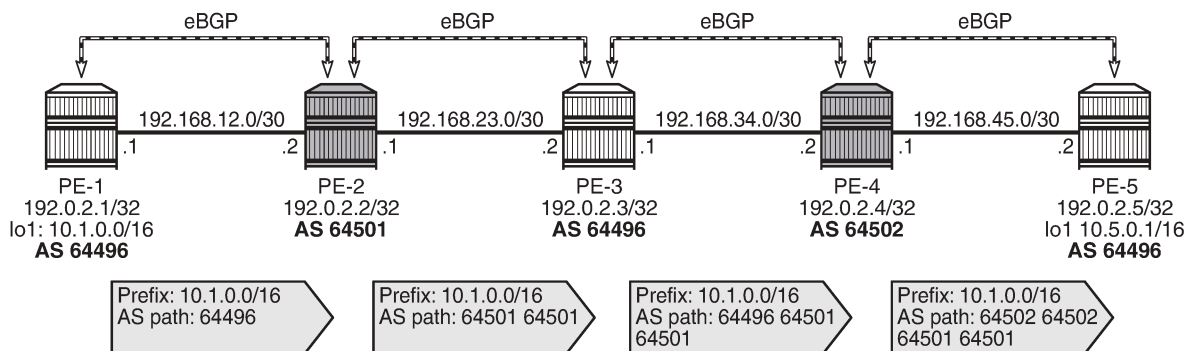
PE-4 detects an AS loop when advertising this route to its peer PE-5 in AS 64496, so it replaces ASN 64496 in the AS-path with its own ASN 64502. PE-5 receives the following valid route from PE-4:

```

*A: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 22: 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 22: No AS loop when BGP AS override is enabled for group "eBGP" on PE-2 and PE-4

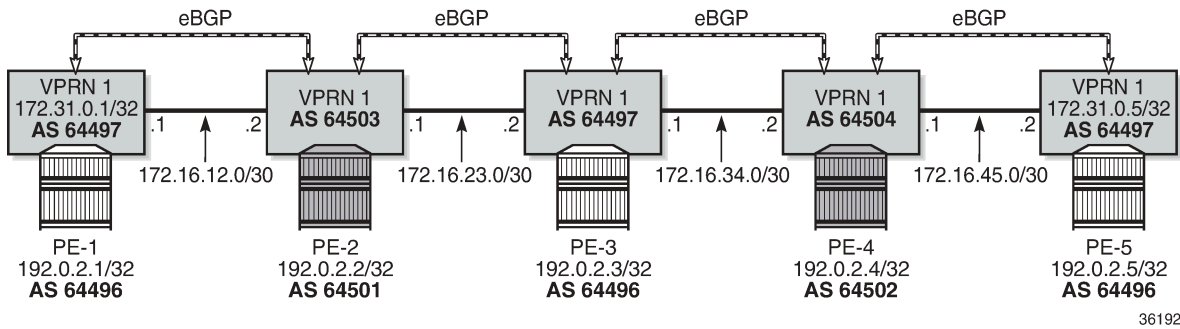


36191

Default: BGP AS override disabled in VPRN

Figure 23: Example topology with VPRN 1 on all PEs shows the example topology with VPRN 1 configured on all PEs.

Figure 23: 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 1 name "VPRN 1" customer 1 create
  router-id 172.31.0.2
  autonomous-system 64503
  route-distinguisher 64503:1
  vrf-target target:1:1
  interface "int-VPRN1-PE-2-PE-1" create
  address 172.16.12.2/30
  sap 1/1/2:1 create
  exit
  no shutdown
exit
interface "int-VPRN1-PE-2-PE-3" create
address 172.16.23.1/30
sap 1/1/1:1 create
exit
no shutdown
exit
interface "system" create
address 172.31.0.2/32
loopback
no shutdown
exit
bgp
  split-horizon
  group "eBGP"
  local-as 64503
  peer-as 64497
  neighbor 172.16.12.1
  exit
  neighbor 172.16.23.2
  exit
exit
exit
```

```
no shutdown
```

The service configuration on the other nodes is similar. The IP addresses and ASNs are shown in [Figure 23: 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
  router Base
    policy-options
      begin
      prefix-list "172.31.0.0/16"
        prefix 172.31.0.0/16 longer
      exit
      policy-statement "export-prefix_172.31"
        entry 10
          from
            protocol direct
            prefix-list "172.31.0.0/16"
          exit
          to
            protocol bgp
          exit
          action accept
          exit
        exit
      exit
    exit
  commit
  exit
exit
service
  vprn 1 name "VPRN 1" customer 1 create
    router-id 172.31.0.1
    autonomous-system 64497
    route-distinguisher 64497:1
    vrf-target target:1:1
    interface "int-VPRN1-PE-1-PE-2" create
      address 172.16.12.1/30
      sap 1/1/1:1 create
      exit
      no shutdown
    exit
    interface "system" create
      address 172.31.0.1/32
      loopback
      no shutdown
    exit
    bgp
      split-horizon
      group "eBGP"
        local-as 64497
        peer-as 64503
        neighbor 172.16.12.2
          export "export-prefix_172.31"
        exit
      exit
    exit
  exit
no shutdown
```

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:

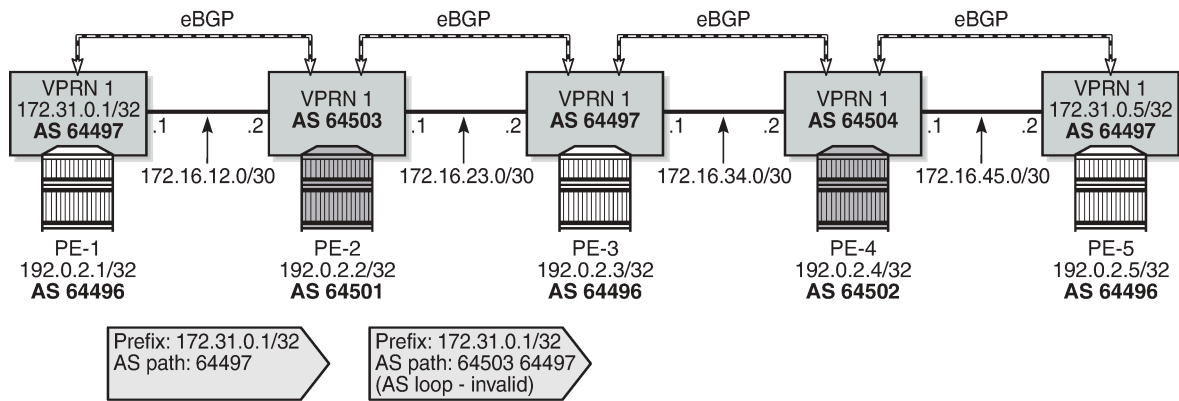
```
*A:PE-2# show router 1 bgp neighbor 172.16.12.1 received-routes
=====
BGP Router ID:172.31.0.2      AS:64503      Local AS:64503
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
u*>i 172.31.0.1/32           n/a        None
      172.16.12.1           None        0
      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: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                Path-Id    Label
-----
i    172.31.0.1/32           n/a        None
      172.16.23.1           None        0
      64503 64497           -
-----
Routes : 1
=====
```

Figure 24: 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 24: AS loop when BGP AS override is not configured in VPRN 1 on PE-2



36193

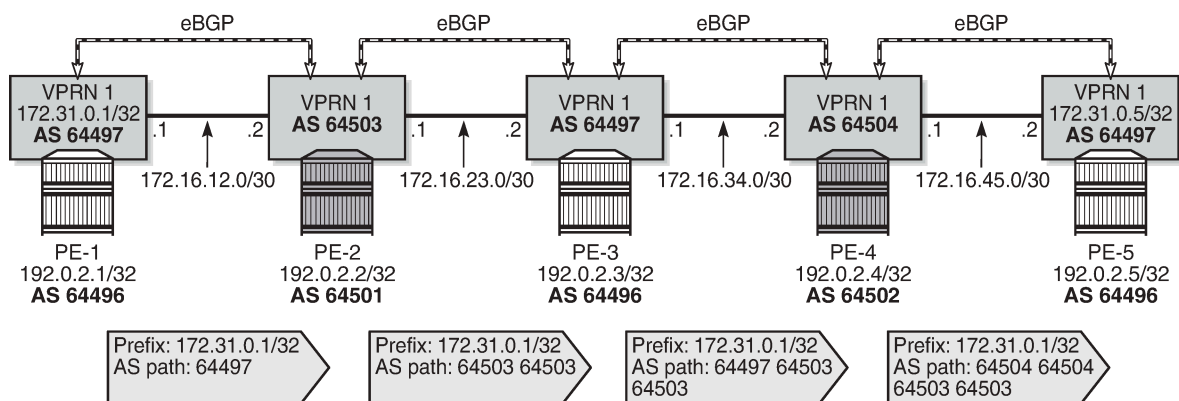
BGP AS override in VPRN

On PE-2 and PE-4, **as-override** is configured in VPRN 1 for group "eBGP", as follows:

```
# on PE-2, PE-4:
configure
service
vprn "VPRN 1"
  bgp
    group "eBGP"
      as-override
    exit
  exit
```

Figure 25: 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 25: 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: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
      Nexthop (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
=====
```

As a result of the **as-override** setting, VPRN 1 on PE-3 receives the following valid route where ASN 64497 is replaced by ASN 64503:

```
*A: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: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: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](#)

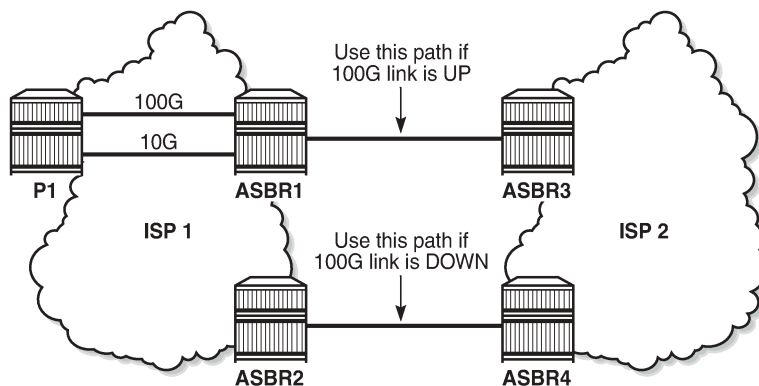
Applicability

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 26: Conditional BGP Route Advertisement - ISP Peering](#) shows an example in which this feature can bring flexibility in an ISP peering scenario.

Figure 26: Conditional BGP Route Advertisement - ISP Peering



26862

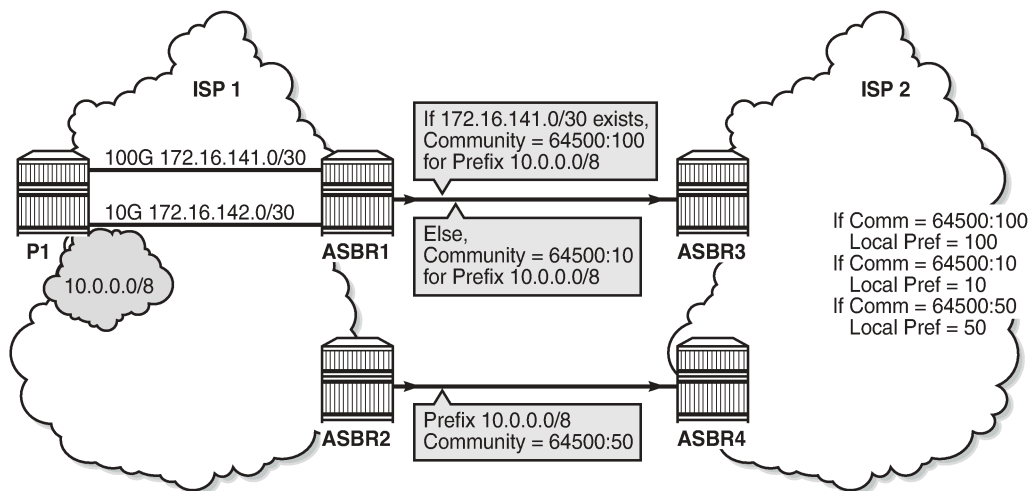
ISP 1 and ISP 2 have two peering points; a first between ASBR1 and ASBR3, and a second between ASBR2 and ASBR4. For redundancy, ISP 1 has two links between ASBR1 and the internal P1 router, one with 100 Gb/s and the other with 10 Gb/s capacity. According to the service agreement, ISP 1 instructs ISP 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 27: Conditional BGP Route Advertisement Implementation Example shows the implementation using the example in Figure 26: Conditional BGP Route Advertisement - ISP Peering.

Figure 27: 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

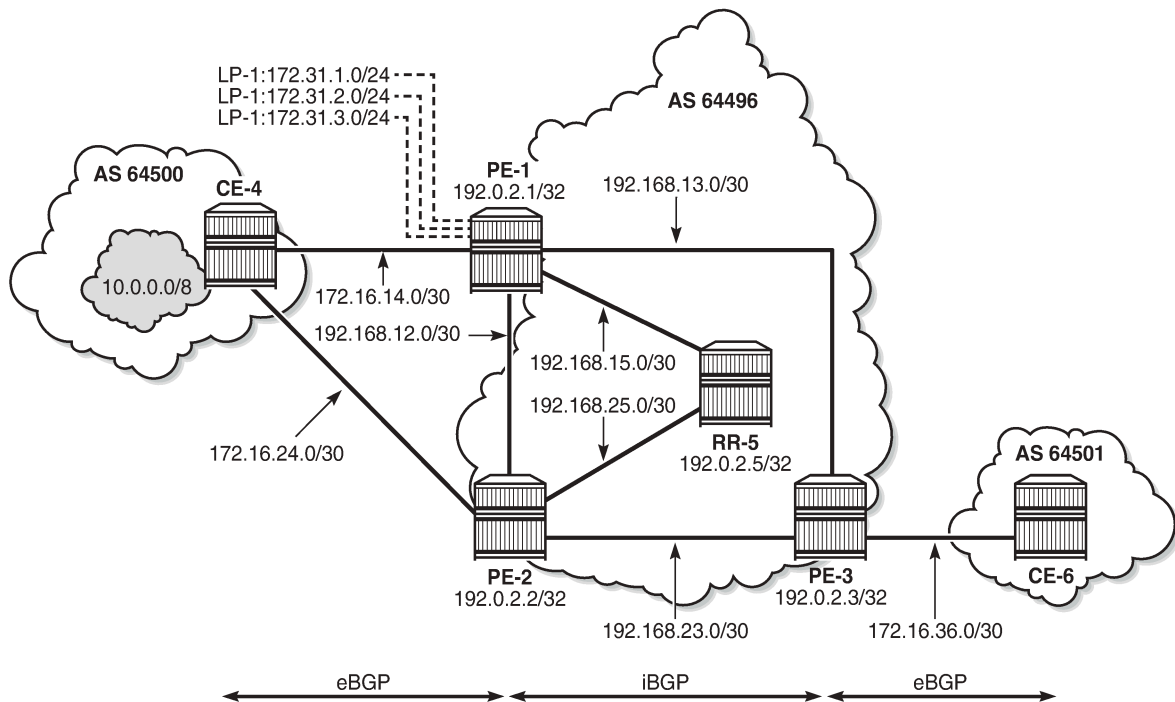
The following configuration examples are covered in this section:

- [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 28: 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 28: 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
  router Base
    interface "int-loopback-1"
      address 10.1.1.1/8
      loopback
      no shutdown
    exit
    autonomous-system 64500
    policy-options
      begin
        prefix-list "10.0.0.0/8"
          prefix 10.0.0.0/8 longer
      exit
      policy-statement "policy-export-bgp"
        entry 10
          from
            prefix-list "10.0.0.0/8"
          exit
          action accept
        exit
      exit
    exit
  commit
exit
bgp
  rapid-withdrawal
  split-horizon
  group "eBGP"
    export "policy-export-bgp"
    peer-as 64496
    neighbor 172.16.14.1
    exit
    neighbor 172.16.24.1
    exit
  exit
  no shutdown
exit
exit all
```

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
  split-horizon
  group "eBGP"
    peer-as 64500
    neighbor 172.16.14.2
  exit
  group "iBGP"
    next-hop-self
    peer-as 64496
    neighbor 192.0.2.5
  exit
  no shutdown
exit
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
    split-horizon
    group "iBGP"
      cluster 5.5.5.5
      peer-as 64496
      neighbor 192.0.2.1
    exit
      neighbor 192.0.2.2
    exit
      neighbor 192.0.2.3
    exit
  exit
  no shutdown
exit
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"
    address 172.31.1.1/24
    loopback
    no shutdown
  exit
  interface "int-loopback-2"
    address 172.31.2.1/24
    loopback
    no shutdown
  exit
  interface "int-loopback-3"
    address 172.31.3.1/24
    loopback

```

```
no shutdown
exit
```

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:RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match "^Nexthop|^Community|^Pref"
expression
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
  router Base
    policy-options
      begin
        prefix-list "10.0.0.0/8"
          prefix 10.0.0.0/8 longer
        exit
      prefix-list "prefix-conditional-routes"
        prefix 172.31.1.0/24 longer
        prefix 172.31.2.0/24 longer
        prefix 172.31.3.0/24 longer
      exit
      community "64500:10" members "64500:10"
      community "64500:100" members "64500:100"
      policy-statement "policy-bgp-community"
        entry 10
          conditional-expression
            route-exists "[prefix-conditional-routes]"
          exit
          from
            prefix-list "10.0.0.0/8"
          exit
          action accept
            community add "64500:100"
          exit
        exit
        entry 20
          from
            prefix-list "10.0.0.0/8"
          exit
          action accept
            community add "64500:10"
          exit
        exit
      exit
    exit
  commit
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
  router Base
    policy-options
      begin
        prefix-list "10.0.0.0/8"
          prefix 10.0.0.0/8 longer
        exit
        community "64500:50" members "64500:50"
        policy-statement "policy-bgp-community"
          entry 10
            from
              prefix-list "10.0.0.0/8"
            exit
            action accept
              community add "64500:50"
            exit
          exit
        exit
      exit
    exit
  commit
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 bgp group "iBGP" export "policy-bgp-community"
```

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:RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match "^Nexthop|^Community|^Pref"
expression
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
  router Base
    policy-options
      begin
        community "64500:10" members "64500:10"
        community "64500:50" members "64500:50"
        community "64500:100" members "64500:100"
        policy-statement "policy-bgp-preference"
          entry 10
            from
              community "64500:100"
```

```

        exit
        action accept
            local-preference 100
        exit
    exit
    entry 20
        from
            community "64500:50"
        exit
        action accept
            local-preference 50
        exit
    exit
    entry 30
        from
            community "64500:10"
        exit
        action accept
            local-preference 10
        exit
    exit
    exit
    commit
    exit all

```

The policy is applied on RR-5:

```

# On RR-5:
configure router bgp group "iBGP" import "policy-bgp-preference"

```

The following command output shows that the correct local preference values are applied on the routes received from PE-1 and PE-2:

```

*A:RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match "^NextHop|^Community|^Pref"
expression
NextHop      : 192.0.2.1
Local Pref.  : 100
Community    : 64500:100
Interface Name : int-RR-5-PE-1
NextHop      : 192.0.2.2
Local Pref.  : 50
Community    : 64500:50
Interface Name : int-RR-5-PE-2
TieBreakReason : LocalPref

```

RR-5 advertises the route with local preference of 100 to PE-3, with next hop PE-1:

```

*A:PE-3# show router bgp routes 10.0.0.0/8 hunt brief | match "^NextHop|^Community|^Pref"
expression
NextHop      : 192.0.2.1
Local Pref.  : 100
Community    : 64500:100
Interface Name : int-PE-3-PE-1

```

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 interface "int-loopback-1" shutdown

```

PE-1 still advertises the prefix 10.0.0.0/8 with the community 64500:100:

```
*A:RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match "^NextHop|^Community|^Pref"
expression
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 interface "int-loopback-2" shutdown
```

PE-1 still advertises the prefix 10.0.0.0/8 with the community 64500:100:

```
*A:RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match "^NextHop|^Community|^Pref"
expression
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 interface "int-loopback-3" shutdown
```

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:RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match "^NextHop|^Community|^Pref"
expression
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:PE-3# show router bgp routes 10.0.0.0/8 hunt brief | match "^NextHop|^Community|^Pref"
expression
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 interface int-loopback-[1..3] no shutdown
```



Note:

Do not use quotes in the interface name when using **ranges**, because it is treated as a new interface creation.

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
  router Base
    policy-options
      begin
        policy-statement "policy-bgp-community"
          entry 10
            conditional-expression
              route-exists "[prefix-conditional-routes] all"
            exit
          exit
        exit
      exit
    commit
  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 interface "int-loopback-1" shutdown
```

PE-1 now advertises the prefix 10.0.0.0/8 with the community 64500:10:

```
*A:RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match "^Nexthop|^Community|^Pref"
expression
Nexthop      : 192.0.2.2
Local Pref.  : 50
Community    : 64500:50      Interface Name : int-RR-5-PE-2
Nexthop      : 192.0.2.1
Local Pref.  : 10
Community    : 64500:10     Interface Name : int-RR-5-PE-1
TieBreakReason : LocalPref
```

RR-5 advertises prefix 10.0.0.0/8 to PE-3 with the next-hop address of PE-2:

```
*A:PE-3# show router bgp routes 10.0.0.0/8 hunt brief | match "^Nexthop|^Community|^Pref"
expression
Nexthop      : 192.0.2.2
Local Pref.  : 50
Community    : 64500:50     Interface Name : int-PE-3-PE-2
```

BGP Conditional Route Advertisement Using "none" Prefix List Match

The loopback interfaces on PE-1 are re-enabled:

```
# On PE-1:
configure router interface int-loopback-[1..3] no shutdown
```

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
  router Base
    policy-options
      begin
        policy-statement "policy-bgp-community"
          entry 10
            conditional-expression
              route-exists "[prefix-conditional-routes] none"
            exit
          exit
        exit
      exit
    commit
  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:RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match "^NextHop|^Community|^Pref"
expression
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 interface int-loopback-[1..3] shutdown
```

PE-1 now advertises the prefix 10.0.0.0/8 with the community 64500:100:

```
*A:RR-5# show router bgp routes 10.0.0.0/8 hunt brief | match "^NextHop|^Community|^Pref"
expression
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:PE-3# show router bgp routes 10.0.0.0/8 hunt brief | match "^Nexthop|^Community|^Pref"
expression
Nexthop      : 192.0.2.1
Local Pref.  : 100
Community    : 64500:100
Interface Name : int-PE-3-PE-1
```

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 configuration in this chapter are based on SR OS Release 20.7.R1. BGP Delayed Route Advertisement is supported in SR OS Release 19.7.R1 and later.

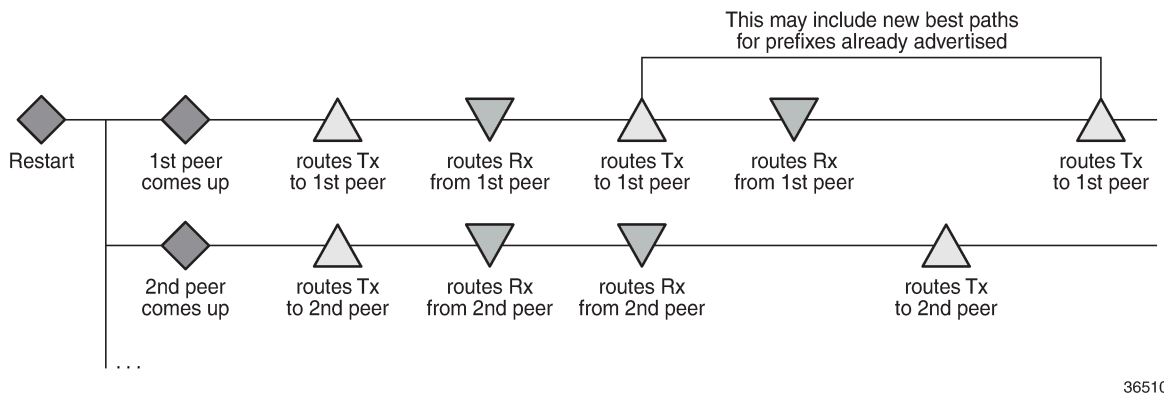
Overview

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 29: Default SR OS behavior when the BGP process restarts](#).

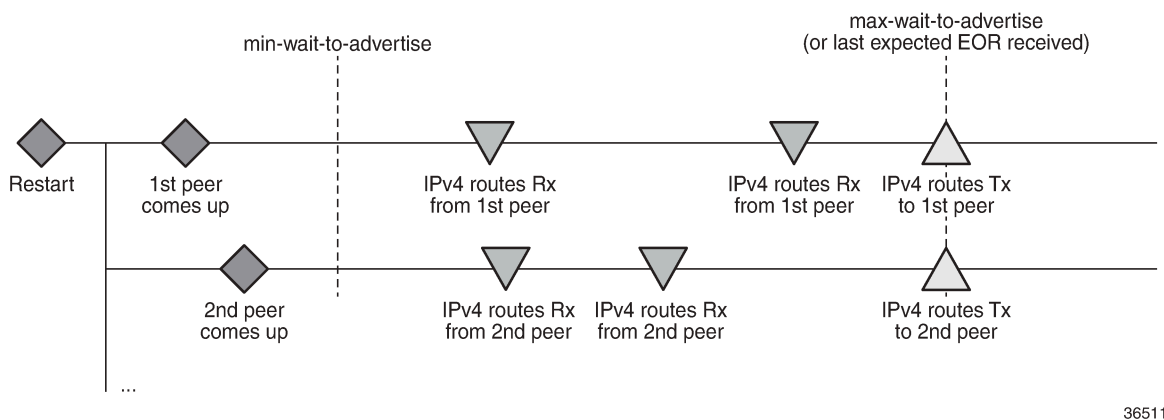
Figure 29: Default SR OS behavior when the BGP process restarts



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 30: BGP convergence tuning with delayed route advertisement](#) shows the BGP convergence tuning.

Figure 30: BGP convergence tuning with delayed route advertisement



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:

```
159 2020/08/06 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 30: 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.

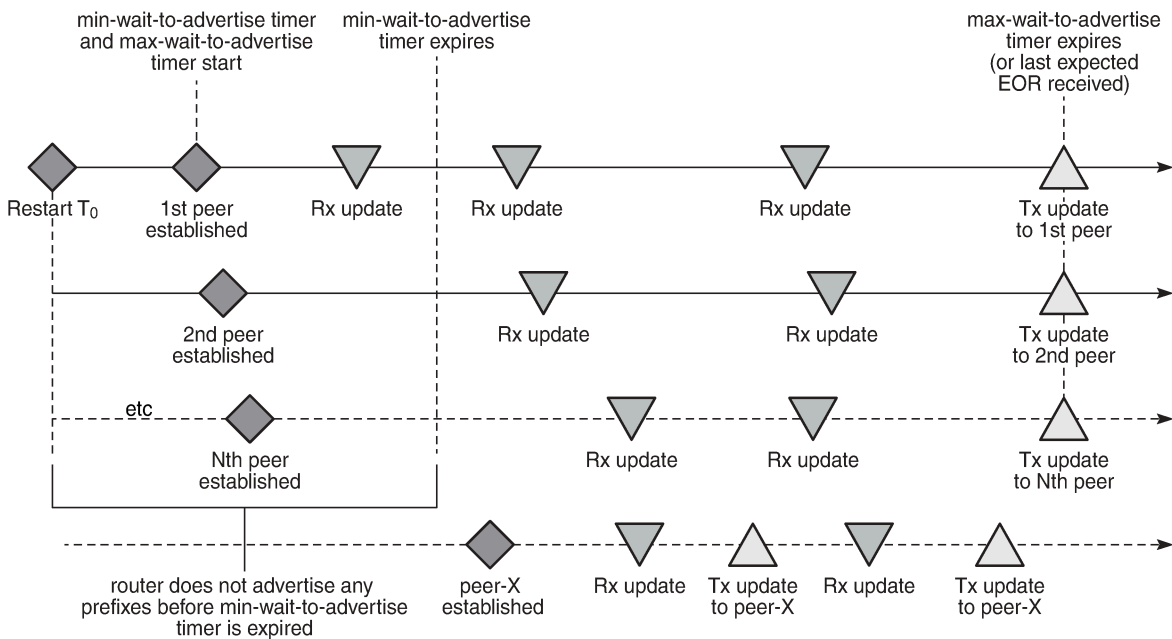


Note:

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 31: 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.

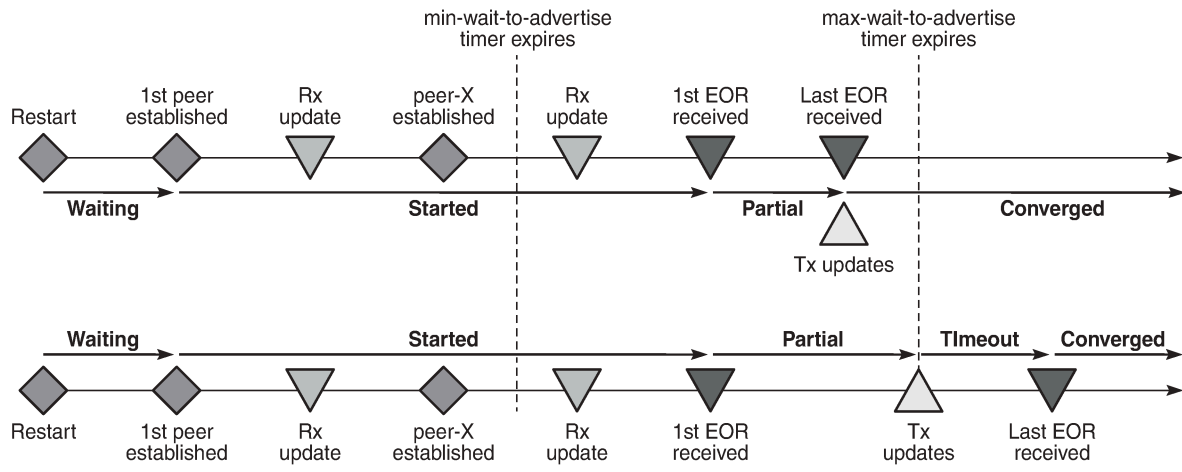
Figure 31: BGP convergence timers



36512

The BGP convergence process can be monitored with the **show router bgp convergence** command. [Figure 32: BGP convergence states](#) shows the different BGP convergence states.

Figure 32: 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 33: 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 33: Example topology

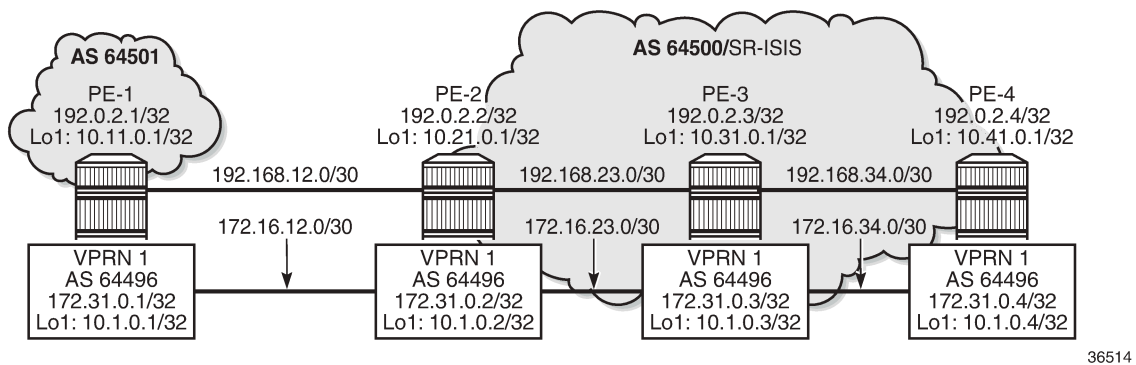


Figure 33: Example topology only shows the IPv4 addresses, but all interfaces also have IPv6 addresses.

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
      group "eBGP"
        local-as 64500
        peer-as 64501
        local-address "int-PE-2-PE-1"
        neighbor 192.168.12.1
          family ipv4
            next-hop-self
            export "export-10.21"
        exit
        neighbor 2001:db8::12:1
          family ipv6
            next-hop-self
            export "export-10:21"
        exit
      exit
    group "iBGP-IPv4"
      family ipv4
        cluster 192.0.2.2
        peer-as 64500
        neighbor 192.0.2.3
          next-hop-self
          export "export-10.21"
        exit
        neighbor 192.0.2.4
```



```

        next-hop-self
        export "export-10.21"
    exit
exit
group "iBGP-IPv6"
    family ipv6
    cluster 192.0.2.2
    peer-as 64500
    neighbor 2001:db8::2:3
        next-hop-self
        export "export-10:21"
    exit
    neighbor 2001:db8::2:4
        next-hop-self
        export "export-10:21"
    exit
exit
no shutdown

```

The export policies are the following:

```

# on PE-2:
configure
    router Base
        policy-options
            begin
            prefix-list "10.21.0.0/16"
                prefix 10.21.0.0/16 longer
            exit
            prefix-list "_::10:21_"
                prefix 2001:db8::10:21:0:0/120 longer
            exit
            policy-statement "export-10.21"
                entry 10
                    from
                        prefix-list "10.21.0.0/16"
                    exit
                    action accept
                exit
            exit
            policy-statement "export-10:21"
                entry 10
                    from
                        prefix-list "_::10:21_"
                    exit
                    action accept
                exit
            exit
        exit
    exit

```

The BGP configuration in the base router is similar on the other PEs, with similar export policies.

BGP is also configured in VPRN 1, with similar export policies. On RR PE-2, the BGP configuration in VPRN 1 is as follows:

```

# on PE-2
configure
    service
        vprn 1 name "VPRN 1" customer 1 create
            autonomous-system 64496
            bgp
                router-id 172.31.0.2

```

```

split-horizon
group "iBGP-VPRN1"
  cluster 172.31.0.2
  peer-as 64496
  neighbor 172.31.0.1
    family ipv4
    local-address 172.31.0.2
    export "export-10.1"
  exit
  neighbor 172.31.0.3
    family ipv4
    local-address 172.31.0.2
    export "export-10.1"
  exit
  neighbor 172.31.0.4
    family ipv4
    local-address 172.31.0.2
    export "export-10.1"
  exit
  neighbor 2001:db8::31:0:1
    family ipv6
    export "export-10:1"
  exit
  neighbor 2001:db8::31:0:3
    family ipv6
    export "export-10:1"
  exit
  neighbor 2001:db8::31:0:4
    family ipv6
    export "export-10:1"
  exit
exit
no shutdown

```

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: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: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
        exit
        family ipv6
          max-wait-to-advertise 30
        exit
      exit
    exit
  info
  exit
exit
service
  vprn "VPRN 1"
    bgp
      convergence
        min-wait-to-advertise 60
        family ipv4
          max-wait-to-advertise 180
        exit
        family ipv6
          max-wait-to-advertise 180
        exit
      exit
    exit
```

With this configuration, the BGP converged state on PE-2 changes to "waiting", because no BGP sessions are reestablished yet, so no BGP convergence tuning has taken place.

```
*A: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      : 00h00m00s
Last session established time       : 00h00m00s
Max Wait advertise timer           : 30
Converged peers                    : 3
Converged state                    : waiting
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 : 00h00m00s
Last session established time  : 00h00m00s
Max Wait advertise timer     : 30
Converged peers              : 3
Converged state               : waiting
Converged time                : N/A
=====

```

The **show router 1 bgp convergence** command shows a similar output for VPRN 1, but is not shown here.

The following **clear** command on PE-2 causes BGP to restart:

```

# on PE-2:
clear router bgp protocol

```

When BGP restarts, the converged state remains "waiting" until the first peer is established.

With the first peer established, the converged state changes to "started", as follows:

```

*A: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  : 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: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: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: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: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: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: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 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:

```
*A:PE-1# configure router bgp send-default
- no send-default
- send-default [ipv4] [ipv6] [export-policy <export-policy>]

<ipv4>           : keyword - provision support of the specific family
<ipv6>           : keyword - provision support of the specific family
```

```
<export-policy> : [64 chars max]
```

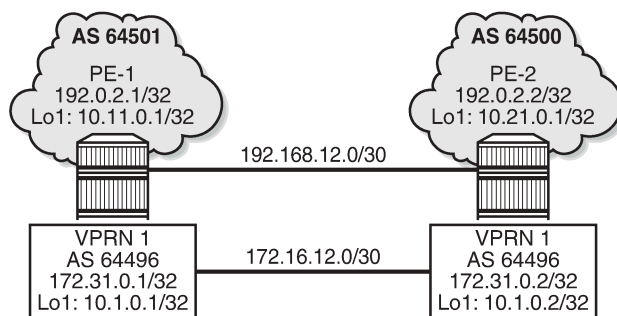
Before modification by a send-default export policy, the properties of the artificially generated default route are as follows:

- 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 34: 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.

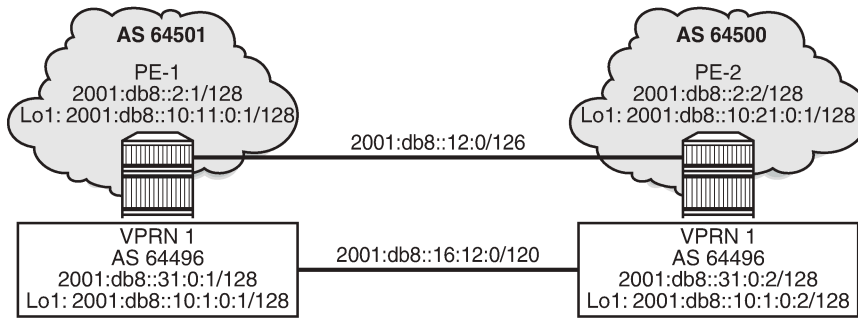
Figure 34: Example topology with IPv4 addresses



36515

Figure 35: Example topology with IPv6 addresses shows the same example topology with IPv6 addresses.

Figure 35: 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
      group "eBGP"
        peer-as 64500
        local-as 64501
        neighbor 192.168.12.2
          family ipv4
          local-address "int-PE-1-PE-2"
          disable-communities large
        exit
      neighbor 2001:db8::12:2
        family ipv6
        local-address 2001:db8::12:1
      exit
    exit
  no shutdown
exit
```

On PE-1, the BGP configuration in VPRN 1 is as follows:

```
# on PE-1:
configure
  service
    vprn 1 name "VPRN 1" customer 1 create
    autonomous-system 64496
    ---snip---
    bgp
      router-id 172.31.0.1
      split-horizon
      group "iBGP-VPRN1"
        type internal
        neighbor 172.31.0.2
```

```

        family ipv4
        local-address 172.31.0.1
        disable-communities large
    exit
    neighbor 2001:db8::31:0:2
        family ipv6
    exit
exit
no shutdown
exit
---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: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: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: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-route-entry 0.0.0.0/0
      black-hole
      no shutdown
    exit
  exit
  static-route-entry ::/0
    black-hole
    no shutdown
  exit
exit
service
  vprn "VPRN 1"
    static-route-entry 0.0.0.0/0
      black-hole
      no shutdown
    exit
  exit
  static-route-entry ::/0
    black-hole
    no shutdown
  exit
exit
exit
```

The following export policies are configured for prefixes 0.0.0.0/0 and ::/0.

```
# on PE-1:
configure
  router Base
    policy-options
      begin
        prefix-list "route_0/0"
          prefix 0.0.0.0/0 exact
        exit
        prefix-list "route_::/0"
          prefix ::/0 exact
```

```

exit
policy-statement "export-route_0/0"
  entry 10
    from
      prefix-list "route_0/0"
    exit
    action accept
      origin igp
    exit
  exit
exit
policy-statement "export-route_::/0"
  entry 10
    from
      prefix-list "route_::/0"
    exit
    action accept
      origin igp
    exit
  exit
exit
commit
exit

```

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 "export-route_0/0" "export-route_::/0"
      exit
    exit

```

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 "export-route_0/0" "export-route_::/0"
      exit
    exit

```

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: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: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: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: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: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 ipv6
      exit
    info
  exit
exit
service
  vprn "VPRN 1"
    bgp
      send-default ipv4 ipv6

```

```

exit
  exit
exit
    
```

The following BGP summary on PE-2 shows that in each BGP session, one route is received and active, and one route is advertised:

```

*A: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      25   0 00h09m03s 1/1/1 (IPv4)
                25   0
2001:db8::12:1
Def. Instance 64501      24   0 00h09m01s 1/1/1 (IPv6)
                23   0

172.31.0.1
Svc: 1         64496      23   0 00h08m54s 1/1/1 (IPv4)
                23   0
2001:db8::31:0:1
Svc: 1         64496      23   0 00h08m54s 1/1/1 (IPv6)
                22   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 on PE-2 are different from the ones received before the **send-default** command was configured, as follows:

```

*A: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*>? 0.0.0.0/0          100        None
      192.168.12.1   None        0
      64501           -
-----
Routes : 1
=====
    
```

The following shows the details of the received BGP-IPv4 route and the advertised BGP-IPv6 route in the base router on PE-2:

```
*A:PE-2# show router bgp routes 0.0.0.0/0 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       : 0.0.0.0/0
Nexthop       : 192.168.12.1
Path Id       : None
From          : 192.168.12.1
Res. Protocol : LOCAL                Res. Metric   : 0
Res. Nexthop  : 192.168.12.1
Local Pref.   : None                 Interface Name: int-PE-2-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: 192.0.2.1
Fwd Class     : None                 Priority       : None
Flags       : Used Valid Best Incomplete
Route Source  : External
AS-Path       : 64501
Route Tag     : 0
Neighbor-AS   : 64501
Orig Validation: NotFound
Source Class  : 0                     Dest Class    : 0
Add Paths Send: Default
RIB Priority   : Normal
Last Modified : 00h00m49s
-----
RIB Out Entries
-----
Network       : 0.0.0.0/0
Nexthop       : 192.168.12.2
Path Id       : None
To           : 192.168.12.1
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      : 0
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id: None                 Peer Router Id: 192.0.2.1
Origin      : Incomplete
AS-Path       : 64500
Route Tag     : 0
```

```
Neighbor-AS   : 64500
Orig Validation: NotFound
Source Class  : 0                               Dest Class   : 0
-----
Routes : 2
=====
```

The origin attribute can be modified by the following export policy that adds the large community 64496:1:1 and sets the MED value to 50 and the origin to IGP (so it will not be Incomplete anymore):

```
# on PE-1, PE-2:
configure
  router Base
    policy-options
      begin
        community "large1"
          members "64496:1:1"
      exit
      policy-statement "export-default"
        default-action accept
        community add "large1"
        origin igp
        bgp-med set 50
      exit
    exit
  commit
```

This 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 ipv6 export-policy "export-default"
      exit
    exit
  exit
  service
    vprn "VPRN 1"
      bgp
        send-default ipv4 ipv6 export-policy "export-default"
      exit
    exit
```

This export policy sets the origin to IGP instead of Incomplete. PE-2 receives the BGP-IPv4 default route with origin IGP and MED 50, as follows:

```
*A: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) As-Path	Path-Id	IGP Cost Label
u*>i	0.0.0.0/0	None	50
	192.168.12.1	None	0
	64501		-

Routes : 1			
=====			

```
*A: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 **disable-communities large** command is configured on PE-1 for the IPv4 neighbors in the base router and in VPRN 1, so no large community is sent to PE-2 for IPv4; only for IPv6. On PE-2, the details of the received default IPv6 route ::/0 in VPRN 1 are as follows:

```
*A: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
Aggregator AS : None          Interface Name : int-VPRN1-PE-2-PE-1
Atomic Aggr.  : Not Atomic   Aggregator    : None
AIGP Metric   : None         MED           : 50
Connector     : None         IGP Cost      : 10
Community     : 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 CLI in the current edition is based on SR OS Release 20.10.R1.

Overview

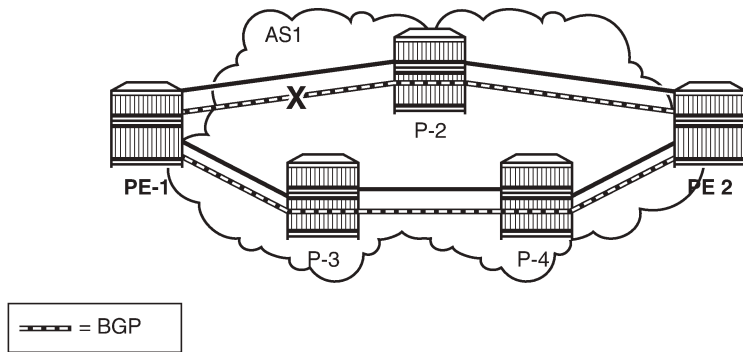
Border Gateway Protocol (BGP) is a key protocol for ISPs, supporting inter-Autonomous System (inter-AS) and intra-Autonomous System (intra-AS) applications with many address families. Additionally, ISPs need to maintain the service level agreements with their customers, even in case of network failures.

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.

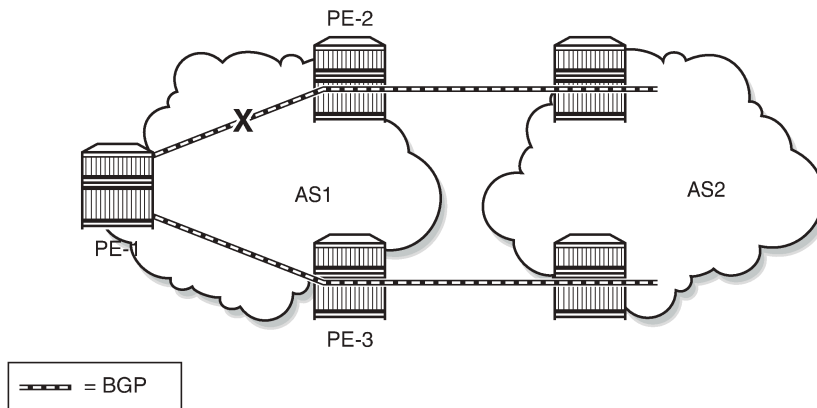
Within SR OS, two BGP FRR functions are supported: Core PIC and Edge PIC. Core PIC describes a scenario where a link or node on the path to the BGP next-hop fails, but the BGP next-hop remains reachable; see [Figure 36: Core PIC](#). Edge PIC describes a scenario where an edge node or edge link fails, which results in a change of the BGP next-hop; see [Figure 37: Edge PIC](#).

Figure 36: Core PIC



26255

Figure 37: 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

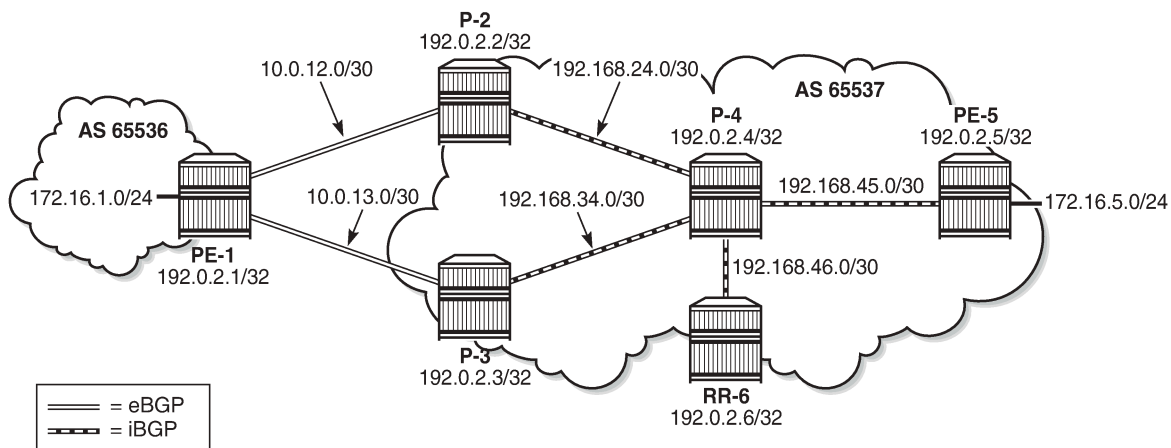
This chapter describes the use of BFD and MRAI for faster network convergence.

Configuration

The example topology used in this chapter is shown in [Figure 38: 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 to P-2 and P-3.
- P-2 changes the local preference to 150 for the route advertised to its route reflector RR-6.
- P-2 and P-3 advertise a BGP route for prefix 172.16.5.0/24 to PE-1.

Figure 38: 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
  bgp
    loop-detect discard-route
    advertise-inactive
    split-horizon
    advertise-external ipv4
    group "eBGP_AS65536"
      peer-as 65536
      neighbor 10.0.13.1
    exit
  exit
  group "iBGP_AS65537"
    next-hop-self
    peer-as 65537
    neighbor 192.0.2.6
  exit
  exit
  no shutdown
exit
```

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: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                               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
  bgp
    group "iBGP_AS65537"
      add-paths
        ipv4 send 2 receive
```

The BGP configuration on P-2 is as follows:

```
# on P-2:
configure
router
  bgp
    loop-detect discard-route
    advertise-inactive
    split-horizon
    group "eBGP_AS65536"
      import "Import_LP150"
      peer-as 65536
      neighbor 10.0.12.1
      exit
    exit
    group "iBGP_AS65537"
      next-hop-self
      peer-as 65537
      add-paths
        ipv4 send 2 receive
      exit
      neighbor 192.0.2.6
      exit
    exit
  no shutdown
exit
```

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
  bgp
    loop-detect discard-route
    split-horizon
    group "iBGP_AS65537"
      cluster 6.6.6.6
      advertise-inactive
      peer-as 65537
      add-paths
        ipv4 send 2 receive
      exit
    neighbor 192.0.2.2
    exit
    neighbor 192.0.2.3
    exit
    neighbor 192.0.2.4
    exit
    neighbor 192.0.2.5
    exit
  exit
  no shutdown
exit
  
```

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: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                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Path-Id    Label
-----
i     172.16.1.0/24                         150        None
      192.0.2.2                             1          0
      65536                                  -
-----
Routes : 1
=====
  
```

```
*A: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 the *BGP Add-Paths* chapter and the *BGP Multipath* chapter.

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
  bgp
    backup-path ipv4
```

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
  bgp
    loop-detect discard-route
    split-horizon
    backup-path ipv4
    group "iBGP_AS65537"
      peer-as 65537
      add-paths
        ipv4 send 2 receive
    exit
    neighbor 192.0.2.6
    exit
  exit
  no shutdown
exit
```

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: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 3 10
      65536 -
*i 172.16.1.0/24 100 None
   192.0.2.3 1 10
   65536 -
-----
Routes : 2
=====

```

The routing table then is as follows:

```

*A: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: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 3 10
      65536 -
ub*i 172.16.1.0/24 100 None
-----

```

```

192.0.2.3          1          10
65536              -
-----
Routes : 2
=====

```

Now the routing table is as follows. The "B" flag indicates that a BGP backup path is available.

```

*A: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: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:P-4# show router fib 1 172.16.1.0/24 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:P-4# show router fib 1 172.16.1.0/24 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:PE-5# traceroute no-dns 172.16.1.1 source 172.16.5.1
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
  interface "int-PE-1-P-2"
    address 10.0.12.1/30
    port 1/1/1
    bfd 100 receive 100 multiplier 3

```



```

        no shutdown
    exit
    interface "int-PE-1-P-3"
        address 10.0.13.1/30
        port 1/1/2
        bfd 100 receive 100 multiplier 3
        no shutdown
    exit
    bgp
        loop-detect discard-route
        bfd-enable
        split-horizon
        backup-path ipv4
        group "eBGP_AS65537"
            export "AS65537_172.16.1.0/24"
            peer-as 65537
            neighbor 10.0.12.2
            exit
            neighbor 10.0.13.2
            exit
        exit
        no shutdown
    exit

```

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
        interface "int-P-2-P-4"
            address 192.168.24.1/30
            port 1/1/1
            bfd 100 receive 100 multiplier 3
            no shutdown
        exit
        interface "int-P-2-PE-1"
            address 10.0.12.2/30
            port 1/1/2
            bfd 100 receive 100 multiplier 3
            no shutdown
        exit
        interface "system"
            address 192.0.2.2/32
            no shutdown
        exit

```

BFD is enabled for group eBGP_AS65536 only, at group level, as follows:

```

# on P-2:
configure
    router
        bgp
            loop-detect discard-route
            advertise-inactive
            split-horizon
            group "eBGP_AS65536"
                import "Import_LP150"
                peer-as 65536
                bfd-enable
                neighbor 10.0.12.1
                exit
            exit
            group "iBGP_AS65537"

```

```

        next-hop-self
        peer-as 65537
        add-paths
            ipv4 send 2 receive
        exit
        neighbor 192.0.2.6
        exit
    exit
exit

```

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
        isis 0
            area-id 49.0001
            interface "system"
                no shutdown
            exit
            interface "int-P-2-P-4"
                interface-type point-to-point
                bfd-enable ipv4
                no shutdown
            exit
        no shutdown
    exit

```

Faster convergence through MRAI

Adjusting the BGP MRAI also can help speed up network convergence, using the following command:

```

configure router bgp min-route-advertisement
- min-route-advertisement <seconds>
- no min-route-advertisement

<seconds>          : [1..255]

```

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
        shutdown

```

The path through the network is PE-5-P-4-P-3-PE-1, as follows:

```

*A:PE-5# traceroute no-dns 172.16.1.1 source 172.16.5.1
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: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            100        None
      192.0.2.3             1          10
      65536                  -
-----
Routes : 1
=====
```

The route table is as follows:

```
*A: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:P-4# show router fib 1 172.16.1.0/24 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:PE-3# tree flat detail | match install-backup-path
configure router policy-options policy-statement default-action install-backup-path
configure router policy-options policy-statement default-action no install-backup-path
configure router policy-options policy-statement entry action install-backup-path
configure router policy-options policy-statement entry action no install-backup-path
```

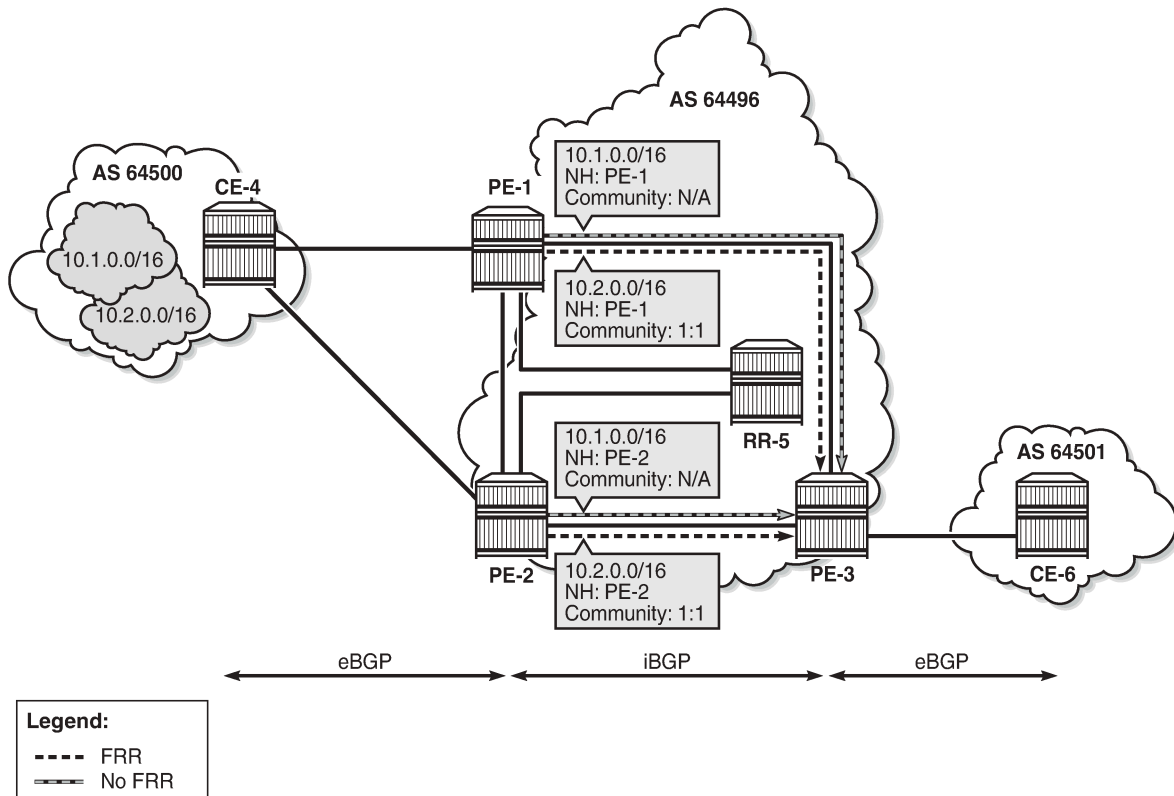
The **install-backup-path** command is effective when configured in BGP-import or VRF-import policies. In cases where this command is configured in an import policy applied in the global **bgp** context, the command applies to the following types of routes:

- 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 39: 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.

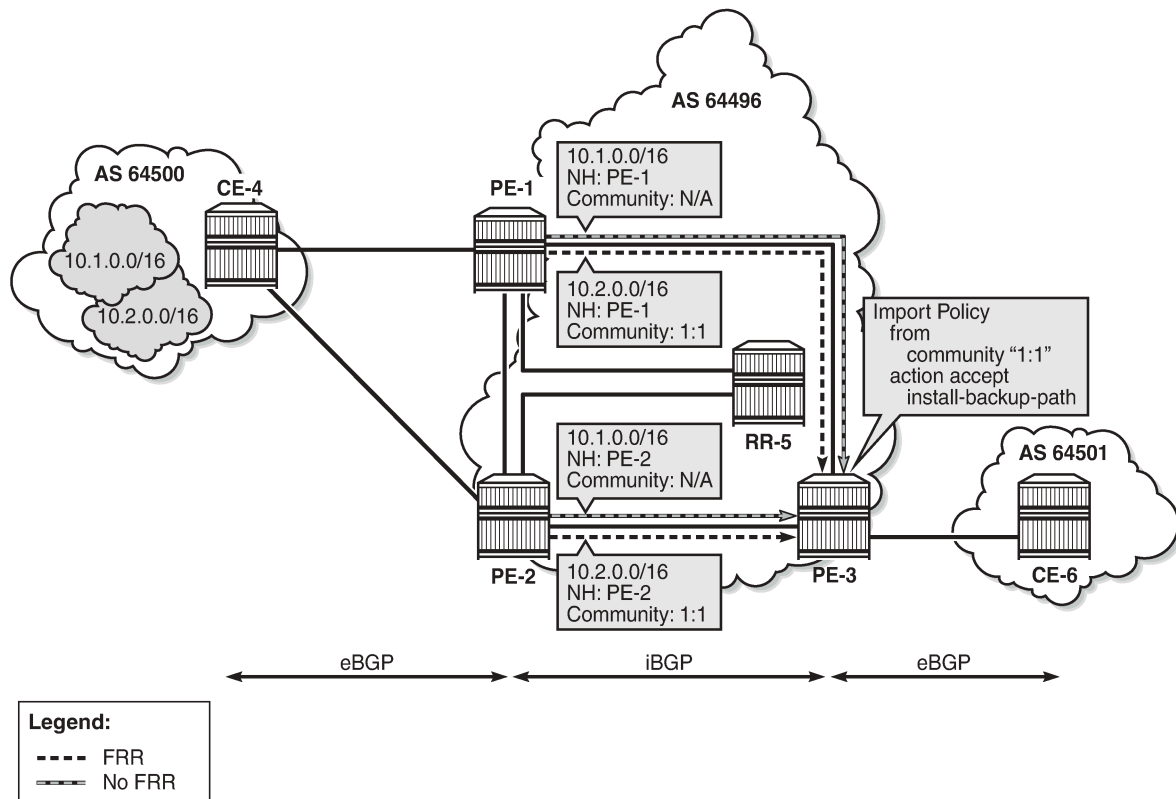
Figure 39: Community addition on PE-1 and PE-2



26770

Figure 40: 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 40: 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

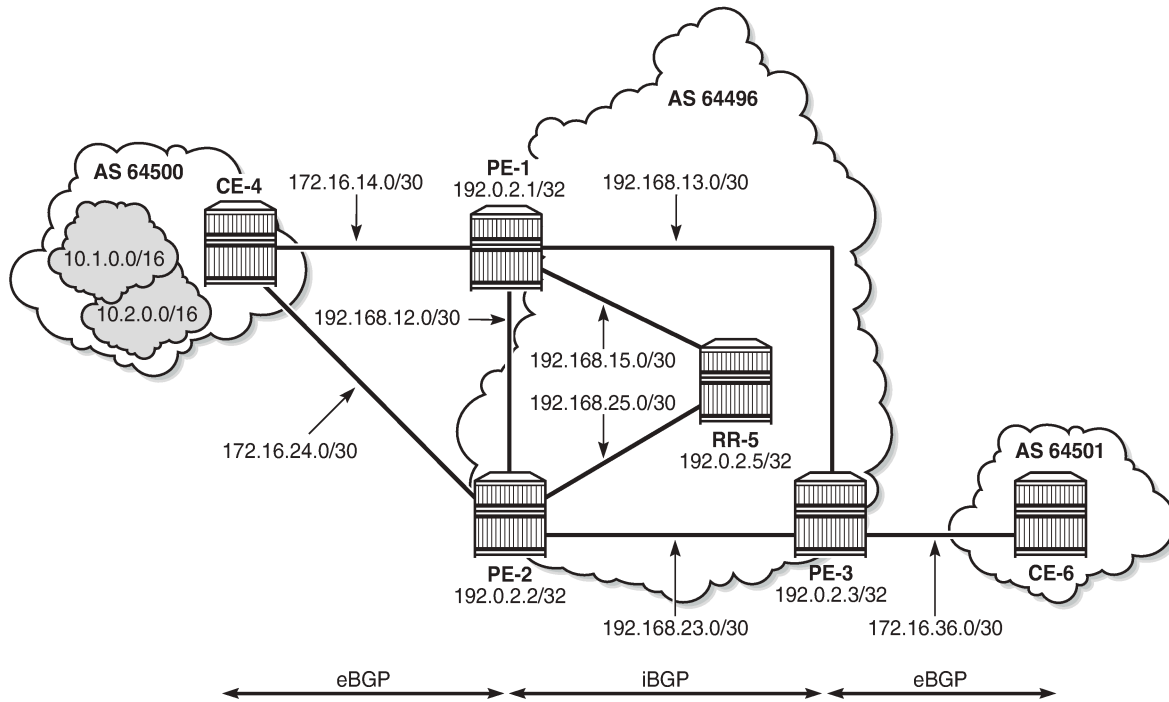
BGP FRR policy control feature for address family IPv4

Figure 41: Example topology - IPv4 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 41: 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
  router "Base"
    autonomous-system 64500
    policy-options
      begin
        prefix-list "10.1.0.0/16"
          prefix 10.1.0.0/16 longer
        exit
      prefix-list "10.2.0.0/16"
```



```

        prefix 10.2.0.0/16 longer
    exit
    policy-statement "export-bgp"
        entry 10
            from
                prefix-list "10.1.0.0/16"
            exit
            action accept
            exit
        exit
        entry 20
            from
                prefix-list "10.2.0.0/16"
            exit
            action accept
            exit
        exit
    exit
    commit
exit
bgp
    rapid-withdrawal
    split-horizon
    group "eBGP"
        export "export-bgp"
        peer-as 64496
        neighbor 172.16.14.1
        exit
        neighbor 172.16.24.1
        exit
    exit
    no shutdown
exit

```

CE-4 also has configured the following loopback interfaces:

```

# on CE-4:
configure
    router "Base"
        interface "int-loopback-1"
            address 10.1.1.1/16
            loopback
            no shutdown
        exit
        interface "int-loopback-2"
            address 10.2.1.1/16
            loopback
            no shutdown
        exit
    exit

```

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
            split-horizon
            group "eBGP"
                peer-as 64500

```

```

        neighbor 172.16.14.2
        exit
    exit
    group "iBGP"
        next-hop-self
        peer-as 64496
        add-paths
            ipv4 send 2 receive
        exit
        neighbor 192.0.2.5
        exit
    exit
    no shutdown
exit

```

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
        split-horizon
        group "iBGP"
            cluster 5.5.5.5
            peer-as 64496
            add-paths
                ipv4 send 2 receive
            exit
            neighbor 192.0.2.1
            exit
            neighbor 192.0.2.2
            exit
            neighbor 192.0.2.3
            exit
        exit
    no shutdown

```

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: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) As-Path	Path-Id	IGP Cost Label
u*>i	10.1.0.0/16 192.0.2.1 64500	100 1	None 10 -
*i	10.1.0.0/16 192.0.2.2 64500	100 9	None 10 -
u*>i	10.2.0.0/16 192.0.2.1 64500	100 4	None 10 -
*i	10.2.0.0/16 192.0.2.2 64500	100 10	None 10 -

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: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                               1          10
      64500                                    -
ub*i 10.1.0.0/16                               100        None
      192.0.2.2                               9          10
      64500                                    -
u*>i  10.2.0.0/16                               100        None
      192.0.2.1                               4          10
      64500                                    -
ub*i 10.2.0.0/16                               100        None
      192.0.2.2                               10         10
      64500                                    -
-----
Routes : 4
=====
```

PE-3 installs BGP routes from PE-2 as backup routes in its route table:

```
*A: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    00h01m18s 170
  192.168.13.1                                     10
10.1.0.0/16 (Backup)                            Remote BGP 00h01m18s 170
  192.168.23.1                                    10
10.2.0.0/16                                       Remote BGP    00h01m18s 170
  192.168.13.1                                     10
10.2.0.0/16 (Backup)                            Remote BGP 00h01m18s 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
      no backup-path
```

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:PE-3# show router bgp routes 10.2.0.0/16 detail | match "^Nexthop |Community" expression
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
  router "Base"
```

```

policy-options
  begin
  prefix-list "10.2.0.0/16"
    prefix 10.2.0.0/16 longer
  exit
  community "1:1"
    members "1:1"
  exit
  policy-statement "add-bgp-community"
    entry 10
      from
        prefix-list "10.2.0.0/16"
      exit
      action accept
        community add "1:1"
      exit
    exit
  exit
  commit
  
```

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 "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:PE-3# show router bgp routes 10.2.0.0/16 detail | match "^Nexthop |Community" expression
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
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
  router "Base"
    policy-options
      begin
      community "1:1"
        members "1:1"
      exit
      policy-statement "policy-bgp-frr-import"
        entry 10
          from
            community "1:1"
          exit
          action accept
            install-backup-path
          exit
        exit
      exit
  
```

```

        exit
    exit
commit

```

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-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: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                1          10
      64500                      -
*i    10.1.0.0/16                100        None
      192.0.2.2                9          10
      64500                      -
u*>i  10.2.0.0/16                100        None
      192.0.2.1                4          10
      64500                      -
ub*i 10.2.0.0/16                100        None
      192.0.2.2                10         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: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

```

```

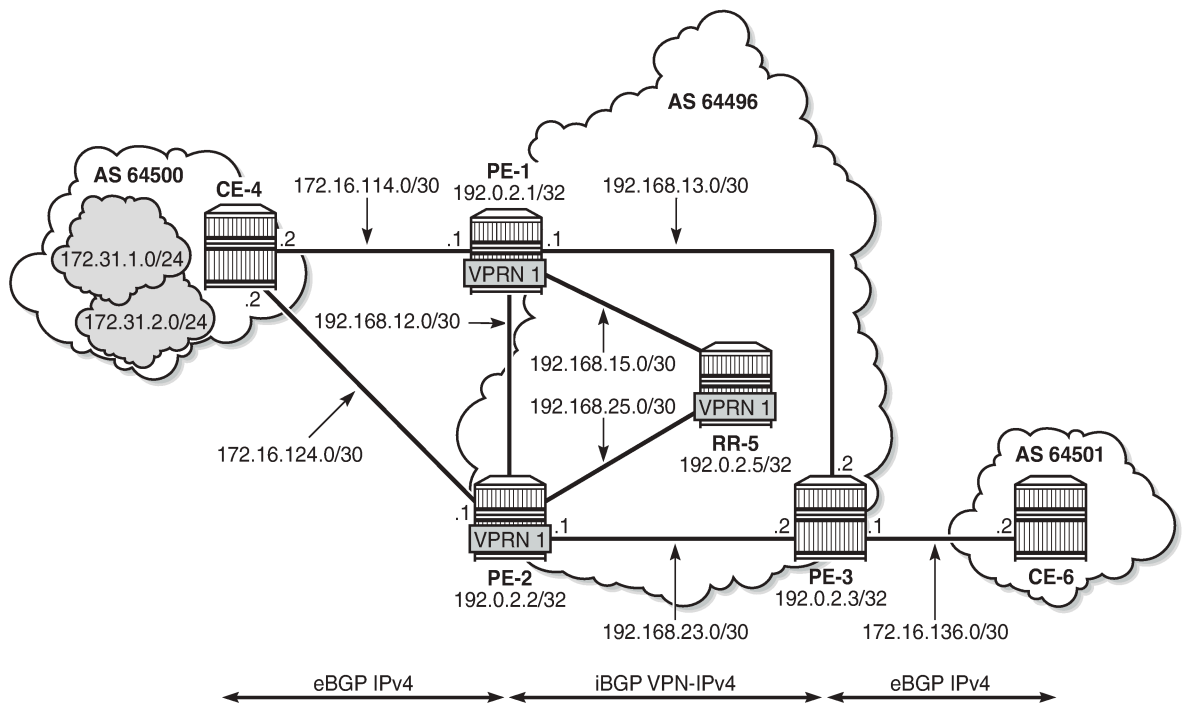
-----
10.1.0.0/16                               Remote BGP      00h02m35s 170
    192.168.13.1                            10
10.2.0.0/16                               Remote BGP      00h01m03s 170
    192.168.13.1                            10
10.2.0.0/16 (Backup)                   Remote BGP      00h01m03s 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 42: 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 42: 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 1 name "VPRN 1" customer 1 create
  autonomous-system 64496
  route-distinguisher 64496:1
  auto-bind-tunnel
    resolution any
  exit
  vrf-target target:64496:1
  interface "int-PE-1-CE-4-VPRN1" create
    address 172.16.114.1/30
    sap 1/1/cl/2:1 create
    exit
  exit
  bgp
    split-horizon
    group "eBGP-1"
      peer-as 64500
      neighbor 172.16.114.2
    exit
  no shutdown
exit
no shutdown

```

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
  service
    vprn 1 name "VPRN 1" customer 1 create
      autonomous-system 64500
      route-distinguisher 64500:1
      interface "int-CE-4-PE-1-VPRN1" create
        address 172.16.114.2/30
        sap 1/1/cl/1:1 create
        exit
      exit
      interface "int-CE-4-PE-2-VPRN1" create
        address 172.16.124.2/30
        sap 1/1/cl/2:1 create
        exit
      exit
      interface "loopback1-VPRN1" create
        address 172.31.1.1/24
        loopback
      exit
      interface "loopback2-VPRN1" create
        address 172.31.2.1/24
        loopback
      exit
    bgp
      split-horizon
      group "eBGP-1"
        export "export-VPRN1"
        peer-as 64496
        neighbor 172.16.114.1
        exit
        neighbor 172.16.124.1
        exit
      exit
    no shutdown
  exit
no shutdown

```


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
router "Base"
  policy-options
  begin
  prefix-list "172.31.0.0/16"
  prefix 172.31.0.0/16 longer
  exit
  policy-statement "export-VPRN1"
  entry 10
  from
  prefix-list "172.31.0.0/16"
  exit
  action accept
  exit
  exit
exit
commit
```

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 vpn-ipv4
```

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
```

The BGP configuration on PE-1 is as follows:

```
# on PE-1:
configure
router "Base"
  bgp
  rapid-withdrawal
  split-horizon
  group "eBGP"
  import "add-bgp-community"
  peer-as 64500
  neighbor 172.16.14.2
  exit
exit
group "iBGP"
  family ipv4 vpn-ipv4
```

```

next-hop-self
peer-as 64496
add-paths
  ipv4 send 2 receive
  vpn-ipv4 send 2 receive
exit
neighbor 192.0.2.5
exit
exit
no shutdown

```

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: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                             11       10
      64500                                  524284
*>i 64496:1:172.31.1.0/24                   100      None
      192.0.2.2                             12       10
      64500                                  524284
u*>i 64496:1:172.31.2.0/24                   100      None
      192.0.2.1                             13       10
      64500                                  524284
*>i 64496:1:172.31.2.0/24                   100      None
      192.0.2.2                             14       10
      64500                                  524284
-----
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
  router "Base"
    policy-options
      begin
        prefix-list "172.31.2.0/24"
          prefix 172.31.2.0/24 longer
        exit
        community "1:1"

```

```

        members "1:1"
    exit
    community "target:64496:1"
        members "target:64496:1"
    exit
    policy-statement "policy-export-VPRN1"
        entry 10
            from
                prefix-list "172.31.2.0/24"
            exit
            action accept
                community add "1:1" "target:64496:1"
            exit
        exit
        entry 20
            from
            exit
            action accept
                community add "target:64496:1"
            exit
        exit
    exit
    exit
    commit

```

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"
            vrf-export "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:PE-3# show router bgp routes 172.31.1.0/24 vpn-ipv4 hunt | match "Comm"
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:PE-3# show router bgp routes 172.31.2.0/24 vpn-ipv4 hunt | match "Comm"
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

```

PE-3 now has a BGP backup path only for prefix 172.31.2.0/24, as indicated by the "b" (backup) flag:

```

*A: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                Path-Id    Label
-----
u*>i  64496:1:172.31.1.0/24      100        None
      192.0.2.1              11         10
      64500                   524284
*>i   64496:1:172.31.1.0/24      100        None
      192.0.2.2              12         10
      64500                   524284
u*>i  64496:1:172.31.2.0/24      100        None
      192.0.2.1              13         10
      64500                   524284
ub*>i 64496:1:172.31.2.0/24      100        None
      192.0.2.2              14         10
      64500                   524284
-----
Routes : 4
=====

```

PE-3 has installed a backup route only for prefix 172.31.2.0/24 in its VPRN route table:

```

*A: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 00h01m08s 170
      192.0.2.1 (tunneled)  10
172.31.2.0/24              Remote BGP VPN 00h01m08s 170
      192.0.2.1 (tunneled)  10
172.31.2.0/24 (Backup)    Remote BGP VPN 00h01m08s 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
    no vpn-apply-import
```

PE-3 removes the backup path for prefix 172.31.2.0/24:

```
*A: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 00h00m57s 170
  192.0.2.1 (tunneled)                            10
172.31.2.0/24                                     Remote BGP VPN 00h00m57s 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
router "Base"
  policy-options
  begin
  community "1:1"
    members "1:1"
  exit
  community "target:64496:1"
    members "target:64496:1"
  exit
  policy-statement "policy-import-VPRN1"
  entry 10
    from
      community expression "[target:64496:1] AND [1:1]"
  exit
  action accept
    install-backup-path
  exit
  exit
  default-action accept
  exit
  exit
  commit
```

The policy is applied as a VRF-import policy in VPRN 1 on PE-3:

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

```
vprn "VPRN 1"
  vrf-import "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: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 00h00m55s 170
  192.0.2.1 (tunneled)                            10
172.31.2.0/24                                     Remote BGP VPN 00h00m55s 170
  192.0.2.1 (tunneled)                            10
172.31.2.0/24 (Backup)                          Remote BGP VPN 00h00m55s 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, *Dissemination of Flow Specification Rules*, 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 *Unicast Guide*, tables *Subcomponents of FlowSpec IPv4 and FlowSpec-VPN IPv4 NLRI* and *Subcomponents of FlowSpec IPv6 and FlowSpec-VPN IPv6 NLRI* list the subcomponent types that are 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 *Unicast 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.

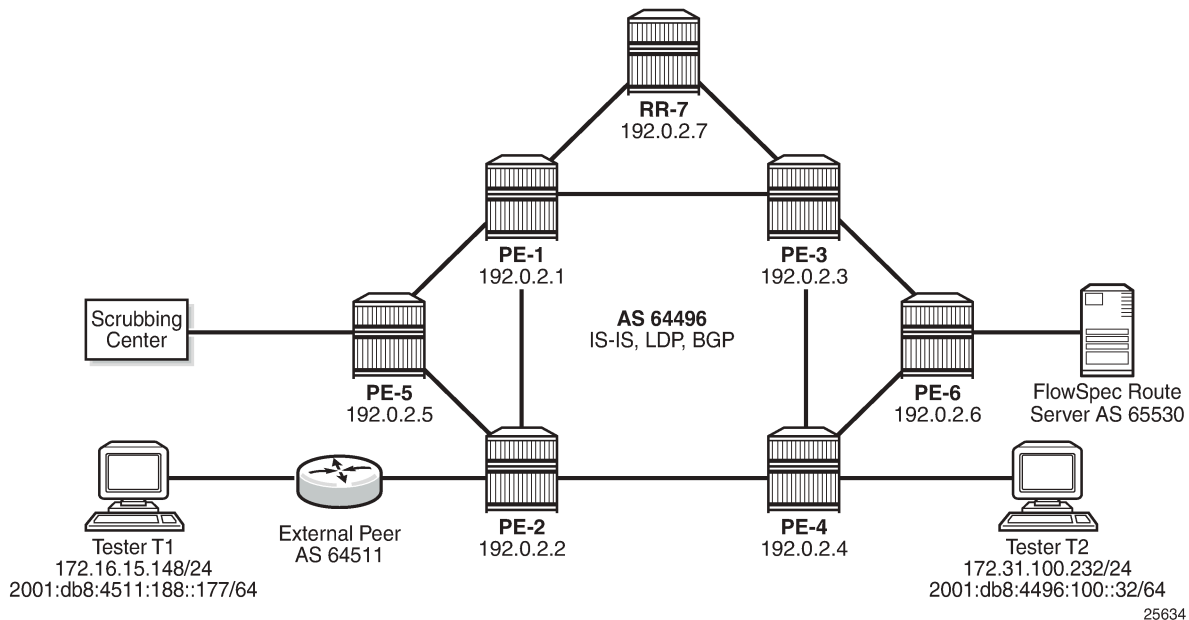
These filters should be instantiated as close as possible to the source of the attack traffic, even if that means applying filters within another operator's domain. This means that FlowSpec routes must be exchanged between ASs, requiring a trust relationship between the ASs, and a method for validating 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 43: 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 43: 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.

```
# on PE-1:
configure
router
  bgp
    loop-detect discard-route
    advertise-inactive
    split-horizon
    group "IBGP"
      family ipv4 ipv6 vpn-ipv4 vpn-ipv6 flow-ipv4 flow-ipv6
                                         label-ipv4 label-ipv6
    peer-as 64496
    neighbor 192.0.2.7
    exit
  exit
no shutdown
exit
```

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.

```
# on PE-2:
configure
service
  ies 10 name "FlowSpec-testshow " customer 1 create
  interface "to-AS64511" create
    address 192.168.2.1/30
    ipv6
      address 2001:db8:2c0d:2121::2/127
    exit
  sap 1/1/4 create
    ingress
      filter ip 104
      filter-ipv6 106
    exit
  exit
exit
no shutdown
exit
```

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.

```
# on PE-2:
```

```
configure
  filter
    ip-filter 104 create
      default-action forward
      embed-filter flowspec router "Base" offset 10000
    exit
  ipv6-filter 106 create
    default-action forward
    embed-filter flowspec router "Base" offset 10000
  exit
```

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: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: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: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
Fragment       : Off
Sampling       : Off
IP-Option      : 0/0
Tcp-flag       : (Not Specified)
Option-pres    : Off
Egress PBR     : Disabled
Primary Action  : Drop
Ing. Matches   : 0 pkts
Egr. Matches   : 0 pkts
ICMP Code      : Undefined
Src Route Opt  : Off
Int. Sampling  : On
Multiple Option: Off
```

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.

```
# on PE-2:
configure
  service
    vprn 2 name "FlowSpec-OffRamp-VRF" customer 1 create
      description "FlowSpec-OffRamp-VRF"
      bgp-ipvpn
        mpls
          auto-bind-tunnel
          resolution any
    exit
```

```

        route-distinguisher 64496:2
        vrf-target target:64496:2
        no shutdown
    exit
    exit
    no shutdown
exit

```

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 IPv6 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.

```

# on PE-5:
configure
  service
    vprn 2 name "FlowSpec-OffRamp-VRF" customer 1 create
    interface "OffRamp-to-Scrubbing-Center" create
      address 192.168.2.5/30
      ipv6
        address 2001:db8:1b0c:2121::4/127
      exit
      sap 1/2/1 create
      exit
    exit
    static-route-entry 0.0.0.0/0
      next-hop 192.168.2.6
      no shutdown
    exit
    exit
    static-route-entry ::/0
      next-hop 2001:db8:1b0c:2121::5
      no shutdown
    exit
    exit
    bgp-ipvpn
      mpls
        auto-bind-tunnel
        resolution any
      exit
      route-distinguisher 64496:2
      vrf-export "vrf2-export"
      no shutdown
    exit
  exit
  no shutdown
exit

```

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.

```

# on PE-5:
configure
  service
    ies 3 name "FlowSpec-OnRamp-IES" customer 1 create
    interface "OnRamp" create
      address 192.168.2.9/30
      ipv6
        address 2001:db8:1b0c:2121::6/127
      exit

```

```

        sap 1/2/2 create
        exit
    exit
    no shutdown
exit

```

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: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
Fragment         : Off
Sampling         : Off
IP-Option        : 0/0
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_1l_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:PE-2# show filter ip 104

=====
IP Filter
=====
Filter Id       : 104
Scope          : Template
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: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
Auth Hdr         : Off
Flow-label       : n/a
Egress PBR      : Disabled
Primary Action   : Drop
Ing. Matches    : 0 pkts
Egr. Matches    : 0 pkts
Routing Type0   : Off
ESP header      : Off
Flow-label Mask : n/a
-----
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:PE-2# show filter ipv6 106 detail
=====
IPv6 Filter
=====
Filter Id       : 106
Scope          : Template
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: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
Auth Hdr      : Off                Routing Type0  : Off
Flow-label    : n/a                ESP header     : Off
Egress PBR    : Disabled           Flow-label Mask: n/a
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: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 : Forward (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: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: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 FlowSpec Route Validation

This chapter provides information about BGP FlowSpec Route Validation.

Topics in this chapter include:

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

Applicability

The information and configuration in this chapter are based on SR OS Release 15.0.R7. This chapter describes the BGP FlowSpec route validation as implemented in SR OS Release 15.0.R1, and later.

Overview

BGP FlowSpec refers to the use of BGP to distribute traffic flow specifications for IPv4 or IPv6 routes throughout a network. Flow specifications provide a means to quickly mitigate Distributed Denial of Service (DDoS) attacks. The BGP FlowSpec standard RFC 5575 defines a method to define and advertise flow filters to upstream BGP peers via BGP Network Layer Reachability Information (NLRI). See the 7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide for the complete list of matching criteria (subcomponent names), such as destination prefix, source prefix, IP protocol, destination port, source port, and so on. The 7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide also lists the FlowSpec actions, such as redirect, rate limit, and so on.

BGP flow specifications might be manipulated and sent with malicious intentions. By default, all flow specifications received from iBGP or eBGP peers are accepted with optional validation. In SR OS Releases prior to 15.0.R1, the validity was checked only at the time when a FlowSpec route was received from the peer. In SR OS Release 15.0.R1, and later, the FlowSpec routes that are in the routing information base (RIB) can become invalid at a later time, depending on the state of the unicast routes. *Draft-ietf-idr-bgp-FlowSpec-oid-03* describes validation procedures for BGP FlowSpec routes in specific route controller, route reflector, and route server scenarios. These recommendations, in combination with the original validation rules mentioned in RFC 5575, are all supported in SR OS Release 15.0.R1, and later. The BGP FlowSpec route validation rules are as follows.

- Rule 1: Flowspec routes originated in the same Autonomous System (AS) as the receiving BGP speaker are always considered valid. This is the case when either of the following applies:
 - The AS_PATH and AS4_PATH attributes of the BGP FlowSpec route are empty.
 - The AS_PATH and AS4_PATH attributes of the BGP FlowSpec route do not contain AS_SET and AS_SEQUENCE segments.
- Rule 2: If Rule 1 does not apply, FlowSpec routes originated outside the local AS without a destination prefix subcomponent are always considered valid.

- Rule 3: If Rule 1 does not apply, FlowSpec routes originated outside the local AS with a destination prefix subcomponent are only considered valid if all the following is true:
 - The neighbor AS (the last non-confederation AS in its AS_PATH attribute) of the BGP FlowSpec route matches the neighbor AS of the unicast IP route that is the best match of the destination prefix.
 - The neighbor AS of the BGP FlowSpec route matches the neighbor AS of all unicast IP routes that are longer matches of the destination prefix.
 - The best match unicast IP route and all longer match unicast IP routes must be BGP routes, so no static or IGP routes.

BGP FlowSpec route validation in the base router is enabled with the following command.

```
configure router bgp flowspec validate-dest-prefix
```

BGP FlowSpec route validation in a VPRN is enabled as follows.

```
configure service vprn <service-id> bgp flowspec validate-dest-prefix
```

When validate-dest-prefix is enabled, the validation checks must be repeated every time there is a change to the best route or any longer match route of the destination prefix.

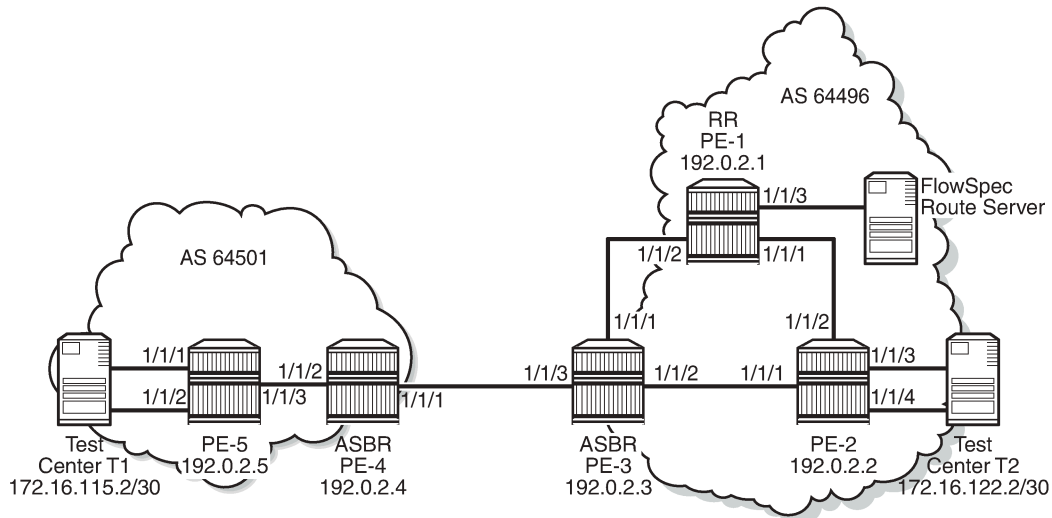
Configuration

In this section, BGP FlowSpec route validation for IPv4 routes in the base router is shown. The action will set the rate to zero, so the matching traffic is dropped. The following use cases will be shown:

- iBGP FlowSpec routes are valid when the AS_PATH attribute is empty. (Rule 1)
- eBGP FlowSpec routes are valid if the best match for the destination prefix is a BGP route toward the neighbor AS from which the BGP FlowSpec route was received (and all longer match unicast IP routes are also toward that AS). (Rule 3)
- eBGP FlowSpec routes are invalid if the best match for the destination prefix is not toward the AS from which the BGP FlowSpec route was received or when the route to the destination prefix is a static or an IGP route instead of a BGP route. (Rule 3)
- eBGP FlowSpec routes without destination prefix subcomponent are valid. (Rule 2)

[Figure 44: Example Topology with FlowSpec Route Server in AS 64496](#) shows the example topology with a FlowSpec route server in AS 64496 that will advertise iBGP FlowSpec routes to PE-1. Afterward, PE-1 will forward the valid FlowSpec routes to its BGP peers, and so on. Test center T1 in AS 64501 will generate traffic toward test center T2 in AS 64496. This traffic may be filtered by PE-5 when it receives a valid FlowSpec route with the correct matching criteria.

Figure 44: Example Topology with FlowSpec Route Server in AS 64496



27508

The initial configuration in the PEs is as follows.

- Cards, MDAs, ports
- Router interfaces
- IGP routing protocol within each AS, but not between the autonomous system border routers (ASBRs) PE-3 and PE-4. It is possible to have OSPF in one AS and IS-IS in the other.

PE-1 is the route reflector (RR) in AS 64496 with clients PE-2 and PE-3. BGP is enabled for the IPv4 and flow-IPv4 address families between the PEs and between PE-1 and the FlowSpec route server. Initially, the FlowSpec route server is in AS 64496, but that will change in a later scenario. The BGP configuration on RR PE-1 is as follows.

```
configure
router
  bgp
    split-horizon
    group "FlowSpec"
      family ipv4 flow-ipv4
      peer-as 64496
      neighbor 192.168.11.2
    exit
  exit
  group "iBGP"
    family ipv4 flow-ipv4
    cluster 192.0.2.1
    peer-as 64496
    advertise-inactive
    neighbor 192.0.2.2
  exit
  neighbor 192.0.2.3
  exit
  exit
  exit
```

The BGP configuration on PE-2 includes export policies for the system address 192.0.2.2/32 and the subnet toward the test center T2, 172.16.122.0/30, as follows. The configuration on PE-5 is similar, with export policies for the system address and for subnet 172.16.115.0/30.

```
configure
router
  policy-options
  begin
  prefix-list "T2"
    prefix 172.16.122.0/28 longer
  exit
  prefix-list "sys"
    prefix 192.0.2.0/29 longer
  exit
  policy-statement "export-T2"
    entry 10
      from
        protocol direct
        prefix-list "T2"
      exit
      action accept
      exit
    exit
  exit
  policy-statement "export-sys"
    entry 10
      from
        protocol direct
        prefix-list "sys"
      exit
      action accept
      exit
    exit
  exit
  commit
exit
bgp
  split-horizon
  group "iBGP"
    family ipv4 flow-ipv4
    export "export-sys" "export-T2"
    peer-as 64496
    neighbor 192.0.2.1
  exit
exit
```

On ASBR PE-3, the BGP configuration includes an iBGP group and an eBGP group. The BGP IPv4 routes for prefixes 192.0.2.2/32 and 172.16.122.0/30 are inactive within AS 64496, and the ASBR will advertise these inactive routes to its eBGP peer PE-4. The BGP configuration on PE-3 is as follows. The configuration is similar on PE-4.

```
configure
router
  bgp
    split-horizon
    group "eBGP"
      family ipv4 flow-ipv4
      peer-as 64501
      neighbor 192.168.34.2
        advertise-inactive
    exit
  exit
```

```

    group "iBGP"
      family ipv4 flow-ipv4
      next-hop-self
      peer-as 64496
      neighbor 192.0.2.1
        advertise-inactive
      exit
    exit
  exit

```

PE-2 and PE-5 both advertise two BGP IPv4 routes: one for the system address and another for the subnet toward the test center. These BGP routes will not be used within the local AS, but they will be advertised by the ASBRs to the peer AS, where these BGP routes will be used. The BGP IPv4 routes on ASBR PE-4 are as follows.

```

*A:PE-4# show router bgp routes
=====
BGP Router ID:192.0.2.4      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     Label
      As-Path
-----
*i   172.16.115.0/30                          100        None
      192.0.2.5
      No As-Path
u*>i 172.16.122.0/30                          None       None
      192.168.34.1
      64496
u*>i 192.0.2.2/32                              None       None
      192.168.34.1
      64496
*i   192.0.2.5/32                              100        None
      192.0.2.5
      No As-Path
-----
Routes : 4

```

The BGP IPv4 routes on PE-5 are as follows.

```

*A:PE-5# show router bgp routes
=====
BGP Router ID:192.0.2.5      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     Label
      As-Path
-----

```

```

As-Path
-----
u*>i 172.16.122.0/30          100    None
      192.0.2.4              None    -
      64496
u*>i 192.0.2.2/32           100    None
      192.0.2.4              None    -
      64496
-----
Routes : 2

```

No flow specifications have been received and no traffic will be filtered. When traffic is generated by T1 with IP destination address (DA) 172.16.122.2 and IP source address (SA) 172.16.115.2, it is forwarded to T2.

Default Treatment of FlowSpec Routes

The FlowSpec route server announces a FlowSpec IPv4 route to PE-1 with destination prefix 172.16.122.2/30, source prefix 172.16.115.2/30, destination port 4191, source port greater than 1024 as matching criteria, and rate limit 0 kbps (drop) as action. By default, there is no validation check for FlowSpec routes. All FlowSpec routes are considered valid and used, even if no BGP route exists to the destination prefix. All FlowSpec routes are advertised to all PEs, within the AS and to neighbor ASs. On all PEs, the FlowSpec route status codes are valid, best, and used. For example, on PE5:

```

*A:PE-5# show router bgp routes flow-ipv4
=====
BGP Router ID:192.0.2.5      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 FLOW IPV4 Routes
=====
Flag  Network      Nexthop      LocalPref    MED
As-Path
-----
u*>i  --           0.0.0.0      100          None
      64496

Community Action: rate-limit: 0 kbps
NLRI Subcomponents:
Dest Pref : 172.16.122.2/30
Src Pref  : 172.16.115.2/30
Ip Proto  : [ == 6 ]
Dest Port : [ == 4191 ]
Src Port  : [ >1024 ]
-----
Routes : 1

```

On all PEs, an embedded IPv4 filter "fSpec-0" will be auto-created for the base router, as follows.

```

*A:PE-5# show filter ip filter-type flowspec
=====
Flowspec IP Filters                                     Total:    1
=====

```

```

Filter-Id  Scope  Applied Description
-----
fSpec-0   Embedded N/A    IPv4 BGP FlowSpec filter for the Base router
=====
*A:PE-5#

```

The details for this embedded filter are retrieved as follows.

```

*A:PE-5# show filter ip "fSpec-0"

=====
IP Filter
=====
Filter Id       : fSpec-0
Scope          : Embedded
Entries        : 1 (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       : 172.16.115.2/30
Src. Port     : gt 1024
Dest. IP      : 172.16.122.2/30
Dest. Port    : eq 4191
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-syn      : Off
TCP-ack      : Off
Option-pres  : Off
Egress PBR   : Disabled
Primary Action : Drop
Ing. Matches : 0 pkts
Egr. Matches : 0 pkts
=====
*A:PE-5#

```

This embedded filter "fSpec-0" is created on all PEs, and no traffic is filtered when no IPv4 filter is configured referencing this embedded filter. For this reason, PE-5 has the following IPv4 filter configured and applied on the ingress direction of interface "int-PE-5-T1". The default action is forward; only traffic matching the embedded FlowSpec filter is dropped (rate limit 0 kbps).

```

configure
  filter
    ip-filter 1 create
      default-action forward
      embed-filter flowspec router "Base"
    exit
  info
  exit
  router
    interface "int-PE-5-T1"
      ingress
        filter ip 1
      exit

```



```
exit
```

The following command on PE-5 shows that IPv4 filter 1 contains embedded filter "fSpec-0".

```
*A:PE-5# show filter ip 1 embedded

=====
IP Filter embedding
=====
In      Offset  From          Inserted      Status
-----
1       0       fSpec-0       1/1          OK
=====
*A:PE-5#
```

Test center T1 generates TCP traffic with IP DA 172.16.122.2, IP SA 172.16.115.2, destination port 4191, and source port 1025. This traffic matches the FlowSpec criteria and will be discarded, because the FlowSpec action is to limit the rate to 0 kbps. The following monitor command on PE-5 shows that the traffic incoming at port 1/1/1 (interface int-PE-5-T1) is dropped instead of being forwarded to port 1/1/3 toward PE-3.

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

=====
Monitor statistics for Ports
=====
                                Input          Output
-----
---snip---

At time t = 3 sec (Mode: Rate)
-----
Port 1/1/1
-----
Octets          544683          27
Packets         4255            0
---snip---

Port 1/1/3
-----
Octets           30             30
Packets          0              0
---snip---
```

The following command shows the IPv4 filter 1 with the filter match criteria. In this example, 67612 packets have matched the filter at the ingress and are dropped, because the primary action in the embedded FlowSpec filter is drop.

```
*A:PE-5# show filter ip 1

=====
IP Filter
=====
Filter Id       : 1                Applied       : Yes
Scope           : Template          Def. Action   : Forward
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)
Description       : (Not Specified)
-----
Filter Match Criteria : IP
-----
Entry             : 256
Origin           : Inserted by embedded filter fSpec-0 entry 256
Description       : (Not Specified)
Log Id           : n/a
Src. IP          : 172.16.115.2/30
Src. Port        : gt 1024
Dest. IP         : 172.16.122.2/30
Dest. Port       : eq 4191
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-syn         : Off
TCP-ack         : Off
Option-pres     : Off
Egress PBR      : Disabled
Primary Action : Drop
Ing. Matches  : 67612 pkts (8654336 bytes)
Egr. Matches    : 0 pkts

=====
*A:PE-5#

```

FlowSpec Route Validation

On all PEs, FlowSpec route validation on the destination prefix is enabled within the base router context, as follows.

```
configure router bgp flowspec validate-dest-prefix
```

iBGP FlowSpec Routes

The FlowSpec route server is in AS 64496, so the AS_PATH attribute will be empty when it sends a FlowSpec IPv4 route to iBGP peer PE-1. For this reason, the FlowSpec route is considered valid. The following FlowSpec IPv4 route is received on PE-1 and the status codes are valid, best, and used:

```

*A:PE-1# show router bgp routes flow-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 FLOW IPV4 Routes
=====
Flag  Network      Nexthop      LocalPref    MED
     As-Path
-----

```

```

u*>i  --                0.0.0.0                100                None
No As-Path

Community Action: rate-limit: 0 kbps
NLRI Subcomponents:
Dest Pref : 172.16.122.2/30
Src Pref  : 172.16.115.2/30
Ip Proto  : [ == 6 ]
Dest Port : [ == 4191 ]
Src Port  : [ >1024 ]
-----
Routes : 1
    
```

PE-1 will forward this valid route to its iBGP peers PE-2 and PE-3, which will also consider this FlowSpec route as valid.

eBGP FlowSpec Routes

Valid eBGP FlowSpec Routes with Destination Prefix

The FlowSpec IPv4 route is not only forwarded to the iBGP peers in AS 64496, but also by PE-3 in AS 64496 to its eBGP peer PE-4 in AS 64501. The eBGP FlowSpec route has a destination prefix subcomponent and it is valid on PE-4 because its neighbor AS (64496) matches the neighbor AS of the unicast IPv4 route that is the best match of destination prefix 172.16.122.2/30. It also matches the neighbor AS of all unicast IPv4 routes that are longer matches of the destination prefix. Also, the best match unicast IPv4 route is a BGP route. The following shows the FlowSpec IPv4 route received by PE-4 as valid, best, and used:

```

*A:PE-4# show router bgp routes flow-ipv4
=====
BGP Router ID:192.0.2.4      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 FLOW IPV4 Routes
=====
Flag  Network          Nexthop          LocalPref        MED
As-Path
-----
u*>i  --                0.0.0.0          n/a              None
64496

Community Action: rate-limit: 0 kbps
NLRI Subcomponents:
Dest Pref : 172.16.122.2/30
Src Pref  : 172.16.115.2/30
Ip Proto  : [ == 6 ]
Dest Port : [ == 4191 ]
Src Port  : [ >1024 ]
-----
Routes : 1
    
```

The following route table entry shows that the best match unicast IPv4 route for destination prefix 172.16.122.0/30 is a BGP route:

```
*A:PE-4# show router route-table 172.16.122.0/30

=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]                Metric
-----
172.16.122.0/30                    Remote BGP   00h19m55s  170
  192.168.34.1                       0
-----
No. of Routes: 1
```

The BGP IPv4 route for destination prefix 172.16.122.0/30 is as follows. The AS_PATH attribute only contains AS 64496, which is the AS where the FlowSpec IPv4 route originated.

```
*A:PE-4# show router bgp routes 172.16.122.0/30

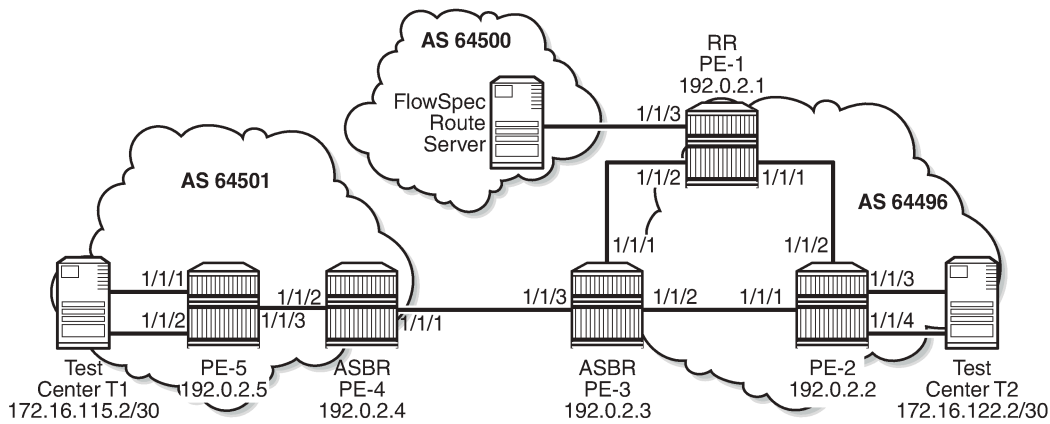
=====
BGP Router ID:192.0.2.4      AS:64501      Local AS:64501
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
Flag Network                LocalPref  MED
  Nexthop (Router)          Path-Id    Label
  As-Path
-----
u*>i 172.16.122.0/30          None       None
      192.168.34.1          None       -
      64496
-----
Routes : 1
```

PE-4 will then forward the valid FlowSpec IPv4 route to its iBGP peer PE-5, which will accept the FlowSpec IPv4 route as valid. As a result, an embedded filter "fSpec-0" will be auto-created. When test center T1 sends a traffic flow to T2 with matching criteria, the traffic will be dropped at the ingress port of interface "int-PE-5-T1" on PE-5.

Invalid eBGP FlowSpec Routes with Destination Prefix

[Figure 45: Topology with FlowSpec Route Server in AS 64500](#) shows an example topology with the FlowSpec route server in AS 64500 and the other nodes in the same ASs as before.

Figure 45: Topology with FlowSpec Route Server in AS 64500



27509

The BGP configuration on RR PE-1 has been modified with a different peer AS in group "FlowSpec", as follows. FlowSpec validation remains enabled on all routers, so that part of the configuration need not be modified.

```
configure
router
  bgp
    split-horizon
    group "FlowSpec"
      family ipv4 flow-ipv4
      peer-as 64500
      neighbor 192.168.11.2
    exit
  group "iBGP"
    family ipv4 flow-ipv4
    cluster 192.0.2.1
    peer-as 64496
    advertise-inactive
    neighbor 192.0.2.2
  exit
  neighbor 192.0.2.3
  exit
  exit
exit
```

The FlowSpec route server advertises FlowSpec IPv4 routes to eBGP peer PE-1. When the FlowSpec route server advertises the preceding FlowSpec IPv4 route with IP DA 172.16.122.2/30, the receiving eBGP peer PE-1 will consider the FlowSpec IPv4 route invalid, because the FlowSpec IPv4 route was received from AS 64500 whereas IP prefix 172.16.122.2/30 is within AS 64496 and an IS-IS route to that prefix is available in the route table. The status codes in the following command on PE-1 show that the received FlowSpec IPv4 route is considered invalid.

```
*A:PE-1# show router bgp routes flow-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
```

```

Origin codes : i - IGP, e - EGP, ? - incomplete
l - leaked, x - stale, > - best, b - backup, p - purge

```

```

=====
BGP FLOW IPV4 Routes
=====

```

Flag	Network As-Path	Nexthop	LocalPref	MED
i	-- 64500	0.0.0.0	n/a	None

```

Community Action: rate-limit: 0 kbps
NLRI Subcomponents:
Dest Pref : 172.16.122.2/30
Src Pref : 172.16.115.2/30
Ip Proto : [ == 6 ]
Dest Port : [ == 4191 ]
Src Port : [ >1024 ]

```

```

-----
Routes : 1

```

The following route table on PE-1 shows that an IS-IS route is available toward destination prefix 172.16.122.0/30.

```
*A:PE-1# show router route-table 172.16.122.2
```

```

=====
Route Table (Router: Base)
=====

```

Dest Prefix[Flags] Next Hop[Interface Name]	Type	Proto	Age Metric	Pref
172.16.122.0/30 192.168.12.2	Remote	ISIS	04h41m53s 20	18

```

-----
No. of Routes: 1

```

Invalid routes are not advertised to the BGP peers, so the other nodes will not receive this route. The following BGP summary on PE-1 shows that one FlowSpec IPv4 route was received from the FlowSpec route server, but it remains inactive and no FlowSpec IPv4 route is sent to PE-2 or PE-3.

```
*A: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)
192.0.2.2 Def. Instance	64496	113 115	0 00h54m56s	2/0/2 (IPv4) 0/0/0 (FlowIPv4)
192.0.2.3 Def. Instance	64496	113 115	0 00h54m56s	2/2/2 (IPv4) 0/0/0 (FlowIPv4)
192.168.11.2 Def. Instance	64500	9	0 00h00m36s	0/0/2 (IPv4)

```

      8      0      1/0/0 (FlowIPv4)
-----
*A:PE-1#

```

The following command on PE-5 shows that IPv4 filter 1 does not have an embedded filter "fSpec-0".

```

*A:PE-5# show filter ip 1 embedded
=====
IP Filter embedding
=====
In      Offset  From              Inserted  Status
-----
1       0       fSpec-0           0/0      OK
=====
*A:PE-5#

```

On PE-5, IPv4 filter 1 does not have an embedded filter "fSpec-0" and the default action of IPv4 filter 1 is forward, so the traffic from IP SA 172.16.115.2 to IP DA 172.16.122.2 with destination port 4191 and source port 1025 will be forwarded to T2.

Valid eBGP FlowSpec Routes without Destination Prefix

The FlowSpec route server advertises a FlowSpec IPv4 route for IP traffic with source prefix 172.16.115.2/30, destination port 4191, and source port greater than 1024. No destination prefix subcomponent is included, so the FlowSpec IPv4 route will be considered valid. The following command on PE-1 shows that the FlowSpec IPv4 route without destination prefix subcomponent is valid, best, and used, while an almost identical FlowSpec IPv4 route with destination prefix subcomponent is invalid.

```

*A:PE-1# show router bgp routes flow-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 FLOW IPV4 Routes
=====
Flag  Network      Nexthop      LocalPref      MED
     As-Path
-----
u*>i  --           0.0.0.0      n/a            None
      64500

      Community Action: rate-limit: 0 kbps
      NLRI Subcomponents:
      Src Pref  : 172.16.115.2/30
      Ip Proto  : [ == 6 ]
      Dest Port : [ == 4191 ]
      Src Port  : [ >1024 ]
i      --           0.0.0.0      n/a            None
      64500

      Community Action: rate-limit: 0 kbps
      NLRI Subcomponents:
      Dest Pref : 172.16.122.2/30

```

```

Src Pref : 172.16.115.2/30
Ip Proto : [ == 6 ]
Dest Port : [ == 4191 ]
Src Port : [ >1024 ]
-----
Routes : 2

```

The valid FlowSpec IPv4 route without destination prefix subcomponent will be advertised to the other PEs. The FlowSpec IPv4 route is valid, best, and used on PE-5, as follows.

```

*A:PE-5# show router bgp routes flow-ipv4
=====
BGP Router ID:192.0.2.5      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 FLOW IPV4 Routes
=====
Flag Network          Nexthop          LocalPref      MED
  As-Path
-----
u*>i  --              0.0.0.0         100            None
      64496 64500

Community Action: rate-limit: 0 kbps
NLRI Subcomponents:
Src Pref : 172.16.115.2/30
Ip Proto : [ == 6 ]
Dest Port : [ == 4191 ]
Src Port : [ >1024 ]
-----
Routes : 1

```

Matching traffic originating from T1 will be discarded on PE-5, as follows.

```

*A:PE-5# monitor port 1/1/1 1/1/2 1/1/3 rate interval 3 repeat 2
=====
Monitor statistics for Ports
=====
                                Input          Output
-----
---snip---
-----
At time t = 3 sec (Mode: Rate)
-----
Port 1/1/1
-----
Octets          540459          27
Packets         4222            0
---snip---
-----
Port 1/1/3
-----
Octets          0                0
Packets         0                0
---snip---
-----

```



```

*A:PE-5# show filter ip 1

=====
IP Filter
=====
Filter Id       : 1                               Applied       : Yes
Scope          : Template                       Def. Action   : Forward
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)
Description    : (Not Specified)
-----
Filter Match Criteria : IP
-----
Entry          : 256
Origin         : Inserted by embedded filter fSpec-0 entry 256
Description    : (Not Specified)
Log Id        : n/a
Src. IP        : 172.16.115.2/30
Src. Port      : gt 1024
Dest. IP       : 0.0.0.0/0
Dest. Port     : eq 4191
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-syn        : Off                             TCP-ack       : Off
Option-pres    : Off
Egress PBR     : Disabled
Primary Action : Drop
Ing. Matches  : 321617 pkts (41166976 bytes)
Egr. Matches   : 0 pkts
=====
*A:PE-5#

```

Conclusion

Flow specifications received from iBGP or eBGP peers are by default accepted without validation. Flowspec routes with destination prefix subcomponent can be validated against BGP unicast routing.

BGP Graceful Restart and Long-Lived Graceful Restart

This chapter provides information about BGP Graceful Restart and Long-Lived Graceful Restart.

Topics in this chapter include:

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

Applicability

This chapter was initially written for SR OS Release 15.0.R8, but the CLI in the current edition corresponds to SR OS Release 19.10.R2.

Overview

BGP was designed assuming that peer router failures should be reacted to immediately so that the forwarding state of the router can converge toward the current state of the network. However, BGP is often used to signal Network Layer Reachability Information (NLRIs) associated with configuration rather than forwarding, such as flow specifications, Route Target (RT) constraints, BGP Auto-Discovery (BGP-AD), and BGP-VPLS. GR can be applied when there is fate separation between the control plane and the forwarding plane, allowing a restart of the control plane without affecting forwarding.

[Table 1: Supported address families for GR and LLGR in base router and in VPRN](#) lists the supported address families for GR and LLGR in the base router and in a BGP instance in a VPRN.

Table 1: Supported address families for GR and LLGR in base router and in VPRN

Address family	AFI/SAFI	BGP in base router	BGP in VPRN
IPv4 unicast	1/1	X	X
Labeled IPv4	1/4	X	X
VPN-IPv4	1/128	X	
RT constraint	1/132	X	
FlowSpec IPv4	1/133	X	X
IPv6 unicast	2/1	X	X
Labeled IPv6	2/4	X	

Address family	AFI/SAFI	BGP in base router	BGP in VPRN
VPN-IPv6	2/128	X	
FlowSpec IPv6	2/133	X	X
L2 VPN	25/65	X	

GR

GR can be applied in the general **bgp** context, in a BGP **group**, or per BGP **neighbor**. BGP GR can be applied for the base router or a VPRN. GR can be enabled as follows:

```
configure router bgp graceful-restart
configure router bgp group <groupName> graceful-restart
configure router bgp group <groupName> neighbor <neighborName> graceful-restart
configure service vprn <vprnId> bgp graceful-restart
configure service vprn <vprnId> bgp group <groupName> graceful-restart
configure service vprn <vprnId> bgp group <groupName> neighbor <neighborName>
                                         graceful-restart
```

The following shows the BGP configuration on the base router for multiple address families. GR is enabled with a stale routes time of 150 seconds and notifications will be sent. No restart time is configured explicitly; the default restart time is 300 seconds at group level and peer level; at BGP instance level, the default restart time is 120 seconds. LLGR is not configured.

```
# on PE-2:
configure
router
  bgp
    split-horizon
    group "iBGP"
      family ipv4 ipv6 vpn-ipv4 vpn-ipv6 l2-vpn flow-ipv4 flow-ipv6
      graceful-restart
        stale-routes-time 150
        enable-notification
    exit
    peer-as 64496
    neighbor 192.0.2.1
    exit
  exit
no shutdown
```

A BGP speaker can advertise a GR capability to indicate that it is able to preserve its forwarding state per address family (AF) during BGP restart. The GR capability can be used to inform the BGP peers that an end-of-RIB (EOR) message will be generated after all routing updates have been sent for an address family, as follows:

```
172 2020/02/12 11:49:58.321 UTC 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 = 0
  Total Path Attr Length = 7
  Flag: 0x90 Type: 15 Len: 3 Multiprotocol Unreachable NLRI:
    Address Family IPV6
End-of-Rib marker (IPV6)
```

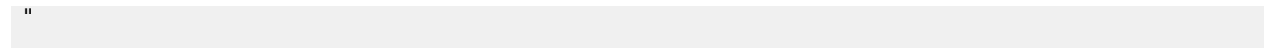
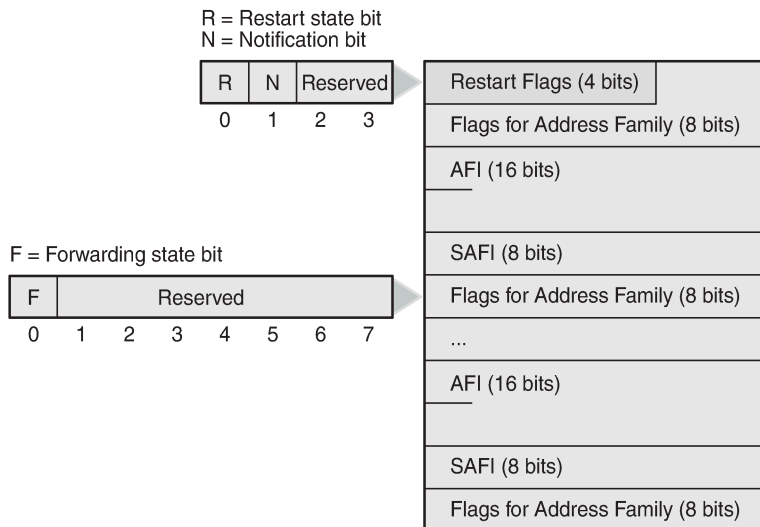


Figure 46: BGP GR capability shows the GR capability with restart flags, restart time, and forwarding flags per address family. RFC 4724 defines the GR BGP capability. The notification bit N is defined in *draft-ietf-idr-bgp-gr-notification-13*.

Figure 46: BGP GR capability



- Restart flags:
 - The restart state bit R is used to avoid a possible deadlock when multiple BGP speakers peering with each other restart simultaneously and are waiting for the EOR. When set (R=1), the bit indicates that the BGP speaker has restarted and its peer must not wait for the EOR before advertising routing information.
 - The notification bit N indicates that the BGP speaker is willing to send and receive BGP notification messages in GR mode, including the BGP Notification Cease message, which is a hard-reset message causing a peer to terminate a BGP session.
 - The remaining two restart flag bits are reserved and must be set to 0.
- The restart time in seconds is the estimated time required to re-establish a BGP session after a restart. When the restart time expires before the BGP session is re-established, the GR helper stops helping and the (stale) routes received from the failed BGP speaker are removed.
- Flags for address family:
 - The forwarding state bit F indicates whether the forwarding state for routes with a certain AFI/SAFI are preserved during BGP restart. When set (F=1), the forwarding state is preserved. After a hard reset caused by a BGP Notification Cease message, the forwarding bit must be set to 1.
 - The remaining bits are reserved and must be 0.

A BGP speaker can advertise GR capability without any AFI/SAFI, indicating that the sender cannot preserve its forwarding state during BGP restart, but supports procedures for the receiving speaker.

Debugging is enabled for BGP Open messages, as follows:

```
# on all PEs:
debug
  router
    bgp
      open
```

The following BGP Open message received by PE-2 from PE-1 shows the GR capability for different address families and with a default start timer of 300 seconds. The restart bit R is false because no GR is taking place on peer PE-1. The notification bit N is set to true. The same AFI/SAFI information is presented in the GR capability as in the MP-BGP capabilities, because GR is always enabled for all configured AFI/SAFIs. LLGR is not enabled yet.

```
50 2020/02/12 13:25:41.971 UTC MINOR: DEBUG #2001 Base BGP
"BGP: OPEN
Peer 1: 192.0.2.1 - Received BGP OPEN: Version 4
AS Num 64496: Holdtime 90: BGP_ID 192.0.2.1: Opt Length 84 (ExtOpt F)
Opt Para: Type CAPABILITY: Length = 82: Data:
  Cap_Code GRACEFUL-RESTART: Length 30
    Bytes: 0x41 0x2c 0x0 0x1 0x1 0x0 0x0 0x2 0x1 0x0 0x0 0x1 0x80 0x0 0x0 0x2 0x80 0x0 0x0
0x19 0x41 0x0 0x0 0x1 0x85 0x0 0x0 0x2 0x85 0x0
  Cap_Code MP-BGP: Length 4
    Bytes: 0x0 0x1 0x0 0x1 # AFI 1/SAFI 1 = IPv4 unicast
  Cap_Code MP-BGP: Length 4
    Bytes: 0x0 0x2 0x0 0x1 # AFI 2/SAFI 1 = IPv6 unicast
  Cap_Code MP-BGP: Length 4
    Bytes: 0x0 0x1 0x0 0x80 # AFI 1/SAFI 128 = VPN-IPv4
  Cap_Code MP-BGP: Length 4
    Bytes: 0x0 0x2 0x0 0x80 # AFI 2/SAFI 128 = VPN-IPv6
  Cap_Code MP-BGP: Length 4
    Bytes: 0x0 0x19 0x0 0x41 # AFI 25/SAFI 65 = L2 VPN
  Cap_Code MP-BGP: Length 4
    Bytes: 0x0 0x1 0x0 0x85 # AFI 1/SAFI 133 = IPv4 FlowSpec
  Cap_Code MP-BGP: Length 4
    Bytes: 0x0 0x2 0x0 0x85 # AFI 2/SAFI 133 = IPv6 FlowSpec
  Cap_Code ROUTE-REFRESH: Length 0
  Cap_Code 4-OCTET-ASN: Length 4
    Bytes: 0x0 0x0 0xfb 0xf0
"
```

The first two octets in the GR capability are 0x41 0x2c (01000001 00101100 in binary). The first four bits-0100-represent the restart flags: R=0, N=1, and the remaining two bits are reserved and set to 0. The remaining twelve bits-000100101100-represent the restart time in seconds: 256+32+8+4=300.

The following four octets in the GR capability are 0x0 0x1 0x1 0x0 (00000000 00000001 00000001 00000000 in binary). The first two octets represent AFI 1 for IPv4, the third octet SAFI 1 for unicast, and the last octet represents the flags, with the forwarding bit F=0 and all other bits reserved and set to zero. The other bytes are for the other AFI/SAFIs that are configured in the example.

Debugging is enabled for GR, as follows:

```
# on all PEs:
debug
  router
    bgp
      graceful-restart
```

The following messages are in the debug trace on PE-2. The first message shows restart bit R false (no restart ongoing), notification bit N true (GR notifications are supported), restart time 300s (default value), and notification restart false (no GR notifications were sent).

```
51 2020/02/12 13:25:41.971 UTC MINOR: DEBUG #2001 Base BGP
"BGP: RESTART
Peer 1: 192.0.2.1: Restart Capability Receive: restart BIT FALSE: Graceful Notification BIT
TRUE: Restart Time 300 secs: NOTIFICATION restart FALSE
"
```

The subsequent messages show the GR capabilities per address family with the value of the forwarding-preserved bit F, for example, for the IPv4 unicast address family, as follows. The forwarding-preserved bit is false.

```
52 2020/02/12 13:25:41.971 UTC MINOR: DEBUG #2001 Base BGP
"BGP: RESTART
Peer 1: 192.0.2.1: Restart Capability Receive: afi: AFI_IPV4 safi: SAFI_UNICAST forwarding-
preserved BIT FALSE
"
```

When routers have negotiated the GR capability for an address family and the BGP session drops, the BGP peers enter the GR helper state and do not immediately delete the routes of that address family received from the failed peer. The helpers mark these routes as stale and keep using them until the BGP session is restored, the BGP routes are refreshed, and an EOR message has been received for the AFI/SAFIs.

However, if the BGP session with the restarting router is not restored before the configured restart time expires, the peer router stops helping and will send withdraw messages for the routes received from the restarting router. When the stale routes time expires, the router will withdraw all routes received from the restarting router. The restart time has an upper bound of 4095 seconds, so this mechanism is designed for relatively short outages in the order of minutes, not for hours. GR can deal with simple control plane restarts in terms of scope and severity.

```
*A:PE-1# configure router bgp graceful-restart restart-time
- restart-time <seconds>
- no restart-time

<seconds>          : [0..4095]
```

LLGR

LLGR can handle failure scenarios where the repair takes several hours, such as a network where redundant route reflectors (RRs) fail simultaneously and the configuration-type BGP routes (that is, non-forwarding BGP routes) for FlowSpec, route target, and L2 VPNs can be preserved. BGP routes for forwarding can also be preserved longer. LLGR can be enabled for all address families that have GR enabled, or for a subset of these address families. LLGR allows a BGP session to stay down for hours or even days. The advertised stale time has an upper bound of 16777215 seconds and the default value is 86400 seconds. LLGR is configured in the GR context, which is in the general **bgp** context, per group, or per neighbor.

```
*A:PE-1# configure router bgp graceful-restart long-lived advertised-stale-time
- advertised-stale-time <seconds>
- no advertised-stale-time
```

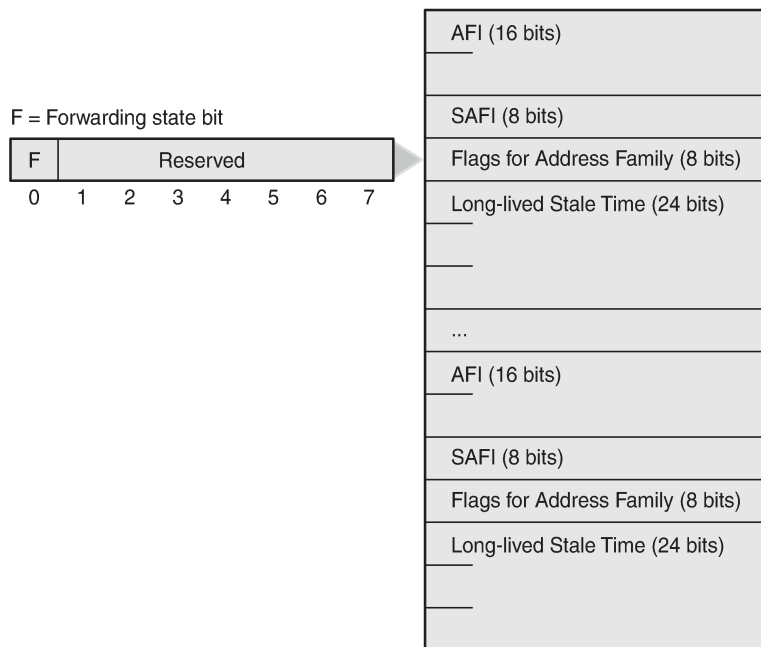
```
<seconds> : [0..16777215]
```

When GR is enabled, it automatically applies for all configured AFs; LLGR can be configured per AF, possibly with different LLGR-stale times, for example, for the L2 VPN address family in group "iBGP", as follows:

```
# on PE-1:
configure
router
  bgp
    group "iBGP"
      graceful-restart
        long-lived
          family l2-vpn
            advertised-stale-time 7200
          exit
```

Figure 47: LLGR capability shows the LLGR capability—as defined in draft-uttaro-idr-bgp-persistence-03—that adds a long-lived stale time per address family. The LLGR capability must be advertised in conjunction with the GR capability.

Figure 47: LLGR capability



27618

GR and LLGR are configured in the "iBGP" group for all configured AFI/SAFIs, as follows. The default value of the **long-lived advertised-stale-time** is 86400 seconds.

```
configure
router
  bgp
    group "iBGP"
      family ipv4 ipv6 vpn-ipv4 vpn-ipv6 l2-vpn flow-ipv4 flow-ipv6
      graceful-restart
        stale-routes-time 150
```

```

enable-notification
long-lived
  advertised-stale-time 3600
exit
exit
exit

```

When LLGR is enabled, the BGP Open message contains a long-lived GR capability and a GR capability, with the supported AFI/SAFIs. The following BGP Open message is received by PE-2 from RR PE-1. GR and LLGR are supported for all the AFI/SAFIs in the BGP session.

```

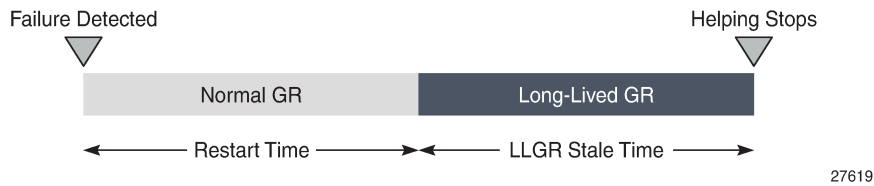
280 2020/02/12 13:56:38.304 UTC MINOR: DEBUG #2001 Base BGP
"BGP: OPEN
Peer 1: 192.0.2.1 - Received BGP OPEN: Version 4
AS Num 64496: Holdtime 90: BGP_ID 192.0.2.1: Opt Length 135 (ExtOpt F)
Opt Para: Type CAPABILITY: Length = 133: Data:
  Cap_Code GRACEFUL-RESTART: Length 30
  Bytes: 0x41 0x2c 0x0 0x1 0x1 0x0 0x0 0x2 0x1 0x0 0x0 0x1 0x80 0x0 0x0 0x2 0x80 0x0 0x0
0x19 0x41 0x0 0x0 0x1 0x85 0x0 0x0 0x2 0x85 0x0
  Cap_Code MP-BGP: Length 4
  Bytes: 0x0 0x1 0x0 0x1
  Cap_Code MP-BGP: Length 4
  Bytes: 0x0 0x2 0x0 0x1
  Cap_Code MP-BGP: Length 4
  Bytes: 0x0 0x1 0x0 0x80
  Cap_Code MP-BGP: Length 4
  Bytes: 0x0 0x2 0x0 0x80
  Cap_Code MP-BGP: Length 4
  Bytes: 0x0 0x19 0x0 0x41
  Cap_Code MP-BGP: Length 4
  Bytes: 0x0 0x1 0x0 0x85
  Cap_Code MP-BGP: Length 4
  Bytes: 0x0 0x2 0x0 0x85
  Cap_Code ROUTE-REFRESH: Length 0
  Cap_Code 4-OCTET-ASN: Length 4
  Bytes: 0x0 0x0 0xfb 0xf0
  Cap_Code LONG-LIVED-GR: Length 49
  Bytes: 0x0 0x1 0x1 0x0 0x0 0xe 0x10 0x0 0x2 0x1 0x0 0x0 0xe 0x10 0x0 0x1 0x80 0x0 0x0
0xe 0x10 0x0 0x2 0x80 0x0 0x0 0xe 0x10 0x0 0x19 0x41 0x0 0x0 0xe 0x10 0x0 0x1 0x85 0x0 0x0 0xe
0x10 0x0 0x2 0x85 0x0 0x0 0xe 0x10
"

```

The first seven octets in the LLGR capability—0x0 0x1 0x1 0x0 0x0 0xe 0x10—are for AF IPv4 unicast. The first two—0x0 0x1—represent AFI=1 for IPv4, the third—0x1—represents SAFI=1 for unicast, the fourth—0x0—indicates that the forwarding-preserved bit F=0 (and the other bits are reserved and must be zero). The next three octets represent the LLGR-stale time: 0x0 0xe 0x10 (00000000 00001110 00010000 in binary) is 2048 + 1024 + 512 + 16 = 3600 in decimal.

The LLGR-stale time in seconds specifies how long LLGR-stale routes for the AFI/SAFI may be retained, possibly added to the GR time. LLGR starts when GR terminates before the failed router has recovered, that is, when either the restart timer or the stale-routes timer expires (whichever expires first), as shown in [Figure 48: GR and LLGR](#). LLGR ends when the advertised LLGR-stale time expires or when the failure is restored and all routes are re-advertised followed by an EOR message. When the AFI/SAFI is not listed in the GR capability, the restart time for GR is 0 seconds. The LLGR-stale time is defined by the **advertised-stale-time** option, which has a default value of 86400 seconds.

Figure 48: GR and LLGR



The forwarding-preserved bit F is configured with the following command. By default, all F bits are 0, indicating that the forwarding state was not preserved during the previous restart. The **forwarding-bits-set** command allows F bits for all AFI/SAFIs to be set to 1, or only the F bits for configuration-type (that is, non-forwarding) AFI/SAFIs, such as L2 VPN, route target, IPv4 FlowSpec, and IPv6 FlowSpec.

```
*A:PE-1# configure router bgp group "iBGP" graceful-restart long-lived forwarding-bits-set
- forwarding-bits-set {all|non-fwd}
- no forwarding-bits-set
```

An address family is only protected with LLGR if the AFI/SAFI is in the advertised LLGR capability and in the received LLGR capability. In SR OS, LLGR can only be enabled when GR is enabled, so each advertised LLGR capability comes with a GR capability. If a peer advertises the LLGR capability without GR capability, the LLGR capability is ignored.

GR is used for short outages where the helpers pretend that everything is normal; LLGR is for longer outages where the helpers inform the other peers. [Table 2: Helper actions during GR and LLGR](#) shows a comparison of the helper actions during GR and LLGR.

Table 2: Helper actions during GR and LLGR

Helper actions during GR	Helper actions during LLGR
Mark GR-eligible routes from the failed peer as stale	Mark LLGR-eligible routes from the failed peer as LLGR-stale
Attempt to reconnect to the peer at periodic intervals	Attempt to reconnect to the peer at periodic intervals
	Depreference LLGR-stale routes so that they are less preferred than any valid non-LLGR-stale route
	If an LLGR-stale route remains the best path, inform the other peers by withdrawing the route or re-advertising the route with new attributes

A route is said to be deprefferenced if it has its route selection preference reduced in reaction to some event. LLGR automatically deprefferences LLGR-stale routes so that any valid non-LLGR-stale route for the same NLRI is more preferred. When advertising LLGR-stale routes to an LLGR-capable peer, LLGR adds

the well-known **llgr-stale** BGP community to the routes, so that the LLGR-capable BGP peers can also deprefer the LLGR-stale routes. The following option controls how LLGR-stale routes are advertised.

```
*A:PE-1# configure router bgp group "iBGP" graceful-restart long-lived advertise-stale-to-all-
neighbors
- advertise-stale-to-all-neighbors [without-no-export]
- no advertise-stale-to-all-neighbors

<without-no-export> : keyword - Advertise stale routes to neighbors with the
                        addition of the LLGR_STALE community
```

- The default is **no advertise-stale-to-all-neighbors**, in which case LLGR-aware routers re-advertise stale best routes to their LLGR-aware peers, with the addition of the well-known **llgr-stale** community. Toward BGP peers that did not advertise the LLGR capability, the stale routes are withdrawn.
- When **advertise-stale-to-all-neighbors** is configured combined with the default **no without-no-export**, the LLGR-stale routes are withdrawn toward eBGP peers that did not advertise the LLGR capability and re-advertised to all other peers with LLGR-stale community. Toward iBGP (and confederation-eBGP) peers that signaled the LLGR capability, the route is re-advertised with the well-known **llgr-stale** and **no-export** communities and the local preference is set to 0.
- When **advertise-stale-to-all-neighbors** is configured combined with **without-no-export**, the LLGR-stale routes are withdrawn toward eBGP peers that did not advertise the LLGR capability and re-advertised to all other peers with LLGR-stale community. Toward iBGP (and confederation-eBGP) peers that signaled the LLGR capability, the route is re-advertised with the LLGR-stale community, but without the no-export community. The local preference is set to 0.

Route policies can match, delete, or add the BGP well-known communities **llgr-stale** and **no-llgr**.

An iBGP peer not supporting LLGR normally does not receive route updates with LLGR-stale community, but if it does, it can only deprefer them based on local preference 0.

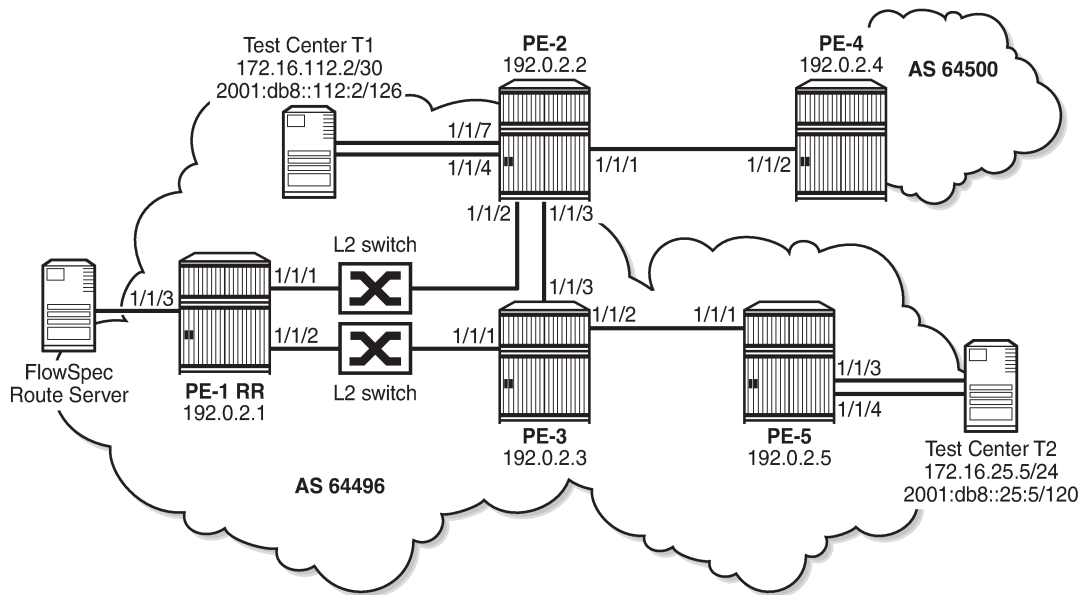
The LLGR-stale routes timer is not stopped when the BGP session with the failed peer is re-established; it only stops when the EOR is received for the AFI/SAFI. When the LLGR-stale routes time expires for an AFI/SAFI, the LLGR phase ends and all remaining LLGR-stale routes for that AFI/SAFI are deleted. However, stale routes will also be deleted before the LLGR stale-routes timer expires when the BGP session with the failed peer is re-established and either of the following applies:

- the GR or LLGR capability is missing
- the AFI/SAFI is missing from the LLGR capability
- the forwarding state bit F=0 for the AFI/SAFI

Configuration

[Figure 49: Example topology](#) shows the example topology with four routers in AS 64496. PE-1 combines the roles of a PE and an RR. A FlowSpec route server sends IPv4 and IPv6 FlowSpec routes to PE-1. Test centers T1 and T2 generate IPv4 and IPv6 traffic to each other, through the base router or a VPLS service. PE-4 is in AS 64500 and has an eBGP session with PE-2 in AS 64496.

Figure 49: Example topology



27620

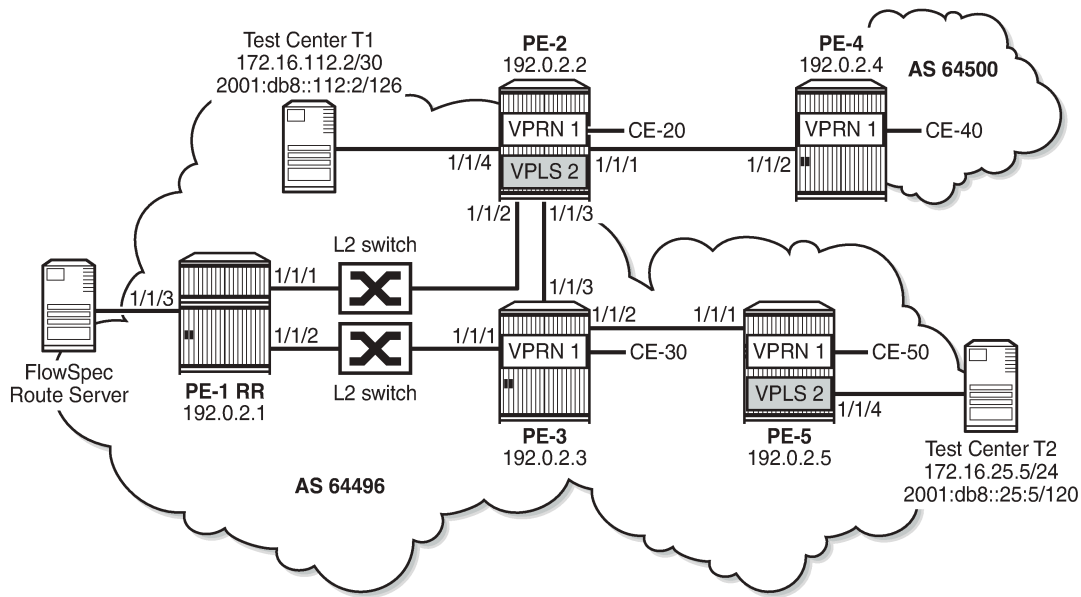
Initial configuration

The initial configuration on the nodes includes:

- Cards, MDAs, ports
- Router interfaces with dual stack
- IS-IS on all interfaces of the routers in AS 64496 (alternatively, OSPF can be used)
- LDP on all interfaces in AS 64496, not between PE-2 and PE-4
- MPLS and RSVP on all interfaces in AS 64496, not between PE-2 and PE-4
- RSVP-TE LSPs between PE-2 and PE-5

Figure 5 shows the configured services on the PEs. VPRN 1 is configured on PE-2, PE-3, PE-4, and PE-5; VPLS 2 with BGP-AD on PE-2 and PE-5.

Figure 50: VPRN 1 and VPLS 2 in the example topology



27621

The service configuration on PE-2 is as follows. The pseudowire (PW) template is required for BGP-AD in VPLS 2, as described in the *LDP VPLS Using BGP-Auto Discovery* chapter.

```
# on PE-2:
configure
service
  pw-template 1 name "PW 1" create
    split-horizon-group "vpls-shg"
  exit
exit
vprn 1 name "VPRN 1" customer 1 create
  route-distinguisher 64496:1
  auto-bind-tunnel
  resolution any
exit
vrf-target target:64496:1
interface "int-VPRN1-PE-2-CE-20" create
  address 172.16.2.1/30
  ipv6
    address 2001:db8::1:2:1/126
  exit
  sap 1/1/5:1 create
  exit
exit
no shutdown
exit
vpls 2 name "VPLS 2" customer 1 create
  bgp
    route-distinguisher 64496:2
    route-target export target:64496:2 import target:64496:2
    pw-template-binding 1 import-rt "target:64496:2"
  exit
exit
bgp-ad
  vpls-id 64496:2
```

```

        vsi-id
            prefix 192.0.2.2
        exit
        no shutdown
    exit
    sap 1/1/5:2 create
    exit
    sap 1/1/4 create
    exit
    no shutdown
exit

```

For the exchange of the routes in the VPRN, the VPN IPv4 and VPN IPv6 address families need to be configured in BGP; for BGP-AD, the L2 VPN address family. BGP is configured on all PEs for the following address families: IPv4, IPv6, VPN-IPv4, VPN-IPv6, L2 VPN, IPv4 FlowSpec, and IPv6 FlowSpec. On RR PE-1, the initial BGP configuration is as follows. The "iBGP" group includes all the PEs in AS 64496, whereas the "FlowSpec" group includes the FlowSpec server only.

```

# on RR PE-1:
configure
router
    bgp
        split-horizon
        group "iBGP"
            family ipv4 ipv6 vpn-ipv4 vpn-ipv6 l2-vpn flow-ipv4 flow-ipv6
            cluster 192.0.2.1
            peer-as 64496
            neighbor 192.0.2.2
            exit
            neighbor 192.0.2.3
            exit
            neighbor 192.0.2.5
            exit
        exit
        group "FlowSpec"
            family ipv4 ipv6 flow-ipv4 flow-ipv6
            peer-as 64496
            neighbor 192.168.11.2
            exit
        exit
    exit
exit

```

On PE-2, the prefixes toward the test center T1 are exported. BGP is configured as follows:

```

# on PE-2:
configure
router
    policy-options
        begin
        prefix-list "T1"
            prefix 172.16.112.0/28 longer
            prefix 2001:db8::112:0/124 longer
        exit
        policy-statement "export-T1"
            entry 10
                from
                    protocol direct
                    prefix-list "T1"
                exit
            action accept
            exit
        exit
    exit

```

```

        exit
        commit
    exit
    bgp
        split-horizon
        group "eBGP"
            family ipv4 ipv6 vpn-ipv4 vpn-ipv6 l2-vpn flow-ipv4 flow-ipv6
            local-as 64496
            peer-as 64500
            neighbor 192.168.24.2
        exit
    exit
    group "iBGP"
        family ipv4 ipv6 vpn-ipv4 vpn-ipv6 l2-vpn flow-ipv4 flow-ipv6
        export "export-T1"
        peer-as 64496
        neighbor 192.0.2.1
    exit
    exit
exit

```

The configuration on PE-5 is similar, but without the "eBGP" group, and for the export, the prefixes from T2 are included, as follows.

```

#on PE-5:
configure
router
    policy-options
        begin
        prefix-list "T2"
            prefix 172.16.225.0/28 longer
            prefix 2001:db8::225:0/124 longer
        exit
        policy-statement "export-T2"
            entry 10
                from
                    protocol direct
                    prefix-list "T2"
                exit
                action accept
            exit
        exit
    exit
    commit
exit
bgp
    split-horizon
    group "iBGP"
        family ipv4 ipv6 vpn-ipv4 vpn-ipv6 l2-vpn flow-ipv4 flow-ipv6
        export "export-T2"
        peer-as 64496
        neighbor 192.0.2.1
    exit
    exit
exit

```

On PE-3, the BGP configuration is similar, without the export policy.

The BGP configuration on PE-4 is as follows:

```

# on PE-4:
configure
router

```

```

bgp
  split-horizon
  group "eBGP"
    family ipv4 ipv6 vpn-ipv4 vpn-ipv6 l2-vpn flow-ipv4 flow-ipv6
    local-as 64500
    peer-as 64496
    neighbor 192.168.24.1
  exit
exit
    
```

BGP routes

Under normal conditions, BGP routes of all the configured address families are advertised. The BGP summary on PE-5 shows the following number of received (Rcv), active (Act), and sent (Sent) BGP routes per address family for neighbor 192.0.2.1. Similar numbers occur on the other RR clients PE-2 and PE-3.

```

*A:PE-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      69   0 00h01m25s 1/1/1 (IPv4)
                21   0                1/0/1 (IPv6)
                4/3/2 (VpnIPv4)
                4/3/2 (VpnIPv6)
                1/1/1 (L2VPN)
                1/1/0 (FlowIPv4)
                1/1/0 (FlowIPv6)
=====
    
```

On PE-2, the following BGP IPv4 route is valid, best, and used.

```

*A:PE-2# show router bgp routes

=====
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 IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)    Path-Id    IGP Cost
      As-Path              Label
-----
u*>i  172.16.225.0/30        100        None
      192.0.2.5           None        20
      No As-Path         -
    
```

```
-----
Routes : 1
```

On PE-2, the following BGP L2 VPN route received from neighbor PE-1 (RR) is valid, best, and used.

```
*A:PE-2# show router bgp neighbor 192.0.2.1 received-routes l2-vpn
=====
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 L2VPN Routes
=====
Flag  RouteType      Prefix      MED
      RD           SiteId
      Nexthop      VeId        BlockSize  LocalPref
      As-Path      BaseOffset  vplsLabelBa
                               se
-----
u*>i  AutoDiscovery    192.0.2.5  -          0
      64496:2        -          -          -
      192.0.2.5     -          -          100
      No As-Path    -          -          -
-----
Routes : 1
```

On PE-2, the following active IPv6 FlowSpec route specifies that all traffic will be dropped (rate limit: 0 kbps) that matches the criteria: DA 2001:db8::225:2/126, SA 2001:db8::112:2/126, destination port 4191, and source port greater than 1024. This route is generated by the FlowSpec route server connected to PE-1.

```
*A:PE-2# show router bgp routes flow-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 FLOW IPV6 Routes
=====
Flag  Network      Nexthop      LocalPref      MED
      As-Path
-----
u*>i  --           ::           100            None
      No As-Path

Community Action: rate-limit: 0 kbps
NLRI Subcomponents:
Dest Pref : 2001:db8::225:2/126 offset 0
Src Pref  : 2001:db8::112:2/126 offset 0
Ip Proto  : [ == 6 ]
Dest Port : [ == 4191 ]
Src Port  : [ >1024 ]
-----
```



```
Routes : 1
```

The following sections describe:

- Default BGP behavior without GR
- GR
- LLGR

Default BGP behavior without GR

The RR PE-1 is isolated from the other PEs by disabling the ports toward PE-2 and PE-3, as follows:

```
# on PE-1:
configure
  port 1/1/1
    shutdown
  exit
  port 1/1/2
    shutdown
  exit
```

All BGP sessions with the BGP peers drop and the BGP peers remove the routes received from RR PE-1; for example, the list of IPv4 routes on PE-2 is empty. The same is true for the other configured address families.

```
*A:PE-2# show router bgp routes
=====
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 IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                       Path-Id    IGP Cost
      As-Path                                Label
-----
No Matching Entries Found.
=====
```

The following BGP summary on PE-2 shows that the session toward PE-4 is established, but the session toward PE-1 is down (state: Connect). A similar output is seen on the other PEs in AS 64496, because all BGP sessions toward the RR are down.

```
*A:PE-2# show router bgp summary all
=====
BGP Summary
=====
Legend : D - Dynamic Neighbor
=====
Neighbor
Description
```

ServiceId	AS	PktRcvd PktSent	InQ OutQ	Up/Down	State	Rcv/Act/Sent (Addr Family)

192.0.2.1						
Def. Instance	64496	108 8	0 0	00h01m10s	Connect	

192.168.24.2						
Def. Instance	64500	137 211	0 0	01h05m02s	0/0/0 (IPv4) 0/0/0 (IPv6) 1/1/2 (VpnIPv4) 1/1/2 (VpnIPv6) 0/0/1 (L2VPN) 0/0/0 (FlowIPv4) 0/0/0 (FlowIPv6)	

The ports on PE-1 are re-enabled and the BGP routes are re-advertised.

```
# on PE-1:
configure
  port 1/1/1
    no shutdown
  exit
  port 1/1/2
    no shutdown
  exit
```

GR

On all PEs, GR is enabled with a stale routes time of 150 seconds and notification enabled, as follows. The default restart time is 300 seconds, but the stale routes will already be deleted when the stale-routes time expires after 150 seconds. LLGR is not enabled yet.

```
# on PE-1, PE-2, PE-3, PE-5:
configure
  router
    bgp
      group "iBGP"
        graceful-restart
        stale-routes-time 150
        enable-notification
      exit
    exit
```

RR PE-1 is isolated, as follows:

```
# on PE-1:
configure
  port 1/1/1
    shutdown
  exit
  port 1/1/2
    shutdown
  exit
```

When the hold timer expires, the BGP session goes down, and the BGP peers enter the helper mode, RR PE-1 as well as its clients. The following debug message occurs on PE-2 if debugging is enabled for graceful restart:

```
153 2020/02/12 12:50:26.757 UTC MINOR: DEBUG #2001 Base BGP
"BGP: RESTART
Peer VR 1: Group iBGP: Peer 192.0.2.1: entering helper mode due to reason hold_timer_expiry
"
```

Log 99 logs the event as follows:

```
119 2020/02/12 12:50:26.757 UTC WARNING: BGP #2018 Base VR 1
"(ASN 64496) Peer 1: 192.0.2.1: graceful restart status changed to restarting"
```

The client PEs do not remove the routes they received from RR PE-1 immediately, but they mark these routes as stale and they keep using them. In the following list of IPv4 unicast routes, the x-status code indicates that the route is stale.

```
*A:PE-2# show router bgp routes
=====
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 IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>xi 172.16.225.0/30          100        None
      192.0.2.5             None        20
      No As-Path            -
-----
Routes : 1
```

When the BGP sessions are restored and an EOR is received for the AFI/SAFIs, the BGP routes are re-advertised and there are no longer any stale routes. However, if the stale routes timer expires before an EOR is received for the AFI/SAFIs, the stale routes are removed. The following command shows that there are no longer any BGP IPv4 routes in PE-2.

```
*A:PE-2# show router bgp routes
=====
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 IPv4 Routes
=====
Flag  Network                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
```

```

As-Path                                     Label
-----
No Matching Entries Found.
=====

```

When the stale routes timer expires before an EOR is received for the AFI/SAFIs, the GR phase is terminated and the PE is no longer a GR helper. The following debug messages are logged on PE-2 when debugging is enabled for GR:

```

154 2020/02/12 12:52:56.757 UTC MINOR: DEBUG #2001 Base BGP
"BGP: RESTART
BGP trying to exit helper for peer Peer 1: 192.0.2.1 with reason stale-routes-time expired for
all address families
"

155 2020/02/12 12:52:56.757 UTC MINOR: DEBUG #2001 Base BGP
"BGP: RESTART
BGP flushing stale routes for peer Peer 1: 192.0.2.1 AF All Address Families
"

156 2020/02/12 12:52:56.758 UTC MINOR: DEBUG #2001 Base BGP
"BGP: RESTART
Peer 1: 192.0.2.1: exit helper mode due to reason stale-routes-time expired
"

```

The following message is logged in log 99 on PE-2.

```

139 2020/02/12 12:52:56.758 UTC WARNING: BGP #2018 Base VR 1
"(ASN 64496) Peer 1: 192.0.2.1: graceful restart status changed to notHelping"

```

The situation on PE-1 is restored and the routes are re-advertised.

```

# on PE-1:
configure
  port 1/1/1
    no shutdown
  exit
  port 1/1/2
    no shutdown
  exit

```

In the example, the stale routes time is 150 seconds and the restart time 300 seconds. The helper mode stops when either of these timers expires. When the stale routes time is increased to 400 seconds and the restart time remains 300 seconds, the helper mode will stop when the restart time expires, as shown by the following debug message.

```

265 2020/02/12 13:40:44.758 UTC MINOR: DEBUG #2001 Base BGP
"BGP: RESTART
Peer 1: 192.0.2.1: exit helper mode due to reason restart-time expired
"

```

LLGR

Initially, LLGR will be configured with the same LLGR-stale time for all the configured AFI/SAFIs, but it is possible to configure LLGR with a different LLGR-stale time per AF. The LLGR-stale time is configured

as **advertised-stale-time**—which is the value that is advertised to the BGP peer—but can be overridden locally without being advertised.

At first, LLGR will be enabled on the "iBGP" group on PE-1, PE-2, PE-3, and PE-5. Later, LLGR will also be enabled on the "eBGP" group on PE-2 and PE-4.

LLGR enabled on iBGP sessions

The following configuration enables LLGR as well as GR in the "iBGP" group on all PEs in AS 64496 for all the already configured AFI/SAFIs.

```
# on PE-1, PE-2, PE-3, PE-5:
configure
  router
    bgp
      group "iBGP"
        graceful-restart
          stale-routes-time 150
          enable-notification
          long-lived
          advertised-stale-time 3600
        exit
      exit
    exit
```

Neither GR nor LLGR is enabled in the "eBGP" group on PE-2 and PE-4. This makes no difference for the GR phase on PE-2; only for the LLGR phase.

When the RR PE-1 gets isolated and the hold timer for the BGP session expires, the GR phase starts for the "iBGP" group and the routes received from PE-1 are marked as stale, but remain in use. In the GR phase, the detailed information for the stale IPv4 route 172.16.225.0/30 on PE-2 shows the flags used, valid, best, IGP, and stale (not LLGR-stale), as follows. PE-2 will keep using the stale routes in the GR phase. PE-2 will not withdraw any stale routes and eBGP peer PE-4 remains unaware of the failure.

```
*A:PE-2# show router bgp routes detail
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
---snip---
=====
BGP IPv4 Routes
=====
Original Attributes

Network       : 172.16.225.0/30
Nextthop     : 192.0.2.5
Path Id      : None
From         : 192.0.2.1
Res. Protocol : ISIS           Res. Metric   : 20
Res. Nextthop : 192.168.23.2
Local Pref.  : 100             Interface Name : int-PE-2-PE-3
---snip---
Community    : No Community Members
Cluster      : 192.0.2.1
Originator Id : 192.0.2.5         Peer Router Id : 192.0.2.1
Fwd Class    : None           Priority       : None
Flags      : Used Valid Best IGP Stale
Route Source : Internal
---snip---
```

The routes keep the stale flag "x", as in the GR phase. The following IPv4 route is marked as stale on PE-2:

```
*A:PE-2# show router bgp routes
=====
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 IPv4 Routes
=====
Flag Network                               LocalPref MED
      Nexthop (Router)                     Path-Id  IGP Cost
      As-Path                               Label
-----
u*>xi 172.16.225.0/30                       100      None
      192.0.2.5                             None     20
      No As-Path                             -
-----
Routes : 1
```

The following detailed information for this route on PE-2 shows the LLGR-stale flag instead of the normal stale flag, as follows:

```
*A:PE-2# show router bgp routes detail
=====
BGP Router ID:192.0.2.2      AS:64496      Local AS:64496
=====
---snip---
=====
BGP IPv4 Routes
=====
Original Attributes

Network       : 172.16.225.0/30
Nexthop       : 192.0.2.5
Path Id       : None
From          : 192.0.2.1
Res. Protocol : ISIS           Res. Metric   : 20
Res. Nexthop  : 192.168.23.2
Local Pref.   : 100           Interface Name : int-PE-2-PE-3
---snip---
Community   : No Community Members
Cluster       : 192.0.2.1
Originator Id : 192.0.2.5         Peer Router Id : 192.0.2.1
Fwd Class     : None          Priority       : None
Flags       : Used Valid Best IGP LlgrStale
---snip---
```

When debugging is enabled for GR, the following message on PE-2 is generated when the GR phase starts.

```
873 2020/02/12 10:44:40.453 UTC MINOR: DEBUG #2001 Base BGP
"BGP: RESTART
Peer VR 1: Group iBGP: Peer 192.0.2.1: entering helper mode due to reason hold_timer_expiry
"
```

The following message on PE-2 is generated when the LLGR phase starts.

```
883 2020/02/12 10:47:10.453 UTC MINOR: DEBUG #2001 Base BGP
"BGP: RESTART
Peer VR 1: Group iBGP: Peer 192.0.2.1: Entering helper, LLGR Phase - reason llgr_Start_on_rt
RtTm_pop
"
```

In the LLGR phase, the following command on PE-2 shows that, for the BGP session with RR PE-1, GR and LLGR are both enabled locally and on the BGP peer, and the GR and LLGR status of the peer PE-1 is "received restart request", so the LLGR phase is ongoing.

The advertised NLRIs of the RR PE-1 and its client PE-2 are similar, so the same AFI/SAFIs (and stale time) have been advertised by PE-2 and received from peer PE-1 for GR, GR notification, and LLGR. LLGR can only work if it is enabled on both BGP peers, which is the case.

```
*A:PE-2# show router bgp neighbor 192.0.2.1 graceful-restart

=====
BGP Neighbor 192.0.2.1 Graceful Restart
=====
Graceful Restart locally configured for peer: Enabled
GR Notification : Enabled
Peer's Graceful Restart feature : Enabled
NLRI(s) that peer supports restart for : ipv4 ipv6 vpn-ipv4 vpn-ipv6
                                         l2-vpn flow-ipv4 flow-ipv6
NLRI(s) that peer saved forwarding for : ipv4 ipv6 vpn-ipv4 vpn-ipv6
                                         l2-vpn flow-ipv4 flow-ipv6
NLRI(s) that restart is negotiated for : None
NLRI(s) of received end-of-rib markers : None
NLRI(s) of all end-of-rib markers sent : None
NLRI(s) peer supports NOTIFICATION GR for : ipv4 ipv6 vpn-ipv4 vpn-ipv6
                                         l2-vpn flow-ipv4 flow-ipv6
Restart time locally configured for peer : 300 seconds
Restart time requested by the peer : 300 seconds
Time until stale routes are deleted or
become long-lived stale : 150 seconds
Graceful restart status on the peer : Rcvd restart request
Long-Lived GR status on the peer : Rcvd restart request
Number of Restarts : 1
Last Restart at : 02/12/2020 09:19:52
-----
LLGR Configuration : Enabled
Peer's LLGR feature : Enabled
NLRI(s) peer signaled LLGR for & stale time
& F-bit : ipv4 : 3600 seconds (F)
                                         ipv6 : 3600 seconds (F)
                                         vpn-ipv4 : 3600 seconds (F)
                                         vpn-ipv6 : 3600 seconds (F)
                                         l2-vpn : 3600 seconds (F)
                                         flow-ipv4 : 3600 seconds (F)
                                         flow-ipv6 : 3600 seconds (F)
NLRI(s) LLGR negotiated for and stale time : ipv4 : 3600 seconds
                                         ipv6 : 3600 seconds
                                         vpn-ipv4 : 3600 seconds
                                         vpn-ipv6 : 3600 seconds
                                         l2-vpn : 3600 seconds
                                         flow-ipv4 : 3600 seconds
                                         flow-ipv6 : 3600 seconds
LLGR Restart time overridden for the peer : n/a
NLRI(s) LLGR advertised & stale time & F-bit: ipv4 : 3600 seconds
                                         ipv6 : 3600 seconds
```

```

vpn-ipv4 : 3600 seconds
vpn-ipv6 : 3600 seconds
l2-vpn : 3600 seconds
flow-ipv4 : 3600 seconds
flow-ipv6 : 3600 seconds
=====

```

On PE-2, the following command shows that GR and LLGR are disabled for the eBGP session with PE-4.

```

*A:PE-2# show router bgp neighbor 192.168.24.2 graceful-restart

=====
BGP Neighbor 192.168.24.2 Graceful Restart
=====
Graceful Restart locally configured for peer: Disabled
GR Notification : Disabled
Peer's Graceful Restart feature : Disabled
NLRI(s) that peer supports restart for : None
NLRI(s) that peer saved forwarding for : None
NLRI(s) that restart is negotiated for : None
NLRI(s) of received end-of-rib markers : ipv4 ipv6
NLRI(s) of all end-of-rib markers sent : ipv4 ipv6
NLRI(s) peer supports NOTIFICATION GR for : None
Restart time locally configured for peer : 120 seconds
Restart time requested by the peer : 0 seconds
Time until stale routes are deleted or
become long-lived stale : 360 seconds
Graceful restart status on the peer : Not currently being helped
Long-Lived GR status on the peer : Not currently being helped
Number of Restarts : 0
Last Restart at : Never
-----
LLGR Configuration : Disabled
Peer's LLGR feature : Disabled
NLRI(s) peer signaled LLGR for & stale time
& F-bit : n/a
NLRI(s) LLGR negotiated for and stale time : n/a
LLGR Restart time overridden for the peer : n/a
NLRI(s) LLGR advertised & stale time & F-bit: n/a
=====

```

In the LLGR phase, the stale routes remain stale, but are deprefferenced. In this example, there are no alternative routes with a better preference, so the stale routes remain valid, best, and used. Traffic between PE-2, PE-3, and PE-5 is still uninterrupted.

However, the eBGP session between PE-2 and PE-4 does not have LLGR enabled. In the LLGR phase, the LLGR-stale routes are immediately withdrawn by PE-2; for example, the following BGP update withdraws the VPN-IPv4 routes toward PE-4. Therefore, VPN traffic can no longer be exchanged between VPRN 1 on PE-3 (or PE-5) and VPRN 1 on PE-4.

```

884 2020/02/12 10:47:10.458 UTC MINOR: DEBUG #2001 Base Peer 1: 192.168.24.2
"Peer 1: 192.168.24.2: UPDATE
Peer 1: 192.168.24.2 - Send BGP UPDATE:
  Withdrawn Length = 5
  172.16.225.0/30
  Total Path Attr Length = 39
  Flag: 0x90 Type: 15 Len: 35 Multiprotocol Unreachable NLRI:
  Address Family VPN_IPV4
  172.16.5.0/30 RD 64496:1 Label 0
  172.16.3.0/30 RD 64496:1 Label 0
"

```


Even though GR is also disabled for the eBGP session between PE-2 and PE-4, the routes are only withdrawn in the LLGR phase, not in the GR phase. GR is meant for short interruptions where the GR helper PE-2 pretends that the situation is normal and traffic can be forwarded based on stale routes, while LLGR is meant for longer failures and the neighbors need to be informed.

The ports on PE-1 are re-enabled and the routes are re-advertised followed by an EOR per AFI/SAFI, which terminates the LLGR phase.

```
# on PE-1:
configure
  port 1/1/1
    no shutdown
  exit
  port 1/1/2
    no shutdown
  exit
```

LLGR enabled on eBGP session

On PE-2 and PE-4, GR and LLGR are enabled for the "eBGP" group, as follows:

```
# on PE-2, PE-4:
configure
  router
    bgp
      group "eBGP"
        graceful-restart
          stale-routes-time 150
          enable-notification
          long-lived
          advertised-stale-time 3600
        exit
      exit
    exit
```

PE-2 will re-advertise the routes it sent to PE-4, but with well-known community **llgr-stale**. PE-4 was unaware of the GR phase; it only got involved in the LLGR phase. The following BGP update was sent by PE-2 to its eBGP peer PE-4 for the IPv4 address family:

```
339 2020/02/12 12:43:06.007 UTC MINOR: DEBUG #2001 Base Peer 1: 192.168.24.2
"Peer 1: 192.168.24.2: UPDATE
Peer 1: 192.168.24.2 - Send 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 < 64496 >
  Flag: 0x40 Type: 3 Len: 4 Nexthop: 192.168.24.1
  Flag: 0xc0 Type: 8 Len: 4 Community:
    llgr-stale
  NLRI: Length = 5
    172.16.225.0/30
"
```

PE-4 does not mark the route as stale in the way that PE-2 does; the BGP route does not get the stale flag "x", as follows:

```
*A:PE-4# show router bgp routes
```

```

=====
BGP Router ID:192.0.2.4      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 172.16.225.0/30                        None      None
      192.168.24.1                          None      0
      64496                                   -
-----
Routes : 1
    
```

The detailed information for this route on PE-4 shows the community **llgr-stale**, but no LLGR-stale flag, as follows:

```

*A:PE-4# show router bgp routes detail
=====
BGP Router ID:192.0.2.4      AS:64500      Local AS:64500
=====
---snip---
=====
BGP IPv4 Routes
=====
Original Attributes

Network       : 172.16.225.0/30
Nexthop       : 192.168.24.1
Path Id       : None
From          : 192.168.24.1
Res. Protocol : LOCAL           Res. Metric   : 0
Res. Nexthop  : 192.168.24.1
Local Pref.   : n/a             Interface Name : int-PE-4-PE-2
---snip---
Community   : llgr-stale
Cluster      : No Cluster Members
Originator Id : None           Peer Router Id : 192.0.2.2
Fwd Class    : None           Priority       : None
Flags      : Used Valid Best IGP
Route Source : External
AS-Path      : 64496
---snip---
    
```

Per-AF LLGR

The following configuration on PE-2 enables GR for the same address families as before, while LLGR will only be applied for IPv4 FlowSpec and IPv6 FlowSpec, with different LLGR-stale times. The default LLGR-stale time—**advertised-stale-time**—is 86400 seconds, but **helper-override-stale-time 0** in the iBGP group context overrides the LLGR-stale time to a zero value for the iBGP group. For FlowSpec routes, the **advertised-stale-time** is set to a value of 20000 seconds. For IPv4 FlowSpec, the **helper-override-stale-time** is set to 2000 seconds; for IPv6 FlowSpec, it is set to 3000 seconds. The forwarding bit is only set

for non-forwarding AFs—**forwarding-bits-set non-fwd**—so it will be set for configuration routes, such as FlowSpec routes.

```
# on PE-2:
configure
router
  bgp
    group "iBGP"
      family ipv4 ipv6 vpn-ipv4 vpn-ipv6 l2-vpn flow-ipv4 flow-ipv6
      graceful-restart
        stale-routes-time 150
        enable-notification
        long-lived
          advertised-stale-time 86400          # default
          helper-override-stale-time 0
          family flow-ipv4
            advertised-stale-time 20000
            helper-override-stale-time 2000
          exit
          family flow-ipv6
            advertised-stale-time 20000
            helper-override-stale-time 3000
          exit
          forwarding-bits-set non-fwd
          no advertise-stale-to-all-neighbors # default
        exit
      exit
    peer-as 64496
    neighbor 192.0.2.1
  exit
```

With this configuration on PE-2, the LLGR phase will be reduced to zero seconds for all AFs except IPv4 FlowSpec and IPv6 FlowSpec, but the **helper-override-stale-time** is not advertised to the BGP peer; only the **advertised-stale-time** is advertised. The GR phase applies for all configured address families with the same timers. When the BGP configuration on PE-1 is preserved and LLGR is enabled for the same address families, the following command shows the GR information on PE-2 for peer PE-1.

```
*A:PE-2# show router bgp neighbor 192.0.2.1 graceful-restart

=====
BGP Neighbor 192.0.2.1 Graceful Restart
=====
Graceful Restart locally configured for peer: Enabled
GR Notification                               : Enabled
Peer's Graceful Restart feature               : Enabled
NLRI(s) that peer supports restart for       : ipv4 ipv6 vpn-ipv4 vpn-ipv6
                                                l2-vpn flow-ipv4 flow-ipv6
NLRI(s) that peer saved forwarding for       : ipv4 ipv6 vpn-ipv4 vpn-ipv6
                                                l2-vpn flow-ipv4 flow-ipv6
NLRI(s) that restart is negotiated for       : ipv4 ipv6 vpn-ipv4 vpn-ipv6
                                                l2-vpn flow-ipv4 flow-ipv6
NLRI(s) of received end-of-rib markers       : ipv4 ipv6 vpn-ipv4 vpn-ipv6
                                                l2-vpn flow-ipv4 flow-ipv6
NLRI(s) of all end-of-rib markers sent       : ipv4 ipv6 vpn-ipv4 vpn-ipv6
                                                l2-vpn flow-ipv4 flow-ipv6
NLRI(s) peer supports NOTIFICATION GR for    : ipv4 ipv6 vpn-ipv4 vpn-ipv6
                                                l2-vpn flow-ipv4 flow-ipv6
Restart time locally configured for peer     : 300 seconds
Restart time requested by the peer           : 300 seconds
Time until stale routes are deleted or
become long-lived stale                       : 150 seconds
Graceful restart status on the peer          : Not currently being helped
```

```

Long-Lived GR status on the peer      : Not currently being helped
Number of Restarts                    : 0
Last Restart at                      : Never
-----
LLGR Configuration                   : Enabled
Peer's LLGR feature                  : Enabled
NLRI(s) peer signaled LLGR for & stale time & F-bit
                                     : ipv4 : 3600 seconds (F)
                                     : ipv6 : 3600 seconds (F)
                                     : vpn-ipv4 : 3600 seconds (F)
                                     : vpn-ipv6 : 3600 seconds (F)
                                     : l2-vpn : 3600 seconds (F)
                                     : flow-ipv4 : 3600 seconds (F)
                                     : flow-ipv6 : 3600 seconds (F)
                                     NLRI(s) LLGR negotiated for and stale time :
ipv4 : 0 seconds
                                     ipv6 : 0 seconds
                                     vpn-ipv4 : 0 seconds
                                     vpn-ipv6 : 0 seconds
                                     l2-vpn : 0 seconds
                                     flow-ipv4 : 2000 seconds
                                     flow-ipv6 : 3000 seconds
                                     LLGR Restart time overridden for the peer : n/a
                                     NLRI(s) LLGR advertised & stale time & F-bit:
ipv4 : 86400 seconds
                                     ipv6 : 86400 seconds
                                     vpn-ipv4 : 86400 seconds
                                     vpn-ipv6 : 86400 seconds
                                     l2-vpn : 86400 seconds(F)
                                     flow-ipv4 : 20000 seconds(F)
                                     flow-ipv6 : 20000 seconds(F)
=====

```

LLGR is enabled on PE-1 and PE-2. BGP peer PE-1 has signaled LLGR-stale times of 3600 seconds for the supported AFI/SAFIs. PE-2 had advertised the default LLGR-stale time of 86400 seconds for all supported AFI/SAFIs except for the FlowSpec AFI/SAFIs, where the LLGR-stale time is 20000 seconds. On PE-2, the F-bit is only set for the non-forwarding routes; in this case, L2 VPN, IPv4 FlowSpec, and IPv6 FlowSpec.

The **helper-override-stale-time** is not advertised to the BGP peer, but considered for the local LLGR behavior (in bold). Only the FlowSpec AFs get a non-zero LLGR-stale time: 2000 seconds for IPv4 FlowSpec; 3000 seconds for IPv6 FlowSpec.

The following GR/LLGR information on peer PE-1 shows the advertised LLGR-stale time, not the **helper-override-stale-time** configured on PE-2.

```

*A:PE-1# show router bgp neighbor 192.0.2.2 graceful-restart
=====
BGP Neighbor 192.0.2.2 Graceful Restart
=====
Graceful Restart locally configured for peer: Enabled
GR Notification                          : Enabled
Peer's Graceful Restart feature          : Enabled
NLRI(s) that peer supports restart for  : ipv4 ipv6 vpn-ipv4 vpn-ipv6
                                           l2-vpn flow-ipv4 flow-ipv6
NLRI(s) that peer saved forwarding for   : l2-vpn flow-ipv4 flow-ipv6
NLRI(s) that restart is negotiated for   : ipv4 ipv6 vpn-ipv4 vpn-ipv6
                                           l2-vpn flow-ipv4 flow-ipv6
NLRI(s) of received end-of-rib markers   : ipv4 ipv6 vpn-ipv4 vpn-ipv6
                                           l2-vpn flow-ipv4 flow-ipv6
NLRI(s) of all end-of-rib markers sent   : ipv4 ipv6 vpn-ipv4 vpn-ipv6

```

```

NLRI(s) peer supports NOTIFICATION GR for      : l2-vpn flow-ipv4 flow-ipv6
                                                : ipv4 ipv6 vpn-ipv4 vpn-ipv6
                                                : l2-vpn flow-ipv4 flow-ipv6
Restart time locally configured for peer       : 300 seconds
Restart time requested by the peer             : 300 seconds
Time until stale routes are deleted or
become long-lived stale                       : 150 seconds
Graceful restart status on the peer           : Restart completed
Long-Lived GR status on the peer              : Restart completed
Number of Restarts                             : 4
Last Restart at                               : 02/12/2020 14:21:25
-----
LLGR Configuration                           : Enabled
Peer's LLGR feature                          : Enabled
NLRI(s) peer signaled LLGR for & stale time
& F-bit                                       : ipv4 : 86400 seconds
                                                : ipv6 : 86400 seconds
                                                : vpn-ipv4 : 86400 seconds
                                                : vpn-ipv6 : 86400 seconds
                                                : l2-vpn : 86400 seconds (F)
                                                : flow-ipv4 : 20000 seconds (F)
                                                : flow-ipv6 : 20000 seconds (F)
NLRI(s) LLGR negotiated for and stale time    : ipv4 : 86400 seconds
                                                : ipv6 : 86400 seconds
                                                : vpn-ipv4 : 86400 seconds
                                                : vpn-ipv6 : 86400 seconds
                                                : l2-vpn : 86400 seconds
                                                : flow-ipv4 : 20000 seconds
                                                : flow-ipv6 : 20000 seconds
LLGR Restart time overridden for the peer     : n/a
NLRI(s) LLGR advertised & stale time & F-bit: ipv4 : 3600 seconds(F)
                                                : ipv6 : 3600 seconds(F)
                                                : vpn-ipv4 : 3600 seconds(F)
                                                : vpn-ipv6 : 3600 seconds(F)
                                                : l2-vpn : 3600 seconds(F)
                                                : flow-ipv4 : 3600 seconds(F)
                                                : flow-ipv6 : 3600 seconds(F)
=====

```

With this configuration, only the FlowSpec routes can get the LLGR-stale flag. No LLGR phase will start for the other AFs, so the stale routes of those AFs will be withdrawn when the GR phase ends.

It is possible to override the GR restart time to enter the LLGR phase immediately without going through the GR phase, as follows, on PE-2:

```

# on PE-2:
configure
router
  bgp
    group iBGP
      graceful-restart
        long-lived
        helper-override-restart-time 0

```

When the BGP session goes down on PE-2, the GR phase is omitted because the restart time of zero seconds expires instantly, so the LLGR phase starts immediately, as follows.

```
*A:PE-2# show router bgp neighbor 192.0.2.1 graceful-restart
```

```
=====
BGP Neighbor 192.0.2.1 Graceful Restart
=====
```

```
Graceful Restart locally configured for peer: Enabled
GR Notification                : Enabled
Peer's Graceful Restart feature : Enabled
---snip---
Restart time locally configured for peer : 300 seconds
---snip---
Graceful restart status on the peer      : Restart completed
Long-Lived GR status on the peer       : Rcvd restart request
---snip---

-----
---snip---

LLGR Restart time overridden for the peer : 0
---snip---
```

When LLGR phase starts immediately, only the FlowSpec address families will be protected while all routes of the other AFs are withdrawn. The FlowSpec routes get the LLGR-stale flag and route updates to eBGP peer PE-4 will get the LLGR-stale community, as follows:

```
*A:PE-2# show router bgp routes flow-ipv4 hunt
=====
---snip---
-----
RIB In Entries
-----
---snip---
From          : 192.0.2.1
---snip---
Flags         : Used Valid Best IGP LLgrStale
---snip---

-----
RIB Out Entries
-----
---snip---
To            : 192.168.24.2
---snip---
Community   : llgr-stale rate-limit: 0 kbps
---snip---
```

Conclusion

Graceful restart helps avoid withdrawing BGP routes immediately when the BGP session goes down. Routes that were received from the failed router are marked as stale, but remain in use. When the BGP session is down for a longer time, such as hours or days, LLGR can take over when the GR ends, possibly only for a subset of AFI/SAFIs. In the LLGR phase, the LLGR-stale routes are deprefferenced, but if they remain best and valid, they can be re-advertised to the BGP peers as LLGR-stale.

BGP Monitoring Protocol Basics

This chapter provides information about BGP Monitoring Protocol Basics.

Topics in this chapter include:

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

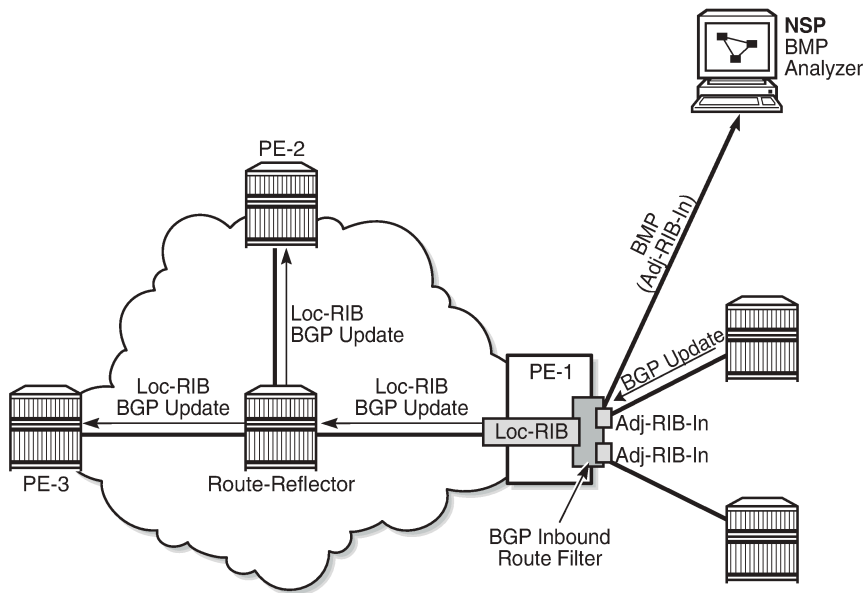
Applicability

The information and configuration in this chapter are based on SR OS Release 16.0.R2. BGP Monitoring Protocol (BMP) support was introduced in SR OS Release 16.0.R1 for unicast IPv4/IPv6, VPN IPv4/IPv6, and labeled IPv4/IPv6. SR OS Release 16.0.R4 provides an additional six address families: EVPN, L2VPN, multicast IPv4/IPv6, multicast VPN IPv4/IPv6.

Overview

The BGP Monitoring Protocol (BMP) is a unidirectional protocol for providers to monitor the behavior of BGP on their routers. A router communicates information about one or more BGP sessions to a BMP station, also known as a BMP collector. A router sends information in BMP messages to a BMP station. A BMP station never sends any messages to a router. BMP is described in detail in RFC 7854. [Figure 51: BMP Operational Overview](#) shows an operational overview of BMP.

Figure 51: BMP Operational Overview



27981

Table 3: [BMP Message Types](#) lists the BMP message types that are defined in RFC 7854.

Table 3: *BMP Message Types*

BMP Message Type	Description
0	Route monitoring
1	Statistics report
2	Peer down notification
3	Peer up notification
4	Initiation message
5	Termination message
6	Route mirroring message

A BMP station (or BMP collector) typically is a dedicated server running network management or network controller software. Current examples of free and open-source BMP station software are OpenBMP and Open Daylight. Nokia has commercial BMP station support available through the Network Services Platform (NSP) controller. The simple operations and packet format of BMP resulted in many providers having created their own proprietary BMP-collector software.

BMP allows a router to report different types of information. A router can:

- send BMP messages with notifications when neighbors go into or out of Established state (for example, the peer goes "up" or "down"). These notifications are called BMP peer-up and peer-down messages.

- periodically send statistical information about one or more neighbors. This information consists of several counters; for example, how many routes are received from a specific neighbor, or how many of those routes were rejected or accepted because of an ingress policy. Other counters report how many errors were encountered; for example, AS-path loops, duplicate prefixes, withdrawals received, and so on.
- report the exact routes that were received from a neighbor. This action is called route monitoring. To do this, a router first re-encapsulates a BGP route into its original BGP update message, then encapsulates that BGP update message within a BMP route monitoring message to send it to the BMP station.



Note:

BMP on an SR OS router will only report information about routes that were received from a neighbor, which is the standard BMP behavior documented in RFC 7854. BMP will also report upon routes leaked or redistributed into the BGP RIB. A limitation of RFC 7854 is that BMP does not monitor routes sent toward a specific BGP neighbor. Nokia supports RFC 7854, so does not support monitoring of routes that were sent toward a BGP neighbor.

Configuration

Basic configuration of BMP

There are two main steps to enable BMP monitoring on an SR OS router:

1. Configure a BMP station. This configuration identifies the target to which BMP information will be sent.
2. Configure one or more BGP neighbors. These are the BGP peering sessions that will be monitored by BMP and the configured BMP station.

Configuring a BMP station

BMP stations and associated parameters are configured in global configuration mode. This allows the BMP station to reside either within the base router instance, or in a VPRN routing instance. The Nokia BMP implementation can monitor BGP peers in a base Internet service or in a VPRN service instance.

BMP will initiate a separate TCP session for each VPRN BGP instance monitored. The BMP router will use a different source TCP port number toward each configured TCP destination port number of the BMP station. For example, if there are four VPRN services configured in addition to the base router instance, the BMP router will instantiate five TCP sessions between the BMP router and the BMP station (one TCP session to monitor the base router instance, and four TCP sessions to monitor the VPRN services).

SR OS supports the configuration of up to eight BMP stations. To configure a BMP station, use the following command syntax:

```
*A:Dut-C# configure bmp station Antwerp create
```

This configuration example creates a BMP station with the name "Antwerp". This name must be used when configuring BGP peers to be monitored by this station. The name can also be used in **show router bmp** commands.

The next step is to configure (at a minimum) the IP address and the TCP destination port the BMP station is listening to. These parameters inform the BMP router where to reach the BMP station. BMP does not use a well-known port number; a provider can select any TCP port number. BMP sessions from an SR OS router can run over either TCP IPv4 or TCP IPv6.

The following configures the IP address 100.1.1.10 and port number 5000 of the BMP station:

```
configure
  bmp
    station "Antwerp" create
      connection
        station-address 100.1.1.10 port 5000
      exit
    exit
  exit
```

This configuration example creates a BMP station that can be used to monitor one or more BGP peers. Next, configure the BGP peers to be monitored by this station.

Assigning the BGP peers to be monitored

To configure one or more BGP neighbors to be monitored by the BMP station, first configure the **monitor** command in the **bgp** context or one of its subcontexts. This command can be configured at the BGP instance level, at the BGP group level, or at the neighbor level.

In the following example, monitoring is enabled (no shutdown) for all BGP peers defined in the **bgp** context, where the BMP reporting goes to BMP station *Antwerp*.

```
configure
  router
    bgp
      monitor
        station Antwerp
        no shutdown
      exit
      group internal-1
        ---snip---
      Exit
      group internal-2
        ---snip---
      exit
    exit
```

By default, BMP, including each individually configured station, is in the administrative shutdown state. To allow BMP to start the BMP sessions, administratively enable the BMP station:

```
configure
  bmp
    no shutdown
    station Antwerp
      no shutdown
    exit
  exit
```

All peers in the BGP instance of the base router are now monitored by station "Antwerp". At this stage, the router will only send BMP peer-up and peer-down messages to the BMP station. To send additional

information (such as periodic statistics messages, or to report incoming BGP routes) requires explicit configuration.

Configuring periodic statistics messages

Enabling periodic statistics messages is done under the **configure bmp station** command. The command to enable periodic statistics is **stats-report-interval <seconds>**:

```
configure
  bmp
    station Antwerp
      stats-report-interval 600
    exit
  exit
```

This configuration example will cause the router to send statistics messages for each monitored peer to the BMP station every 10 minutes (600 seconds).

Verifying that the BMP session between router and BMP station works

To display the state of a BMP session to a BMP station, use the **show router bmp station <station-name>** command:

```
show router bmp station Antwerp
```

The output of the **show** command for BMP station Antwerp is as follows:

```
*A:Dut-C# show router bmp station "Antwerp"

=====
BMP Station "Antwerp" (monitoring router "Base")
=====
Admin State      : enabled                Global BMP State : enabled
Station Address  : 100.1.1.10             Station Port     : 5000
Via Router       : Base
Stats Report     : 30 seconds
Connect Interval: 5 seconds              Local Routes     : not reporting
Reported families: ipv4

Session State    : ESTABLISHED          Last State Change: 08/29/2018 13:23:19
Reason Last Down: admin shutdown         Last Msg Sent    : 08/29/2018 13:23:19
Local Address    : 100.1.1.3             Local Port       : 51446
Routes Timer     : 2 seconds left         Stats Timer      : 3 seconds left
Connect Timer    : not running           Monitored Peers  : 0 of 1
Initiation Msgs : 1                     Goodbye Msgs     : 0
Peer Up Msgs    : 0                     Peer Down Msgs   : 0
Route Report Msgs: 0                     Stat Report Msgs : 0
Bytes Sent      : 276                    Output Queue     : 0/5
=====
*A:Dut-C#
```

The output consists of two blocks of information.

- The first block shows configuration information about this specific BMP station.

- The second block shows dynamic information about the current BMP session from the router instance to the BMP station.

Verify that the Session State is "ESTABLISHED".

Configuring BMP route monitoring

Configuring BMP route monitoring requires explicit configuration under the **monitor** command in the BGP instance context.

It is possible to configure BMP to report pre-policy routes, or post-policy routes, or both. Pre-policy routes are incoming routes as they were before applying any ingress policy. Post-policy routes are resulting routes in the Adj-RIB-In and reflect the routes after applying any BGP ingress policy.

Configuring BMP to report both pre- and post-policy routes will result in the doubling of BMP messages to the BMP station. This is because the router will send a route-monitor message for each pre-policy route, and for each post-policy route. This doubles the amount of resources consumed by BMP (such as bandwidth consumed on the link between the router and the BMP station, and CPU usage). The impact of enabling BMP route monitoring on the router CPU is similar to adding a BGP neighbor.

To configure route monitoring, use the **route-monitoring [pre-policy] [post-policy]** command in the monitor configuration mode in the BGP configuration context:

```
configure
router
  bgp
    monitor
      station Antwerp
      route-monitoring pre-policy
      no shutdown
    exit
  exit
```

The BMP route monitoring is enabled within the context where the **monitor station** command is configured: in the general **bgp** context, the **group** context, or the **neighbor** context. With this configuration, the BMP router will start sending route monitoring messages for every route received from every neighbor in the base router BGP instance. This can be verified via the **show router bmp station <station-name>** command, which displays the counter for "Route Report Msgs:".

Advanced BMP configuration options

The BMP configuration can be fully customized. The following sections describe some additional configuration options.

Configuring route monitoring for different address families

When route monitoring is enabled, by default the BMP router will only report received IPv4 routes to the BMP station. This aligns with the default BGP behavior, where only unicast IPv4 is enabled when configuring a neighbor under BGP. To enable route monitoring for additional BGP address families, additional explicit configuration is required. The additional address families are available and can be configured under the **configure bmp station** command context, as follows:

```
configure
```

```

bmp
  station Antwerp
    family
  exit
exit

```

In SR OS Release 16.0.R1, a Nokia BMP router supports route monitoring of six address families:

- unicast IPv4, unicast IPv6
- VPN-IPv4, VPN-IPv6
- label-IPv4, label-IPv6

SR OS Release 16.0.R4 provides an additional six address families:

- EVPN
- L2VPN
- mcast-IPv4, mcast-IPv6
- mcast-VPN-IPv4, mcast-VPN-IPv6

Configuring monitoring of locally generated routes

RFC 7854 BMP reports only the routes in the Adj-RIB-In that were received from monitored neighbors. However, the BGP-RIB can hold more routes than those routes BGP has learned from neighbors. These locally generated routes are called imported or leaked routes.

Imported routes are learned via redistributing routes into BGP from external sources, like static, connected, IS-IS, or OSPF. Leaked routes are BGP routes from other BGP service instances that are leaked into the base router BGP.

To configure the Nokia BMP router to extend route reporting and report these imported and leaked routes to a configured BMP station, configure the **report-local-routes** command under the BMP station:

```

configure
  bmp
    station Antwerp
      report-local-routes
    exit
exit

```

Configuring the frequency of router statistics reports

When periodic statistics are enabled, the router will send all the statistics as described in RFC 7854, section 4.8, except for statistic number 13 (number of duplicate update messages received).

The Nokia BMP router-supported statistics are:

- 0 - number of prefixes rejected by inbound policy
- 1 - number of duplicate prefix advertisements received
- 2 - number of duplicate withdraws received
- 3 - number of received updates invalidated due to cluster-list loop
- 4 - number of received updates invalidated due to AS-path loop

- 5 - number of received updates invalidated due to originator-id
- 6 - number of received updates invalidated due to as-confed loop
- 7 - total number of routes in Adj-RIB-In (all families)
- 8 - total number of routes in loc-RIB (all families)
- 9 - number of routes per address family in Adj-RIB-In (see Note)
- 10 - number of routes per address family in loc-RIB (see Note)
- 11 - number of updates subjected to treat-as-withdraw
- 12 - number of prefixes subjected to treat-as-withdraw
- 13 - not supported/reported by SR OS (number of duplicate update messages received)



Note:

These two statistics are per address family. The address family is specified as a BGP AFI/SAFI pair. Regardless of what families are configured or supported for route monitoring, a router will report the statistics of all address families that were negotiated with the neighbor.

The values shown in the preceding counters are the same values that are shown by the **show router <vrid> bgp neighbor <ip-addr> [detail]** command.

Customizing the TCP connection to the BMP station

BMP uses TCP sessions to send BMP messages to the BMP station. It is possible to customize the TCP-session settings using several configuration options. These options are under the **configure bmp station <name> connection** command context.

Setting the local address of the TCP session

For increased operational security, BMP collectors might restrict accepting BMP sessions from unknown routers. It is important to have a configuration option to force a BMP router to accept specific IP addresses. To enforce the source address of a BMP session, the provider can configure the "local-address <ip-address>".

A Nokia router BMP session can be over an IPv4 or IPv6 TCP session. The source IP address used by the BMP router can be configured using the **local-address** command. The local address can be an IPv4 or an IPv6 address. The address family (IPv4 or IPv6) must match the address family of the IP address configured in the **station-address <ip-address> port <portnr>** command:

```
configure
  bmp
    station "Antwerp"
      connection
        station-address 100:200:300::1 port 5000
        local-address 100:200:300::2
      exit
    exit
  exit
```

Setting the routing context of the BMP session

A Nokia router allows a provider to configure multiple virtual router instances.

The base router is such a virtual router. Each VRPN instance is also a virtual router.

A Nokia BMP router allows a provider to monitor a BGP VPRN session while the TCP connection of the BMP session is configured in another VPRN instance.

This functionality allows the provider to let a single BMP station connection, within a specific VPRN instance, monitor BGP sessions and instances resident in other virtual routers.

The TCP connection of a BMP session is by default active in the base router. This can be changed by adding additional **vprn** context configuration when configuring a BMP station, as follows:

```
configure
  bmp
    station "Antwerp"
      connection
        router service-name vprn-22
      exit
    exit
```

Connect-retry command

When a router initiates a BMP session, it will try to establish the TCP connection to the BMP station. If this attempt fails, the router will wait a short while, then retry to bring up the connection. The time between two such attempts increases over time. The first attempt waits 3 seconds. After each failed attempt, the waiting time doubles (exponential increase). The maximum time to wait between two attempts is by default 2 minutes (120 seconds). This maximum waiting time is configurable, as follows:

```
configure
  bmp
    station "Antwerp"
      connection
        connect-retry 600
      exit
    exit
```

This configuration example will set the maximum waiting time between two connection attempts to 10 minutes (600 seconds).

TCP keepalives

BMP does not have any mechanism to detect the liveness of a BMP station. As the protocol is unidirectional, a router will not detect that a BMP station is down or unreachable, until it tries to send data to the station. During normal operation, the TCP layer will inform the BMP layer of an error when BMP tries to send a message to a BMP station that is down or unreachable. After discovering the TCP error, BMP will close the BMP session and try to re-establish a new session. However, when the BMP router has nothing to send to the unreachable BMP station, the station is not detected that easily.

Providers might need to detect a BMP failure even quicker. To do that, providers have the option to configure "TCP keepalives" on the BMP session. TCP keepalives are a feature of the TCP protocol. TCP keepalives are used to ensure the liveness of a TCP connection, even when no data is sent.

BMP on a Nokia router can use TCP keepalives. No special support is needed on the BMP station or host operating system because this functionality is a basic operation of the TCP session.

TCP keepalives are disabled by default. To enable a BMP session with TCP keepalives, configure:

```
configure
  bmp
    station "Antwerp"
      connection
        tcp-keepalive
          no shutdown
        exit
      exit
    exit
  exit
```

The default operational values of TCP keepalives on a BMP session are:

- keep-idle (sometimes called keep-wait) 600 seconds
- keep-interval 15 seconds
- keep-count 4 times

A provider can change these values. Configuring more aggressive values-tuning values for faster convergence-will have a slight impact on CPU and bandwidth usage. Configuring less aggressive values lowers the risk of false positives. For normal BMP operation, the default values are a good starting point. The following is an example if a provider wants to use non-default TCP keepalive values.

```
configure
  bmp
    station "Antwerp"
      connection
        tcp-keepalive
          keep-count 5
          keep-idle 300
          keep-interval 10
          no shutdown
        exit
      exit
    exit
  exit
```

Conclusion

In this chapter, the basic operation of Nokia BMP technology is described. The BMP implementation on a Nokia router is fully dual-stack IPv4/IPv6 aware and supports the monitoring of active BGP neighbor state (up or down), the BGP pre- and post-policy routes received, and a set of associated statistics for the BGP Adj-RIB-In and RIB-IN.

Usually, the impact upon the router performance for each configured BMP station is similar to adding a BGP neighbor. The Nokia BMP implementation supports the monitoring of twelve address families (unicast IPv4/IPv6, VPN IPv4/IPv6, label IPv4/IPv6, EVPN, L2VPN, mcast-IPv4/IPv6, mcast-VPN-IPv4/IPv6) in SR OS Release 16.0.R4, and later.

The Nokia BMP implementation can use TCP timers to detect unreachable BMP collectors. There is support for monitoring BGP neighbors in the base router or in a VPRN instance and support for BMP collectors located in the GRT or in any other VPRN service instance.

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 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 **multi-path** 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 **multi-path** 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:

```
*A:PE-5# configure router bgp multi-path ?
[no] ipv4          - Configure ipv4 multi-path maximum-paths
[no] ipv6          - Configure ipv6 multi-path maximum-paths
[no] label-ipv4    - Configure label-ipv4 multi-path maximum-paths
[no] label-ipv6    - Configure label-ipv6 multi-path maximum-paths
[no] maximum-paths - Configure multi-path maximum-paths
```

maximum-paths | ipv4 | ipv6 | label-ipv4 | label-ipv6 max-paths [ebgp ebgp-max-paths] [ibgp ibgp-max-paths] [restrict {same-neighbor-as | exact-as-path}] [unequal-cost]

```
*A:PE-5# configure router bgp multi-path maximum-paths ?
- maximum-paths <max-paths> [ebgp <ebgp-max-paths>] [ibgp <ibgp-max-paths>] [restrict
{same-neighbor-as|exact-as-path}] [unequal-cost]
- no maximum-paths

<max-paths>          : [1..64]
<ebgp-max-paths>    : [1..64]
<ibgp-max-paths>    : [1..64]
```

- multi-path configuration per address family (**ipv4 | ipv6 | label-ipv4 | label-ipv6**) overrides the generic **maximum-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.

```
*A:PE-5# configure router bgp best-path-selection ?
- best-path-selection

[no] always-compare* - Determine how the Multi-Exit Discriminator (MED) path attribute is
used in the BGP route selection process
[no] as-path-ignore - Determine whether the AS Path is used in determining the best BGP
route
[no] compare-origin* - Enable/Disable compare validation state
[no] d-path-length-* - Enable/disable d-path-length-ignore
[no] deterministic-* - Enable/Disable deterministic Multi-Exit Discriminator
[no] ebgp-ibgp-equal - Determine whether EBGP and IBGP learned paths are considered equal
[no] ignore-nh-metr* - Enable/Disable ignore next-hop metric
[no] ignore-router-* - Enable/Disable ignore router-id
[no] origin-invalid* - Enable/Disable origin invalid unusable routes.
```



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:

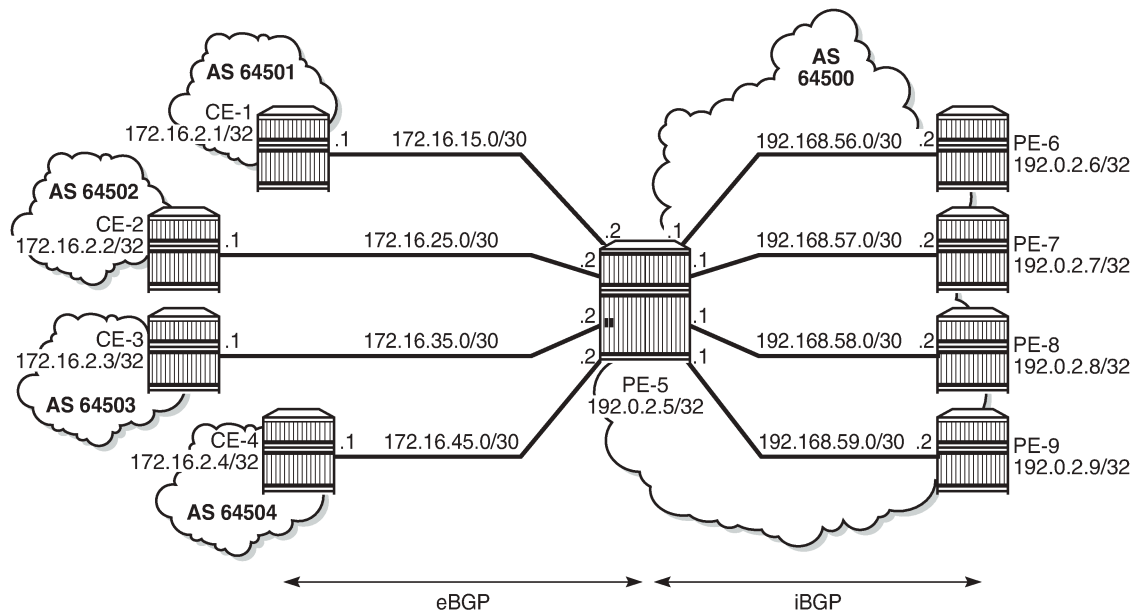
- If the best path came from a neighbor marked as multipath-eligible, then only paths marked as multipath-eligible are candidates for the BGP multipath and the best N are chosen for installation as ECMP next-hops.
- If none of the paths from the set of all possible multipaths came from a neighbor marked 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 52: 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 52: 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
  autonomous-system 64501
  bgp
    split-horizon
    group "eBGP"
      export "export-bgp"
      peer-as 64500
      neighbor 172.16.15.2
    exit
  exit
```

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
  autonomous-system 64500
  bgp
    split-horizon
    group "eBGP"
      neighbor 172.16.15.1
      peer-as 64501
    exit
      neighbor 172.16.25.1
      peer-as 64502
    exit
      neighbor 172.16.35.1
      peer-as 64503
    exit
      neighbor 172.16.45.1
      peer-as 64504
    exit
  exit
  group "iBGP"
    peer-as 64500
    neighbor 192.0.2.6
  exit
    neighbor 192.0.2.7
  exit
    neighbor 192.0.2.8
  exit
    neighbor 192.0.2.9
  exit
  exit
exit
```

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
  bgp
    multi-path
      maximum-paths 8 ebgp 2 ibgp 3
    exit
```

It is mandatory to specify a maximum for BGP multipaths, as follows, but that is overruled by the individual limits for eBGP and iBGP. It is optional to configure limits for eBGP and iBGP.

```
*A:PE-5# configure router bgp multi-path maximum-paths ebgp 2 ibgp 3
                                     ^
Error: Missing parameter
```

It is allowed to specify a lower value for **maximum-paths** than for either eBGP or iBGP because the configured number of paths for eBGP and iBGP overrule the maximum-paths limitation, as follows:

```
# on PE-5:
configure
router
  bgp
    multi-path
      maximum-paths 1 ebgp 2 ibgp 3
    exit
```

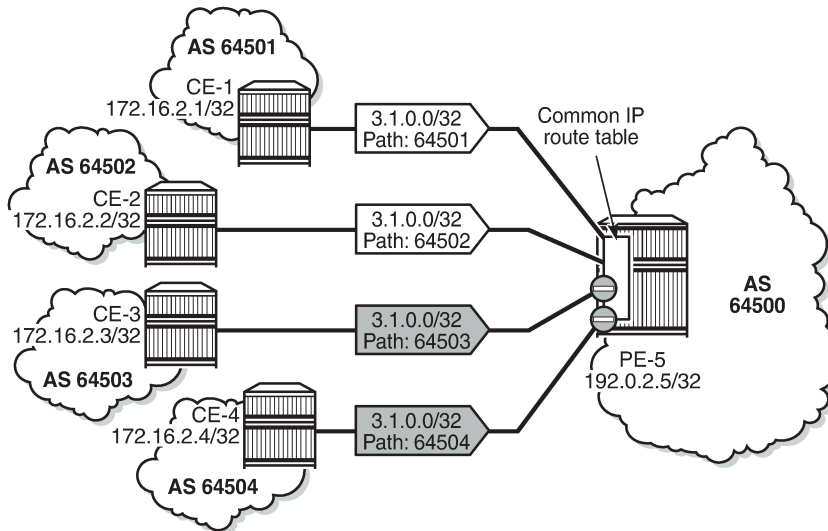
With this configuration, regardless of the value of maximum-paths, there can be two eBGP routes for the same prefix and three iBGP routes for the same prefix. If the best route is eBGP, the *ebgp-max-paths* value is 2; if the best route is iBGP, the *ibgp-max-paths* value is 3. The value for maximum-paths (1) is never used when limits for both eBGP and iBGP are configured.

```
# on PE-5:
configure
router
  bgp
    multi-path
      maximum-paths 3 ebgp 2
    exit
```

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 53: BGP multipath with eBGP limit 2](#).

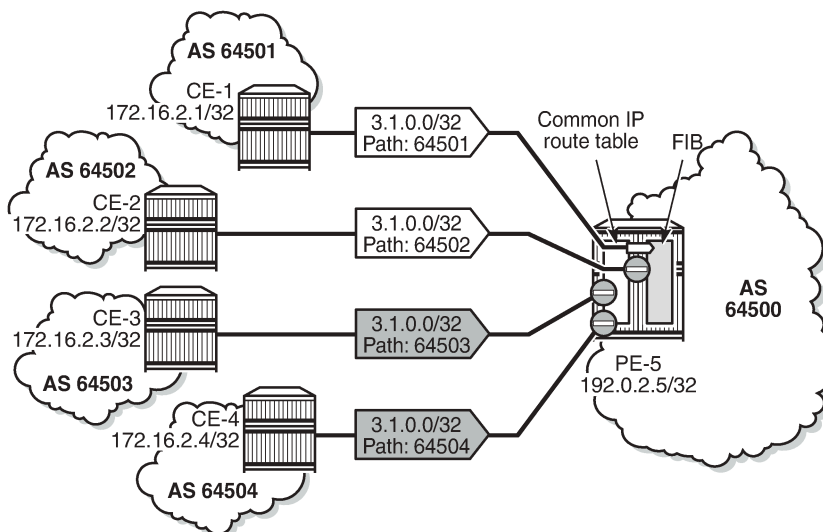
Figure 53: BGP multipath with eBGP limit 2



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 54: eBGP multipath with limit 2 and ECMP disabled](#).

Figure 54: 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: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
    ecmp 8
exit
```

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: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
-----
```

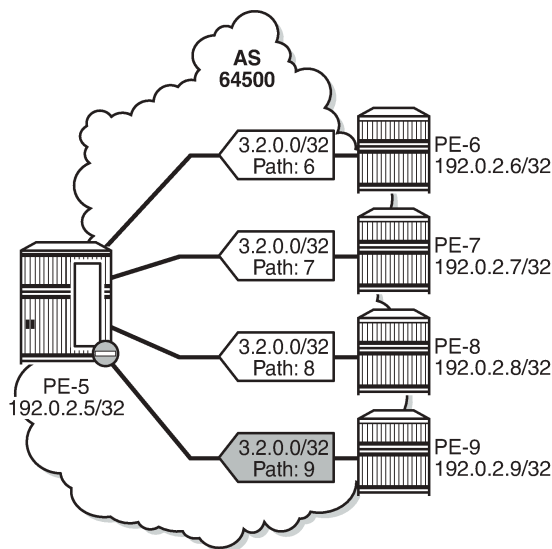
```

-----
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 55: BGP multipath with iBGP limit 3 and ECMP limit 8](#).

Figure 55: 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: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) As-Path	Path-Id	IGP Cost 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			
=====			

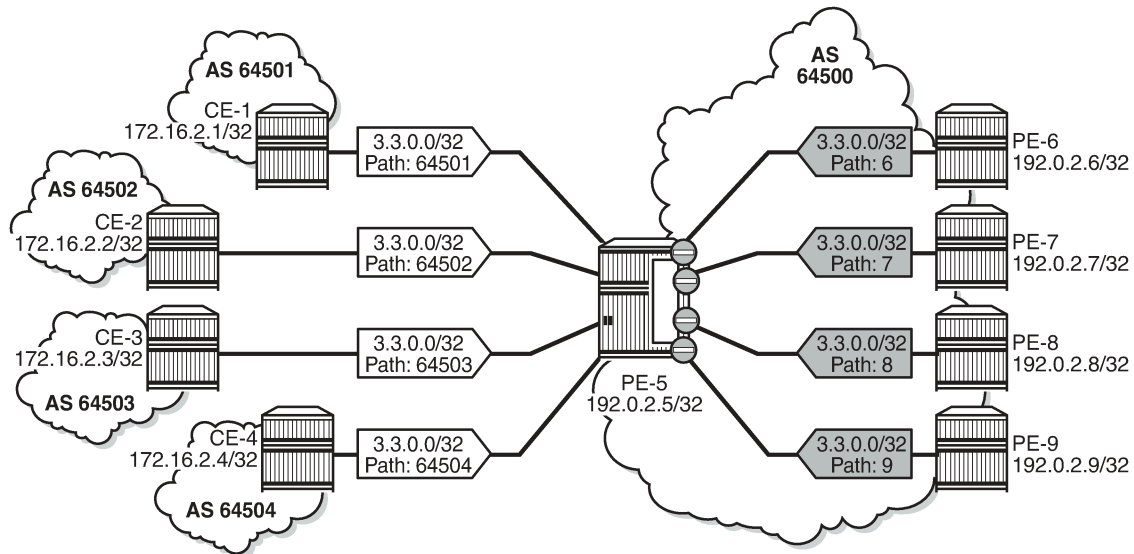
BGP multipath with equal eBGP and iBGP path treatment

It is optional to specify limits for eBGP and iBGP; an overall multipath limitation is sufficient, such as:

```
# on PE-5:
configure
router
  bgp
    multi-path
      maximum-paths 6
  exit
```

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 56: BGP multipath with limit 6 and eBGP preferred](#).

Figure 56: BGP multipath with limit 6 and eBGP preferred



26057

Only the four eBGP paths are used, as follows:

```
*A: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
  bgp
    best-path-selection
    ebgp-ibgp-equal ipv4
  exit

```

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 multi-path 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.

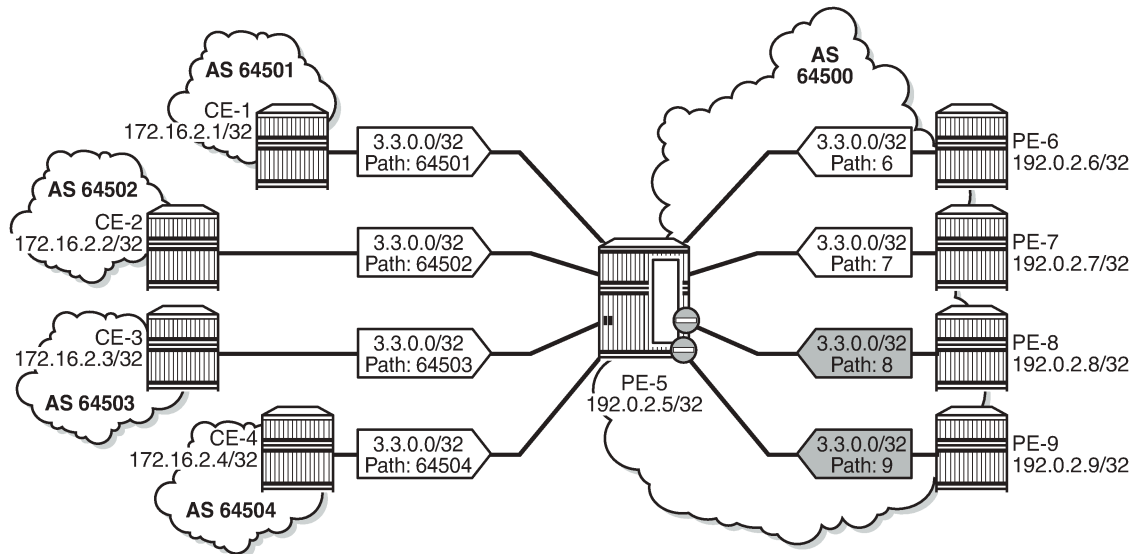
```

# on PE-5:
configure
router
  bgp
    multi-path
      maximum-paths 6 unequal-cost
  exit

```

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 57: BGP multipath with limit 6, eBGP equal to iBGP, and other path options identical](#).

Figure 57: 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: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
=====
```

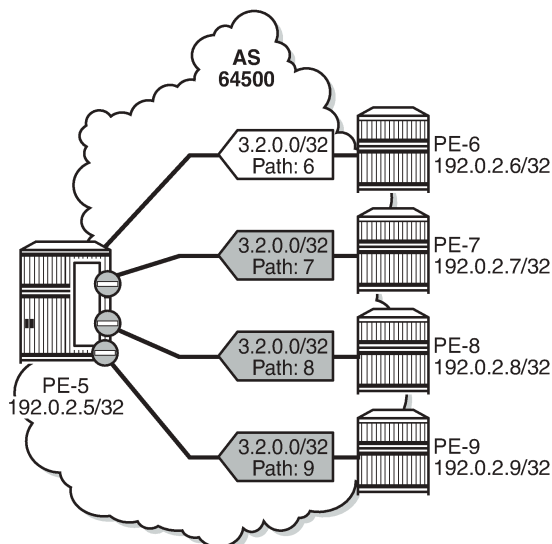
BGP multipath restricted to the same neighbor AS

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
  bgp
    multi-path
      maximum-paths 8 ebgp 2 ibgp 3 restrict same-neighbor-as
exit
```

Figure 58: 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 58: 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:PE-5# show router bgp routes 3.2.0.0/32
```

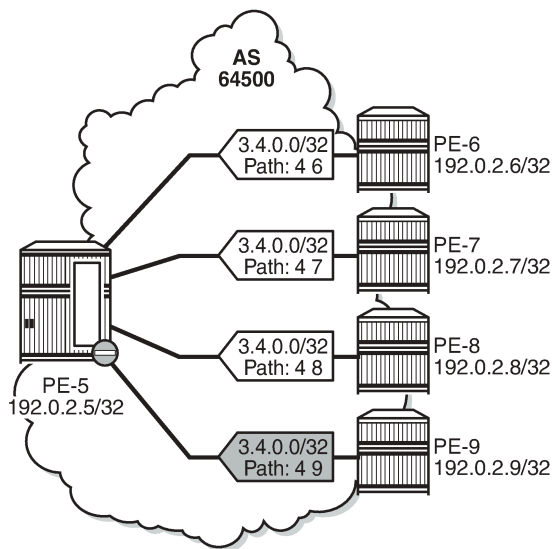
```

=====
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 59: 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 59: BGP multipath restricted to the same neighbor AS: AS paths with same length



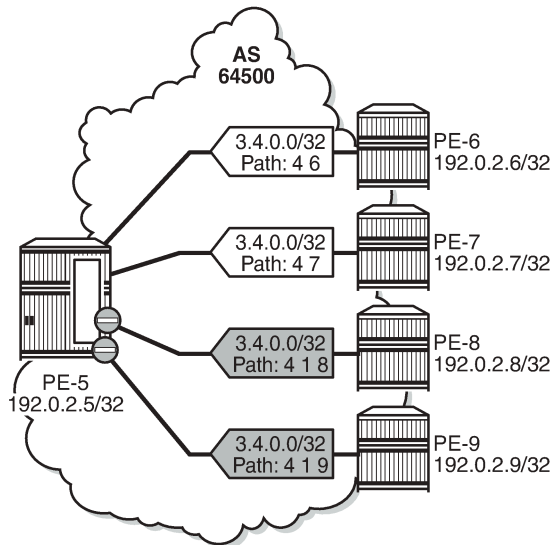
26060

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: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                Path-Id    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 8
*>i  3.4.0.0/32                100        None
      192.0.2.9              None       10
      4 9
-----
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 60: BGP multipath restricted to the same neighbor AS: AS paths of different lengths](#).

Figure 60: BGP multipath restricted to the same neighbor AS: AS paths of different lengths



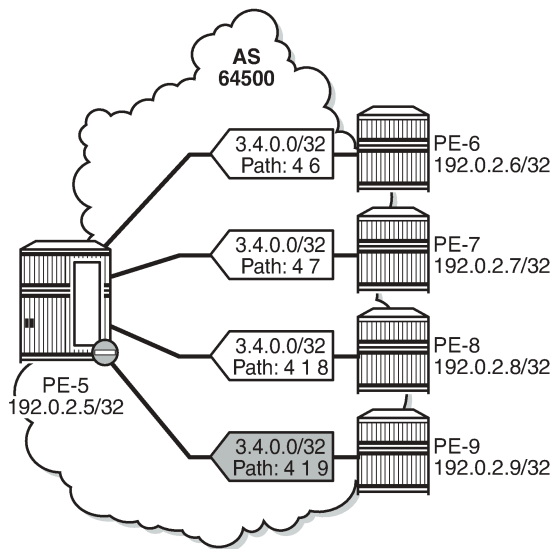
26061

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: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 61: BGP multipath restricted to the same neighbor AS: AS paths of different lengths, AS path ignored](#).

Figure 61: 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
  bgp
    best-path-selection
      as-path-ignore ipv4
  exit
```

Three of the four eligible BGP routes are used, as follows:

```
*A: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
  bgp
    best-path-selection
    no as-path-ignore
  exit

```

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:

```

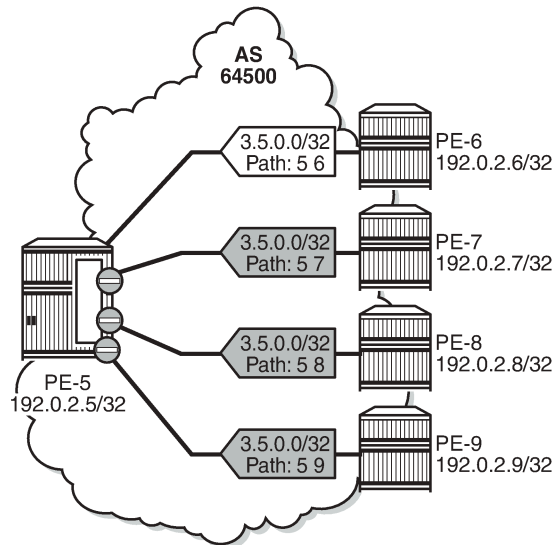
# on PE-5:
configure
router
  bgp
    multi-path
      maximum-paths 8 ebgp 2 ibgp 3 restrict exact-as-path
  exit

```

The four iBGP neighbors advertise prefixes 3.5.0.0/32 and 3.6.0.0/32 to PE-5, see [Figure 62: BGP multipath restricted to exact same AS. All AS paths are different.](#) and [Figure 63: 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 62: BGP multipath restricted to exact same AS. All AS paths are different.](#)

Figure 62: BGP multipath restricted to exact same AS. All AS paths are different.



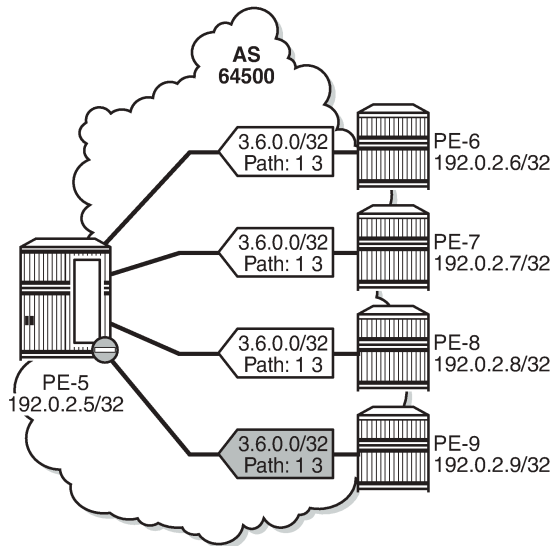
26063

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: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                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Label
-----
u*>i  3.5.0.0/32              100        None
      192.0.2.6              None       10
      5 6
*>i   3.5.0.0/32              100        None
      192.0.2.7              None       10
      5 7
*>i   3.5.0.0/32              100        None
      192.0.2.8              None       10
      5 8
*>i   3.5.0.0/32              100        None
      192.0.2.9              None       10
      5 9
-----
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 63: BGP multipath restricted to exact same AS. All AS paths are identical.](#)

Figure 63: 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: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                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  3.6.0.0/32                             100        None
      192.0.2.6                             None       10
      1 3
u*>i  3.6.0.0/32                             100        None
      192.0.2.7                             None       10
      1 3
u*>i  3.6.0.0/32                             100        None
      192.0.2.8                             None       10
      1 3
*>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
    multi-path
      ipv4 2 ibgp 2
      label-ipv4 3 ibgp 3
      label-ipv6 4 ibgp 4
      no maximum-paths
    exit
  group "iBGP"
    family ipv4 label-ipv4 label-ipv6
  exit
```

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: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

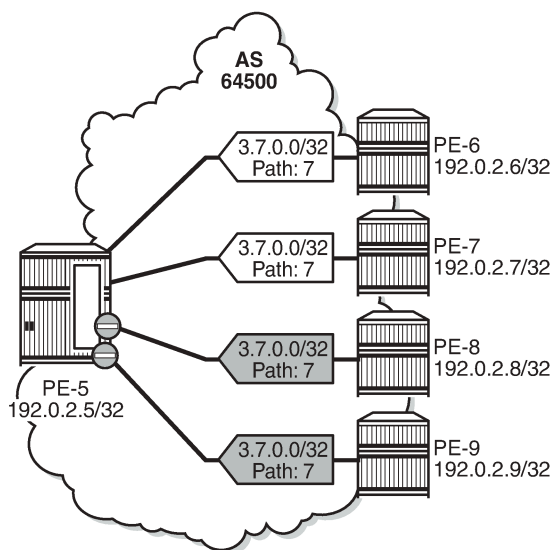
=====
BGP IPv4 Routes
=====
Flag Network                               LocalPref MED
      Nexthop (Router)                     Path-Id   IGP Cost
      As-Path                               Label
-----
u*>i 3.7.0.0/32                               100      None
      192.0.2.6                               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: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 64: BGP multipath for the IPv4 address family shows that two of the four received IPv4 routes are used.

Figure 64: 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: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      None
      192.0.2.8                       None      10        10
      8                               524282
*>i  3.8.0.0/32                       100      None      None
      192.0.2.9                       None      10        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:

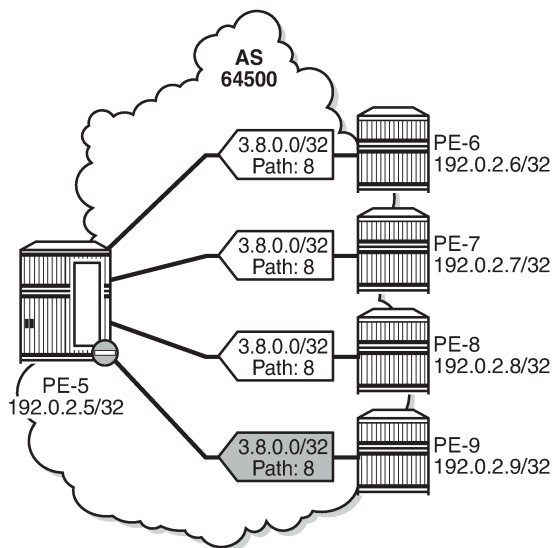
```

*A: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 65: BGP multipath for the label-IPv4 address family shows that three of the received label-IPv4 routes are used.

Figure 65: 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: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

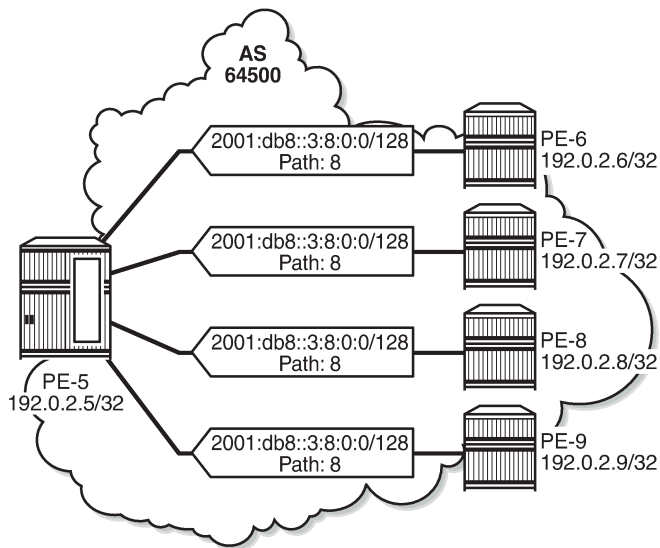
```

	Nexthop (Router) As-Path	Path-Id	IGP Cost Label
u*>i	2001:db8::3:8:0:0/128 ::ffff:192.0.2.6 8	100 None	None 10 2
u*>i	2001:db8::3:8:0:0/128 ::ffff:192.0.2.7 8	100 None	None 10 2
u*>i	2001:db8::3:8:0:0/128 ::ffff:192.0.2.8 8	100 None	None 10 2
u*>i	2001:db8::3:8:0:0/128 ::ffff:192.0.2.9 8	100 None	None 10 2

Routes : 4			
=====			

Figure 66: BGP multipath for the label-IPv6 address family shows that all four received label-IPv6 routes are used.

Figure 66: 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' (**no 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
  bgp
    multi-path
      ipv4 2 ibgp 2
      label-ipv4 3 ibgp 3
      label-ipv6 4 ibgp 4
      no maximum-paths
    exit
  group "iBGP"
    neighbor 192.0.2.6
      no multipath-eligible      # default
    exit
    neighbor 192.0.2.7
      multipath-eligible
    exit
    neighbor 192.0.2.8
      multipath-eligible
    exit
    neighbor 192.0.2.9
      no multipath-eligible      # default
    exit
  exit
```

When the best path originates from a neighbor that is configured as **no 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: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

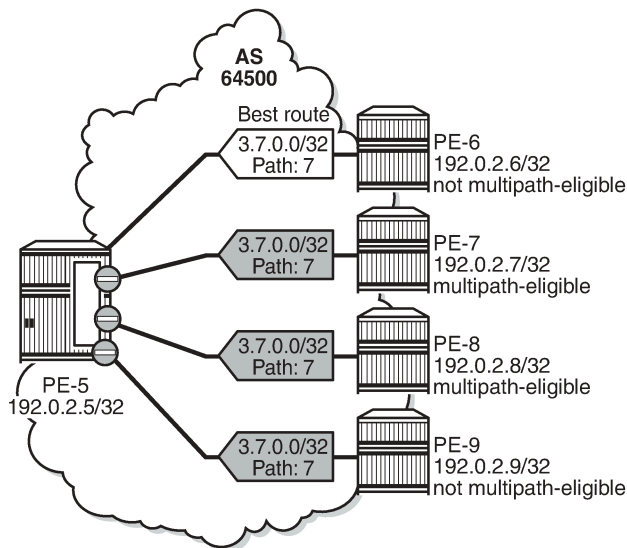
=====
BGP IPv4 Routes
=====
Flag  Network                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               Label
-----
u*>i  3.7.0.0/32                               100        None
      192.0.2.6                               None       10
      7
*>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: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 67: 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 67: 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
  bgp
    multi-path
      ipv4 2 ibgp 2
      label-ipv4 3 ibgp 3
      label-ipv6 4 ibgp 4
      no maximum-paths
    exit
  group "iBGP"
    neighbor 192.0.2.6
      multipath-eligible
    exit
    neighbor 192.0.2.7
      no multipath-eligible      # default
    exit
    neighbor 192.0.2.8
```

```

        multipath-eligible
    exit
    neighbor 192.0.2.9
        multipath-eligible
    exit
exit

```

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: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
*>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:

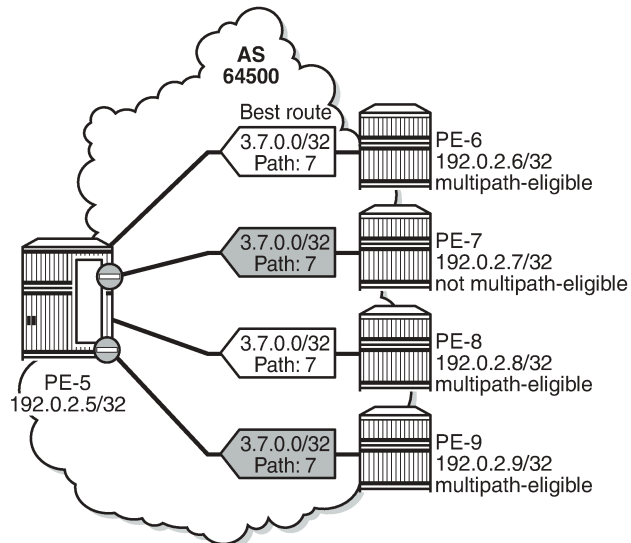
```

*A: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 68: Two IPv4 paths from multipath-eligible BGP peers are used](#) 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 68: 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

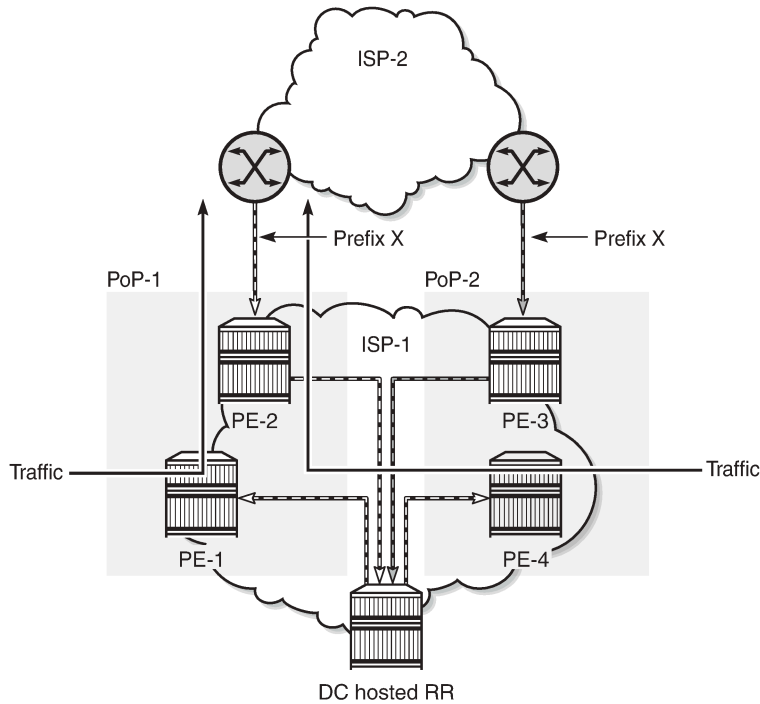
This chapter was initially written based on SR OS Release 15.0.R4, but the CLI in the current edition corresponds to SR OS Release 23.7.R2.

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 69: 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.

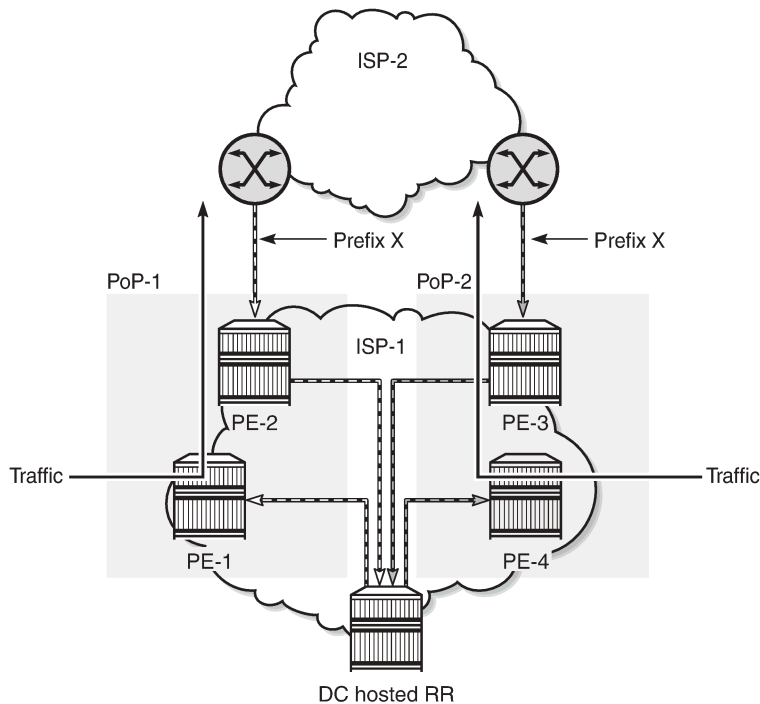
Figure 69: Centralized route reflection



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 70: 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 70: 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.

```
*A:RR-5>config>router>bgp# optimal-route-reflection ?
- optimal-route-reflection

    location      + Configure location ID for route reflector
[no] spf-wait    - Configure the spf-wait parameters
```

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.

```
*A:RR-5>config>router>bgp>orr# spf-wait ?
- spf-wait <max-wait> [initial-wait <initial-wait>] [second-wait <second-wait>]
```

```

<max-wait>           : [1..600] in seconds
<initial-wait>      : [1..300] in seconds
<second-wait>       : [1..300] in seconds

```

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.

```

*A:RR-5>config>router>bgp>orr# location ?
- location <location-id> [primary-ip-address <ipv4-address>] [secondary-ip-address <ipv4-
address>]
  [tertiary-ip-address <ipv4-address>]

<location-id>       : 1..255

[no] primary-ip-add* - Configure Primary IP address for location ID
[no] primary-ipv6-a* - Configure Primary IPv6 address for location ID
[no] secondary-ip-a* - Configure Secondary IP address for location ID
[no] secondary-ipv6* - Configure Secondary IPv6 address for location ID
[no] tertiary-ip-ad* - Configure Tertiary IP address for location ID
[no] tertiary-ipv6-* - Configure Tertiary IPv6 address for location ID

```

The locations are then referred to with the **cluster** command (residing in the BGP **group** context) through the **orr-location** argument, as follows.

```

*A:RR-5>config>router>bgp>group# cluster ?
- cluster <cluster-id> orr-location <orr-location> [allow-local-fallback]
- cluster <cluster-id>
- no cluster

<cluster-id>       : expressed in dotted decimal format (a.b.c.d)
<orr-location>     : [1..255]
<allow-local-fallb*> : configure to allow fallback on default orr location

*A:RR-5>config>router>bgp>group# neighbor 192.0.2.3 cluster ?
- cluster <cluster-id> orr-location <orr-location> [allow-local-fallback]
- cluster <cluster-id>
- no cluster

<cluster-id>       : expressed in dotted decimal format (a.b.c.d)
<orr-location>     : [1..255]
<allow-local-fallb*> : configure to allow fallback on default orr location

```

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

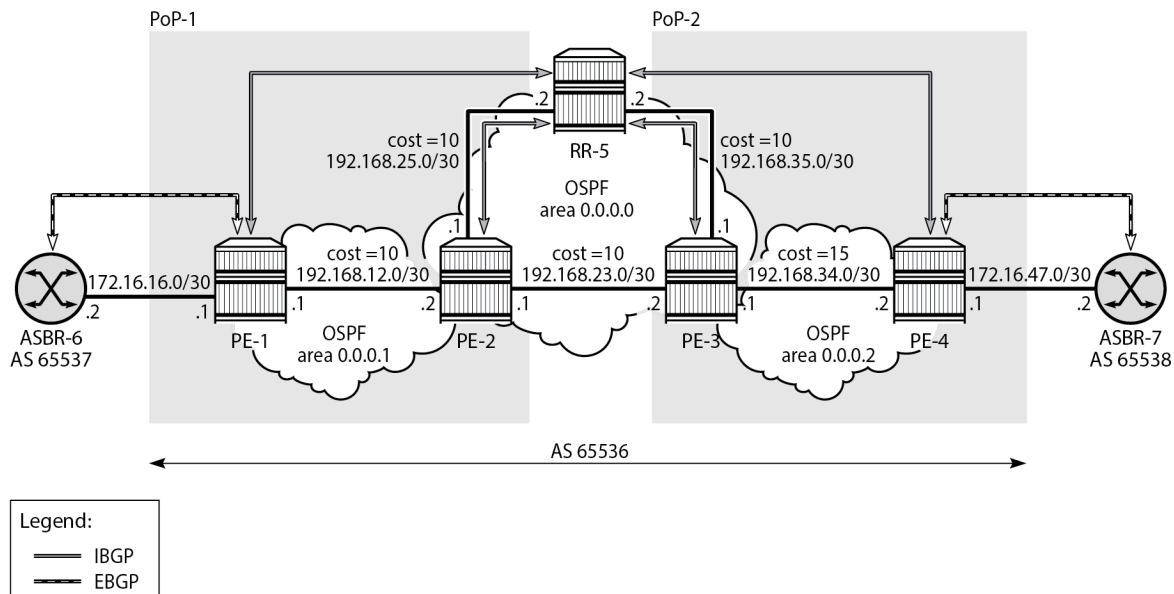
The following properties apply to ORR in SR OS:

- 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 71: 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 71: Example hierarchical networking using OSPF



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
    traffic-engineering
    area 0.0.0.0
      interface "system"
        no shutdown
      exit
      interface "int-PE-3-PE-2"
        interface-type point-to-point
        metric 10
        no shutdown
      exit
      interface "int-PE-3-RR-5"
        interface-type point-to-point
        metric 10
        no shutdown
      exit
    exit
  area 0.0.0.2
    stub
    exit
    interface "int-PE-3-PE-4"
      interface-type point-to-point
      metric 15
      no shutdown
    exit
    interface "int-LB-BGP"
      no shutdown
    exit
  exit
no shutdown
exit
```

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
    group "IBGP"
      cluster 192.0.2.5
      peer-as 65536 # type internal
      neighbor 192.0.2.1
      exit
      neighbor 192.0.2.2
      exit
      neighbor 192.0.2.3
      exit
```

```

        neighbor 192.0.2.4
        exit
    exit
    no shutdown
exit

```

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
    group "EBGP"
      neighbor 172.16.16.2
      peer-as 65537
    exit
  exit
  group "IBGP"
    next-hop-self
    peer-as 65536
    neighbor 192.0.2.5
  exit
  exit
  no shutdown
exit

```

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
    group "IBGP"
      peer-as 65536
      neighbor 192.0.2.5
    exit
  exit
  no shutdown
exit

```

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: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: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: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      00h00m49s  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: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: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      00h02m39s  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: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: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    00h00m17s  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: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
-----
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:

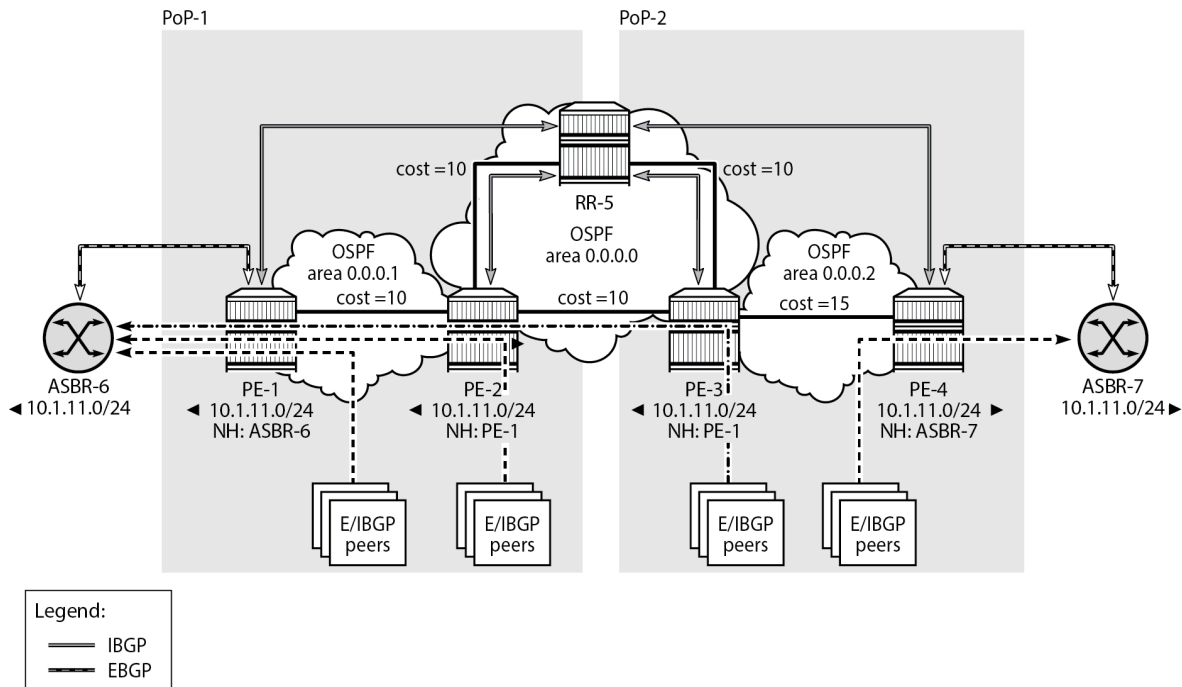
```

*A: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     00h00m41s  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 72: 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 72: Suboptimal route reflection



26686

Route reflection with ORR

Implementing ORR using the hierarchical topology from [Figure 72: 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
    traffic-engineering
    database-export identifier 1 bgp-ls-identifier 1
  area 0.0.0.1
    stub
  exit
interface "system"
  no shutdown
```



```

        exit
        interface "int-PE-1-PE-2"
            interface-type point-to-point
            no shutdown
        exit
    exit
    no shutdown
exit
bgp
    loop-detect discard-route
    split-horizon
    link-state-import-enable
    group "EBGP"
        neighbor 172.16.16.2
            peer-as 65537
        exit
    exit
    group "IBGP"
        family ipv4 bgp-ls
        next-hop-self
        peer-as 65536
        neighbor 192.0.2.5
        exit
    exit
    no shutdown
exit

```

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** 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 77: 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 **disable-route-table-install** command. The overall BGP configuration of RR-5 is as follows:

```

# on RR-5:
configure
    router Base
        autonomous-system 65536
        bgp
            family ipv4 bgp-ls
            loop-detect discard-route
            disable-route-table-install
            split-horizon
            link-state-export-enable
            optimal-route-reflection
                spf-wait 1 initial-wait 1 second-wait 1
                location 1
                    primary-ip-address 192.0.2.1
                exit
                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
            exit
        exit
    exit

```

```

    exit
  exit
  group "IBGP-1"
    cluster 192.0.2.5 orr-location 1 allow-local-fallback
    peer-as 65536
    neighbor 192.0.2.1
    exit
    neighbor 192.0.2.2
    exit
  exit
  group "IBGP-2"
    cluster 192.0.2.5 orr-location 2 allow-local-fallback
    peer-as 65536
    neighbor 192.0.2.3
    exit
    neighbor 192.0.2.4
    exit
  exit
no shutdown

```

With these changes applied, the following command can be used for verification of the BGP sessions:

```

*A: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      16   0 00h01m35s 1/0/0 (IPv4)
                  15   0           18/0/18 (LinkState)
192.0.2.2
Def. Inst          65536       7   0 00h01m35s 0/0/1 (IPv4)
                  8   0
192.0.2.3
Def. Inst          65536       7   0 00h01m35s 0/0/1 (IPv4)
                  9   0
192.0.2.4
Def. Inst          65536      15   0 00h01m17s 1/0/0 (IPv4)
                  17  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: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:

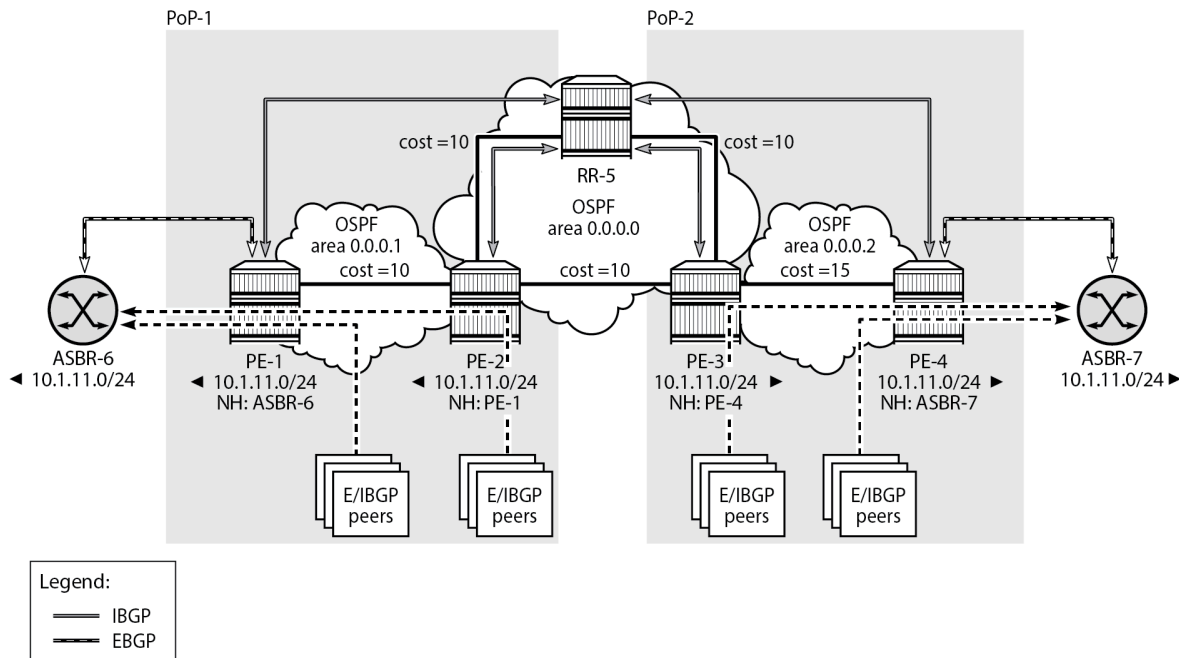
```

*A: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    00h00m34s 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 73: Optimal route reflection](#).

Figure 73: 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: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           : 00h04m56s
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

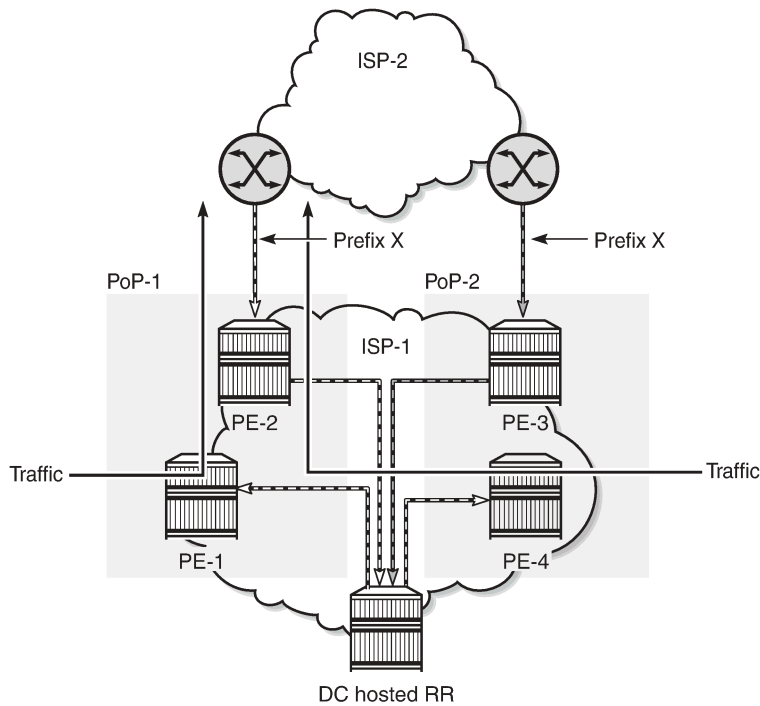
This chapter was initially written based on SR OS Release 15.0.R4, but the CLI in the current edition corresponds to SR OS Release 23.7.R2.

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 74: 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.

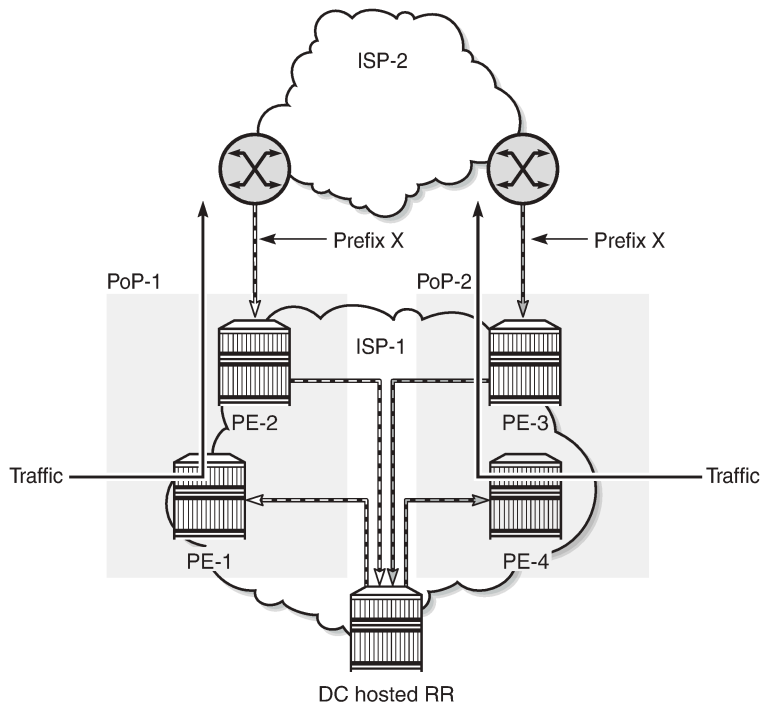
Figure 74: Centralized route reflection



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 75: 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 75: 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.

```
*A:RR-5>config>router>bgp# optimal-route-reflection ?
- optimal-route-reflection

    location      + Configure location ID for route reflector
[no] spf-wait    - Configure the spf-wait parameters
```

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.

```
*A:RR-5>config>router>bgp>orr# spf-wait ?
- spf-wait <max-wait> [initial-wait <initial-wait>] [second-wait <second-wait>]
```



```
<max-wait>           : [1..600] in seconds
<initial-wait>       : [1..300] in seconds
<second-wait>        : [1..300] in seconds
```

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.

```
*A:RR-5>config>router>bgp>orr# location ?
- location <location-id> [primary-ip-address <ipv4-address>] [secondary-ip-address <ipv4-
address>]
  [tertiary-ip-address <ipv4-address>]

<location-id>       : 1..255

[no] primary-ip-add* - Configure Primary IP address for location ID
[no] primary-ipv6-a* - Configure Primary IPv6 address for location ID
[no] secondary-ip-a* - Configure Secondary IP address for location ID
[no] secondary-ipv6* - Configure Secondary IPv6 address for location ID
[no] tertiary-ip-ad* - Configure Tertiary IP address for location ID
[no] tertiary-ipv6-* - Configure Tertiary IPv6 address for location ID
```

The locations are then referred to with the **cluster** command (residing in the BGP **group** context) through the **orr-location** argument, as follows.

```
*A:RR-5>config>router>bgp>group# cluster ?
- cluster <cluster-id> orr-location <orr-location> [allow-local-fallback]
- cluster <cluster-id>
- no cluster

<cluster-id>        : expressed in dotted decimal format (a.b.c.d)
<orr-location>      : [1..255]
<allow-local-fallb*> : configure to allow fallback on default orr location

*A:RR-5>config>router>bgp>group# neighbor 192.0.2.3 cluster ?
- cluster <cluster-id> orr-location <orr-location> [allow-local-fallback]
- cluster <cluster-id>
- no cluster

<cluster-id>        : expressed in dotted decimal format (a.b.c.d)
<orr-location>      : [1..255]
<allow-local-fallb*> : configure to allow fallback on default orr location
```

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

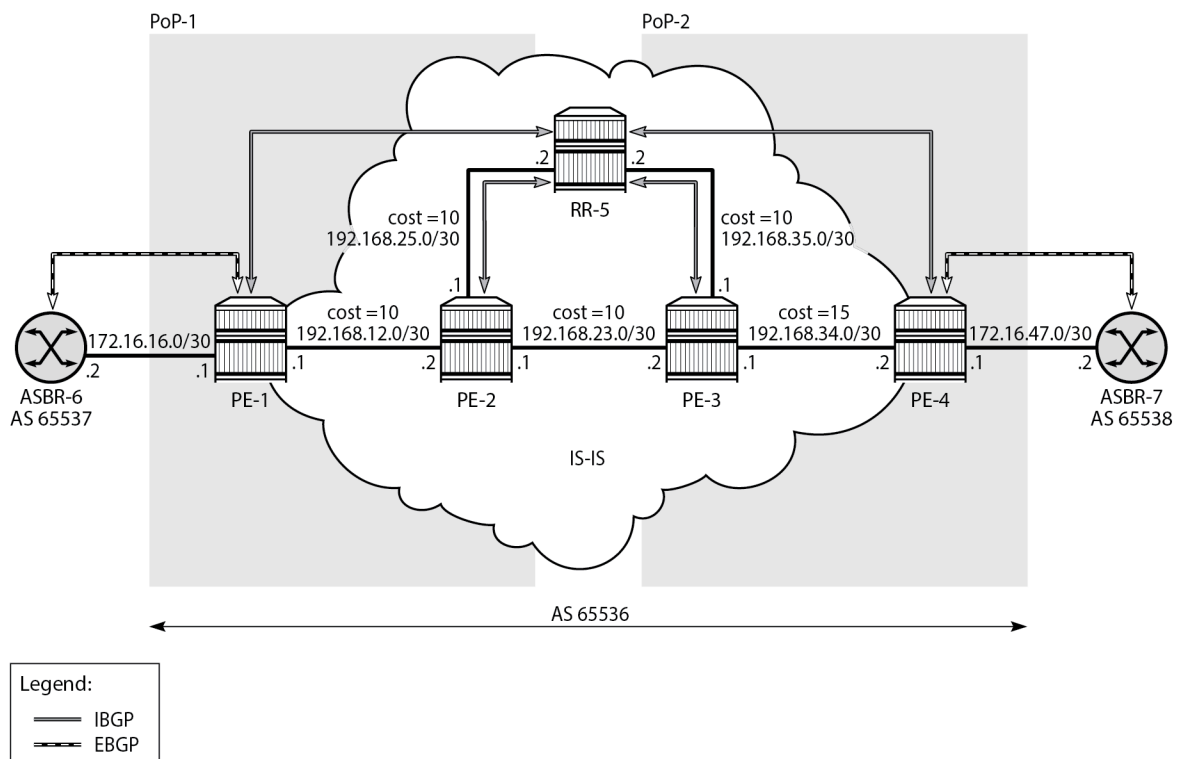
The following properties apply to ORR in SR OS:

- 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 76: 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 76: 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
      area-id 49.0001
      traffic-engineering
      interface "system"
        no shutdown
      exit
      interface "int-RR-5-PE-2"
        interface-type point-to-point
        no shutdown
      exit
      interface "int-RR-5-PE-3"
        interface-type point-to-point
        no shutdown
      exit
      no shutdown
    exit
```

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
      group "IBGP"
        cluster 192.0.2.5
        peer-as 65536          # type internal
        neighbor 192.0.2.1
        exit
        neighbor 192.0.2.2
        exit
        neighbor 192.0.2.3
        exit
        neighbor 192.0.2.4
        exit
      exit
      no shutdown
    exit
```

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
    group "EBGP"
      neighbor 172.16.16.2
      peer-as 65537
    exit
  exit
  group "IBGP"
    next-hop-self
    peer-as 65536
    neighbor 192.0.2.5
  exit
  exit
  no shutdown
exit
```

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
    group "IBGP"
      peer-as 65536
      neighbor 192.0.2.5
    exit
  exit
  no shutdown
exit
```

PE-4 also belongs to the cluster defined in the route reflector, but 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: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: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: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 00h01m33s 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: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: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   00h00m40s 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: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: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 00h01m05s 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.

```

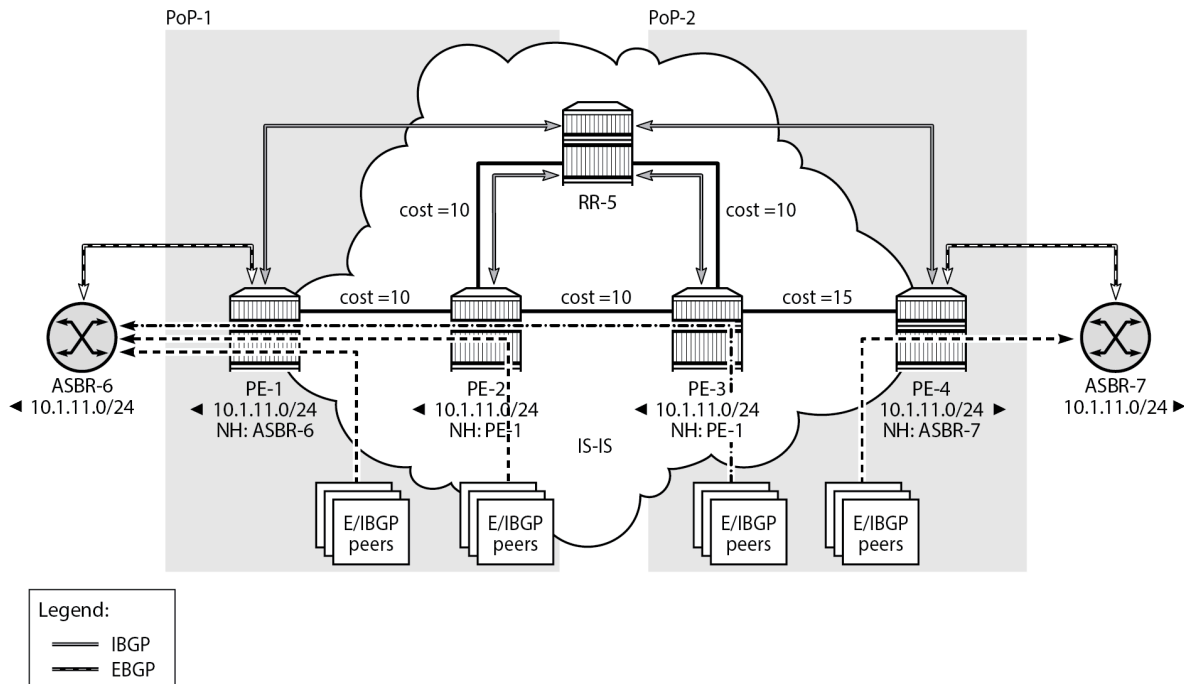
*A:PE-4# show router bgp routes
=====
BGP Router ID:192.0.2.4 AS:65536 Local AS:65536
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes : i - IGP, e - EGP, ? - incomplete
=====
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:PE-4# show router route-table protocol bgp
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]
Type Proto Age Metric Pref
-----
10.1.11.0/24
172.16.47.2 Remote BGP 00h01m54s 170
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 77: 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 77: Suboptimal route reflection



26683

Route reflection with ORR

For implementing ORR using the non-hierarchical topology from [Figure 77: 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
      optimal-route-reflection
        spf-wait 1 initial-wait 1 second-wait 1
        location 1
          primary-ip-address 192.0.2.1
        exit
        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
        exit
      exit
    group "IBGP-1"
      cluster 192.0.2.5 orr-location 1 allow-local-fallback
      peer-as 65536
      neighbor 192.0.2.1
    exit
    neighbor 192.0.2.2
    exit
  exit
  group "IBGP-2"
    cluster 192.0.2.5 orr-location 2 allow-local-fallback
    peer-as 65536
    neighbor 192.0.2.3
  exit
  neighbor 192.0.2.4
  exit
  exit
  no shutdown
exit
```

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.

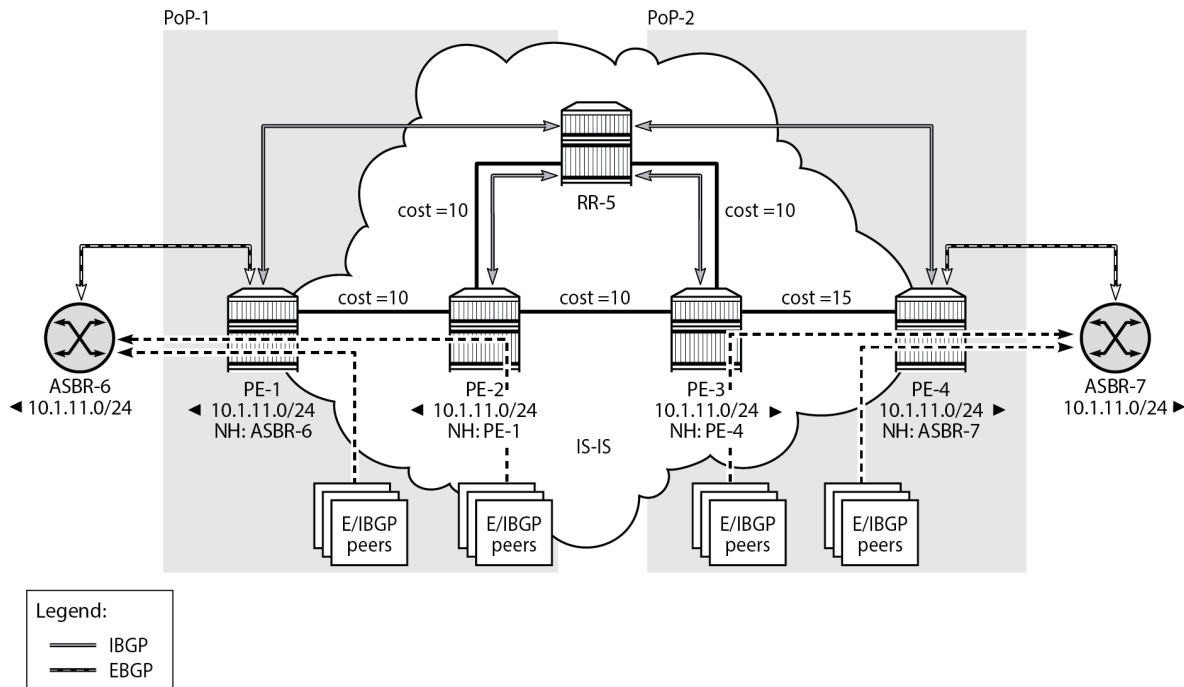
```
*A:PE-3# show router bgp routes
=====
BGP Router ID:192.0.2.3      AS:65536      Local AS:65536
=====
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - 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: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    00h03m12s  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 78: Optimal route reflection](#).

Figure 78: 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: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       : 00h04m02s
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	15
192.0.2.4						
1	192.0.2.4/32	IGP	Remote	ISIS	35	15
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 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 4: 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 4: 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	–

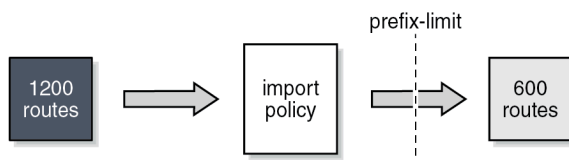
Address family	Base router	VPRN
vpn-ipv6	X	–
mvpn-ipv4	X	–
mvpn-ipv6	X	–
mcast-vpn-ipv4	X	–
mcast-vpn-ipv6	X	–
flow-vpn-ipv4	X	–
flow-vpn-ipv6	X	–
sr-policy-ipv4	X	–
sr-policy-ipv6	X	–
l2-vpn	X	–
mdt-safi	X	–
ms-pw	X	–
route-target	X	–
evpn	X	–
bgp-ls	X	–

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 idle-timeout option can be given the value *forever*, which corresponds to the default behavior of requiring a manual intervention if the limit is exceeded.

The **post-import** option indicates that the limit should be applied only to the routes accepted by import policies, as shown in [Figure 79: 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 79: 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** 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.

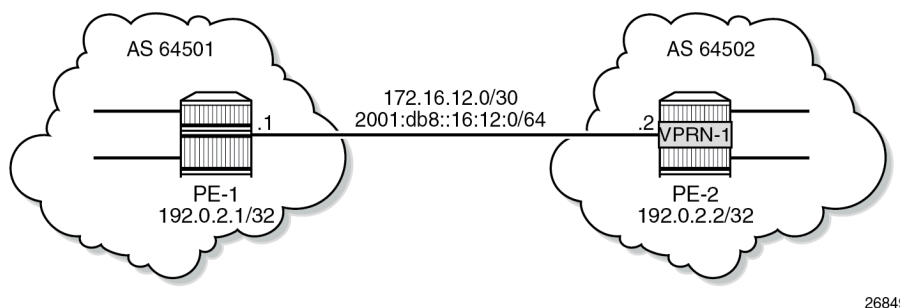
Configuration

Figure 80: 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 80: Example topology



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 1 name "VPRN-1" customer 1 create
      description "VPRN with BGP prefix limit"
      autonomous-system 64502
      route-distinguisher 64502:1
      interface "int-VPRN-1_PE-2.1-PE-1" create
        address 172.16.12.2/30
        ipv6
```

```

        address 2001:db8::16:12:2/126
        exit
        sap 1/1/c2/1:1 create
        exit
    exit
    bgp
        family ipv4 ipv6
        split-horizon
        loop-detect discard-route
        group "EBGP-to-AS64501"
            prefix-limit ipv4 10 threshold 50 idle-timeout 1
            prefix-limit ipv6 10 threshold 80 idle-timeout 4
            peer-as 64501
            neighbor 172.16.12.1
        exit
    exit
    no shutdown
exit
no shutdown

```

The debug configuration is as follows:

```

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 1, as follows:

```

configure
  log
    log-id 1 name "log-1"
    from debug-trace
    to memory
    no shutdown
  exit

```

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: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    : 1            Total Peers         : 1
Current Internal Groups : 1          Max Internal Groups : 1
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 Supressed 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	10	0	00h01m33s	3/3/0 (IPv4)	
		8	0		4/4/0 (IPv6)	

The following three BGP IPv4 routes are received by VPRN-1 on PE-2 and they are all active:

```
*A: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
	Nextthop (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:PE-2# show log log-id "99"
```

```

=====
Event Log 99 log-name 99
=====
Description : Default System Log
Memory Log contents [size=500  next event=111  (not wrapped)]

110 2022/11/24 09:51:46.230 UTC MINOR: BGP #2035 vprn1 Peer 2: 172.16.12.1
"(ASN 64501) VR 2: Group EBGp-to-AS64501: 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:PE-2# show log log-id "99"
---snip---

111 2022/11/24 09:52:51.229 UTC MINOR: BGP #2035 vprn1 Peer 2: 172.16.12.1
"(ASN 64501) VR 2: Group EBGp-to-AS64501: Peer 172.16.12.1: 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: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=41  (not wrapped)]

40 2022/11/24 09:53:51.229 UTC 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
"

39 2022/11/24 09:53:51.229 UTC MINOR: DEBUG #2001 vprn1 BGP
"BGP: STATE
Peer 2: 172.16.12.1 - Change State from ESTABLISHED to IDLE due to MAXPREFIX_EXCEEDED
"

38 2022/11/24 09:53:51.229 UTC 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
    10.1.2.0/24
    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
"
10.1.10.0/24

```

The BGP session is torn down and the corresponding state is disabled, as follows:

```

*A: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    : 1           Total Peers         : 1
Current Internal Groups : 0         Max Internal Groups : 1
Total BGP Paths      : 5           Total Path Memory   : 1760

Total IPv4 Remote Rts : 0      Total IPv4 Rem. Active Rts : 0
Total IPv6 Remote Rts : 0      Total IPv6 Rem. Active Rts : 0
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 Supressed 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      0    0 00h00m39s Disabled
          0          0
-----

```

Also, this event is recorded in the system logs, as follows:

```

*A:PE-2# show log log-id "99"
=====
Event Log 99 log-name 99
=====
Description : Default System Log
Memory Log contents [size=500 next event=132 (not wrapped)]
131 2022/11/24 09:56:47.236 UTC WARNING: BGP #2012 vprn1 Peer 2: 172.16.12.1

```

```
"(ASN 64501) Peer 2: 172.16.12.1: Closing connection: VR 2: Group EBGp-to-AS64501: Peer
172.16.12.1 not enabled or not in configuration"

130 2022/11/24 09:56:47.229 UTC WARNING: BGP #2005 vprn1 Peer 2: 172.16.12.1
"(ASN 64501) VR 2: Group EBGp-to-AS64501: Peer 172.16.12.1: sending notification: code CEASE
subcode MAX_PFX_RCHD"

129 2022/11/24 09:56:47.229 UTC WARNING: BGP #2039 vprn1 Peer 2: 172.16.12.1
"(ASN 64501) VR 2: Group EBGp-to-AS64501: 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 is then adapted as follows:

```
# on PE-2:
configure
  service
    vprn "VPRN-1"
      bgp
        family ipv4 ipv6
        loop-detect discard-route
        import "import-10.1-ranges"
        split-horizon
        group "EBGP-to-AS64501"
          prefix-limit ipv4 10 threshold 50 idle-timeout 1 post-import
          peer-as 64501
          neighbor 172.16.12.1
        exit
      exit
    no shutdown
```

The *import-10.1-ranges* policy is defined as follows:

```
# on PE-2:
configure
  router Base
    policy-options
      begin
        prefix-list "pfx-10.1-ranges"
          prefix 10.1.0.0/16 longer
        exit
      policy-statement "import-10.1-ranges"
        entry 10
          from
            prefix-list "pfx-10.1-ranges"
          exit
          action accept
        exit
```

```

exit
default-action drop
exit
exit
commit
    
```

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:PE-2# show router 1 route-table protocol bgp
=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]                               Type   Proto   Age           Pref
  Next Hop[Interface Name]                       Metric
-----
10.1.0.0/24                                     Remote BGP     00h00m44s    170
      172.16.12.1                               0
10.1.1.0/24                                     Remote BGP     00h00m44s    170
      172.16.12.1                               0
10.1.2.0/24                                     Remote BGP     00h00m44s    170
      172.16.12.1                               0
10.1.3.0/24                                     Remote BGP     00h00m44s    170
      172.16.12.1                               0
10.1.4.0/24                                     Remote BGP     00h00m44s    170
      172.16.12.1                               0
10.1.5.0/24                                     Remote BGP     00h00m44s    170
      172.16.12.1                               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 BGP session remains established with twelve received routes and six of these being active, as follows:

```

*A: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    : 1           Total Peers          : 1
Current Internal Groups : 1       Max Internal Groups  : 1
Total BGP Paths      : 6           Total Path Memory    : 2120

Total IPv4 Remote Rts : 12      Total IPv4 Rem. Active Rts : 6
Total IPv6 Remote Rts : 0          Total IPv6 Rem. Active Rts : 0
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	22	0	00h05m59s	12/6/0 (IPv4)	
		16	0		0/0/0 (IPv6)	

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** option causes BGP to remove only the private ASNs until the first public ASN.
 - The **skip-peer-as** 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** 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** 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:

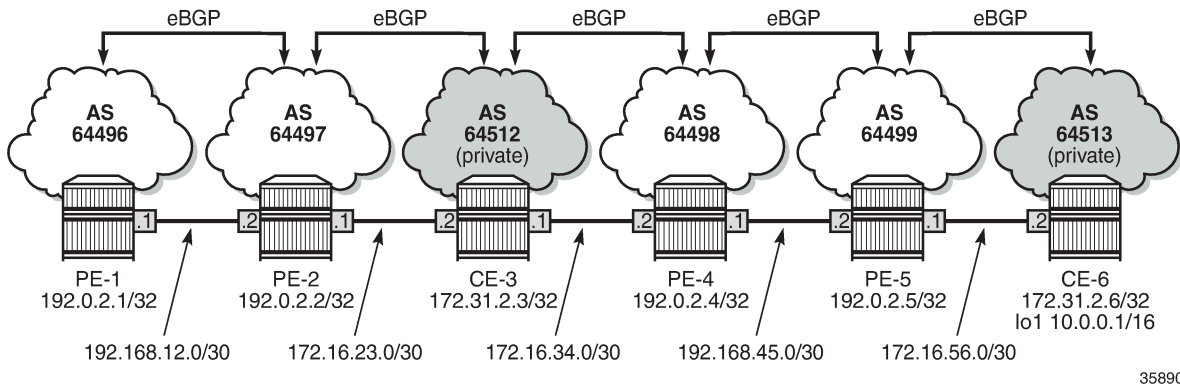
When **as-override** is enabled in the same session as **remove-private**, processing related to **remove-private** occurs first, followed by the processing related to **as-override**.

Configuration

Use case 1: Initial example topology

Figure 81: 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 81: Use case 1 topology



35890

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
    group "eBGP"
      family ipv4
      neighbor 172.16.23.2
```



```

        peer-as 64512
    exit
    neighbor 192.168.12.1
        peer-as 64496
    exit
exit
no shutdown
exit

```

CE-6 exports prefix 10.0.0.0/16. The configuration is as follows:

```

# on CE-6:
configure
  router Base
    interface "int-CE-6-PE-5"
      address 172.16.56.2/30
      port 1/1/c1/2:100
      no shutdown
    exit
    interface "lo1"
      address 10.0.0.1/16
      loopback
      no shutdown
    exit
    interface "system"
      address 172.31.2.6/32
      no shutdown
    exit
    autonomous-system 64513
    policy-options
      begin
      prefix-list "10.0.0.0/16"
        prefix 10.0.0.0/16 longer
      exit
      policy-statement "export-prefix"
        entry 10
          from
            prefix-list "10.0.0.0/16"
          exit
          action accept
          exit
        exit
      exit
    commit
  exit
  bgp
    split-horizon
    group "eBGP"
      family ipv4
      neighbor 172.16.56.1
        export "export-prefix"
        peer-as 64499
      exit
    exit
  no shutdown
exit

```

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: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:

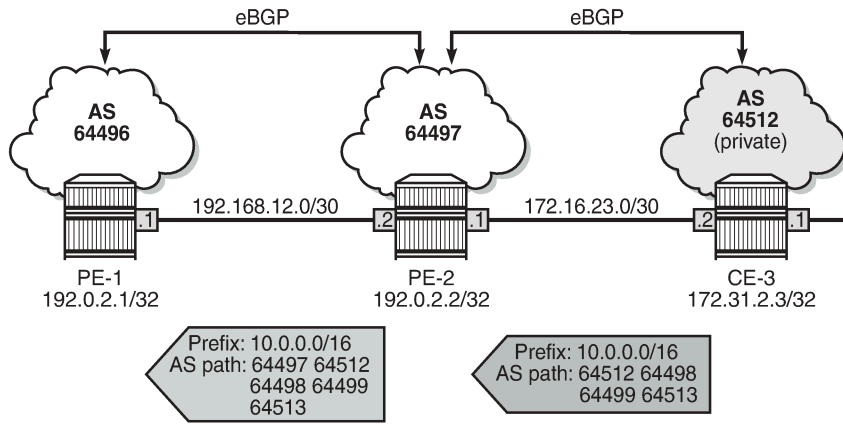
```

*A: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 82: 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 82: 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 and afterward with replace.

- **remove-private** ASN without any extra option (= default setting)
- **remove-private** ASN with **limited** option
- **remove-private** ASN with **skip-peer-as** 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
    group "eBGP"
      family ipv4
        neighbor 172.16.23.2
          peer-as 64512
        exit
      neighbor 192.168.12.1
        remove-private
        peer-as 64496
      exit
    exit
  no shutdown
exit
```

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:PE-1# show router bgp routes 10.0.0.0/16
```

```
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
```

```
Legend -
```

```

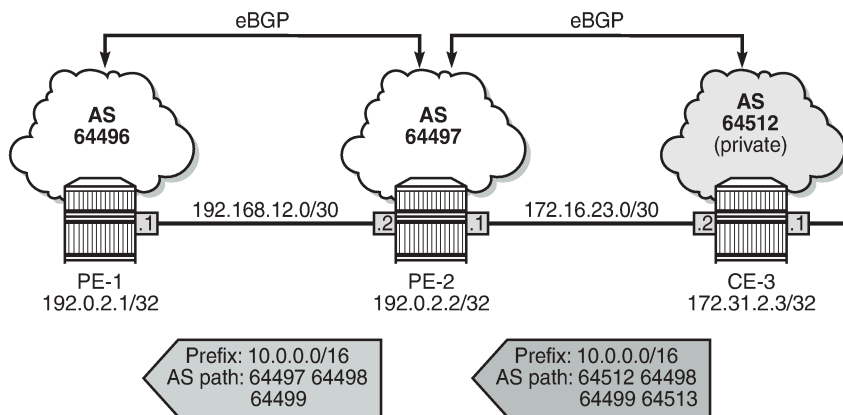
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete

=====
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 83: 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 83: 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** option for neighbor 192.168.12.1, as follows:

```

# on PE-2:
configure
  router Base
    bgp
      split-horizon
      group "eBGP"
        family ipv4
          neighbor 172.16.23.2
            peer-as 64512
          exit
        neighbor 192.168.12.1
          remove-private replace
            peer-as 64496

```

```

exit
exit
no shutdown
exit
    
```

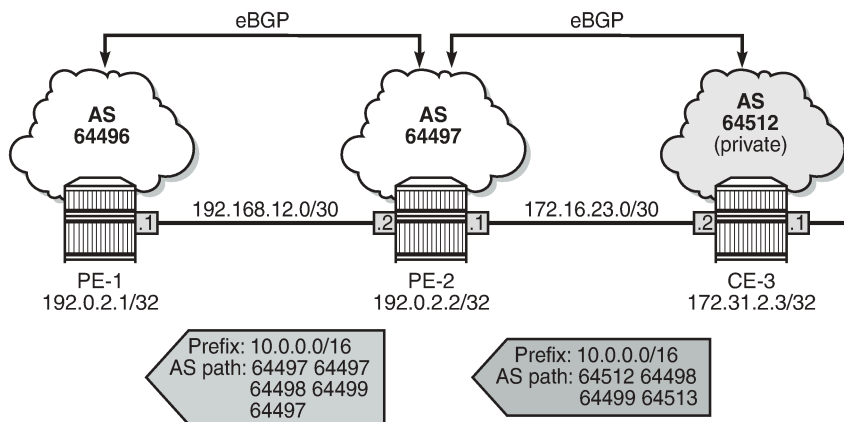
PE-2 adds its ASN 64497 and replaces the private ASNs 64512 and 64513 with its own public ASN 64497 (in bold), so ASN 64497 occurs three times in the AS path, as follows:

```

*A: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 84: 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 84: PE-2 adds its own ASN and replaces all private ASNs with its own ASN

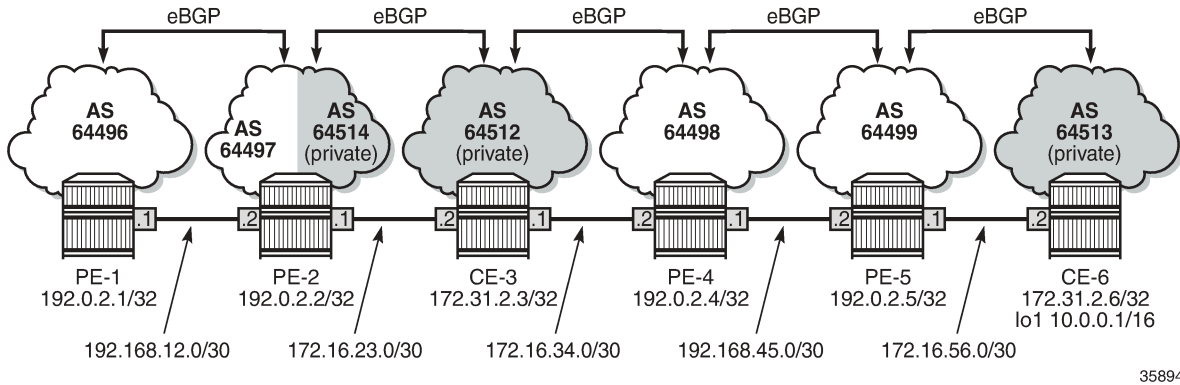


35893

Use case 2: Local private ASN in PE-2

Figure 85: 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 85: Use case 2 topology



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
    group "eBGP"
      family ipv4
        neighbor 172.16.23.2
          local-as 64514
          peer-as 64512
        exit
      neighbor 192.168.12.1
        no remove-private
        peer-as 64496
      exit
    exit
  no shutdown
exit
```

The BGP configuration on CE-3 is modified as follows:

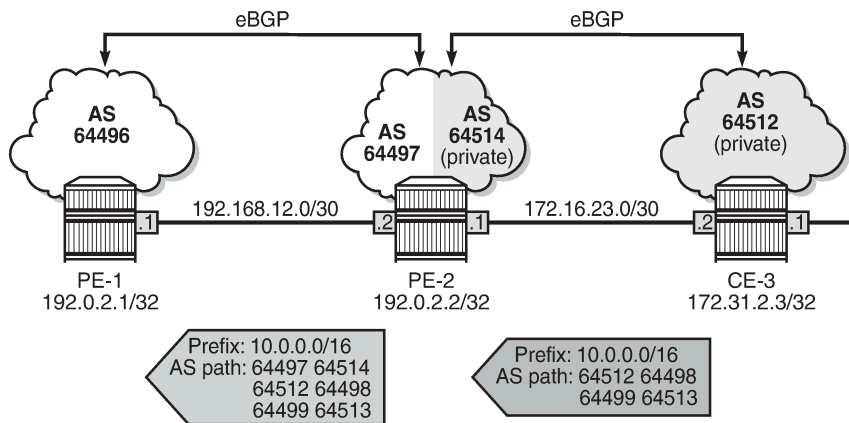
```
# on CE-3:
configure
router Base
  bgp
    group "eBGP"
      neighbor 172.16.23.1
        peer-as 64514
      exit
    exit
```

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: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 86: 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 86: 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
    split-horizon
    group "eBGP"
    family ipv4
```

```

neighbor 172.16.23.2
  local-as 64514 private
  peer-as 64512
exit
neighbor 192.168.12.1
  peer-as 64496
exit
exit
no shutdown
exit
    
```

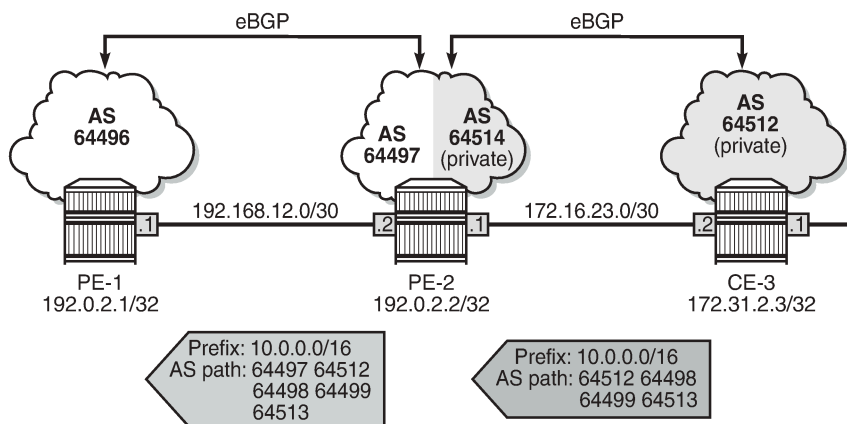
PE-1 receives the BGP route for prefix 10.0.0.0/16 with an AS path that does not include the private ASN 64514 anymore, as follows:

```

*A: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 87: 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 87: 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** option, as follows:

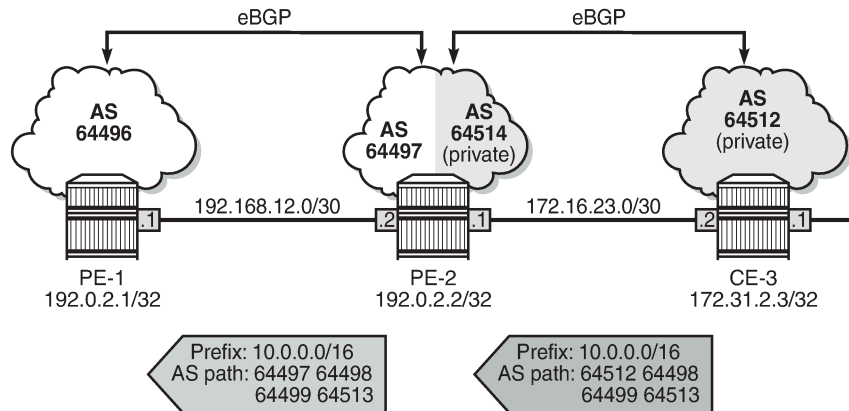
```
# on PE-2:
configure
router Base
  bgp
    split-horizon
    group "eBGP"
      family ipv4
        neighbor 172.16.23.2
          local-as 64514 private
          peer-as 64512
        exit
        neighbor 192.168.12.1
          remove-private limited
          peer-as 64496
        exit
      exit
    exit
  no shutdown
exit
```

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: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 88: PE-2 removes the private ASNs until the first public ASN shows the BGP routes received by PE-2 and PE-1.

Figure 88: 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** option is added to the **remove-private** settings:

```
# on PE-2:
configure
router Base
  bgp
    split-horizon
    group "eBGP"
      family ipv4
        neighbor 172.16.23.2
          local-as 64514 private
          peer-as 64512
        exit
        neighbor 192.168.12.1
          remove-private limited replace
          peer-as 64496
        exit
      exit
    exit
  no shutdown
exit
```

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:

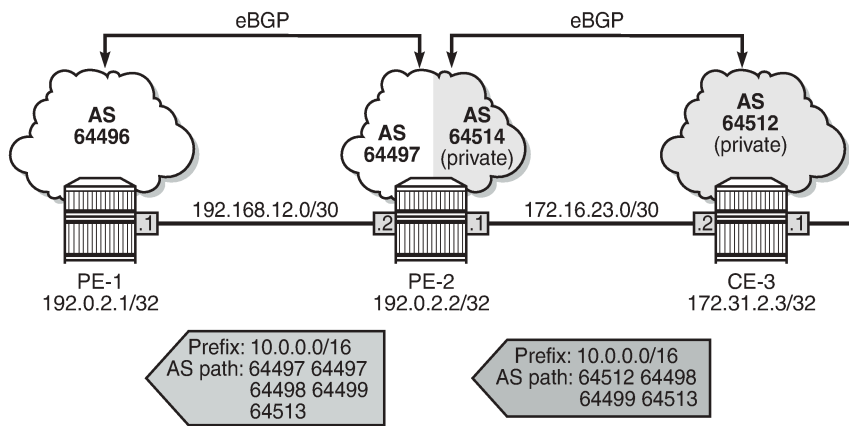
```
*A:PE-1# show router bgp routes 10.0.0.0/16
=====
BGP Router ID:192.0.2.1      AS:64496      Local AS:64496
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP IPv4 Routes
=====
```

Flag	Network	LocalPref	MED
	Nextthop (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 64513		-

Routes : 1			
=====			

This route is shown in [Figure 89: PE-2 replaces the private ASNs until the first public ASN.](#)

Figure 89: PE-2 replaces the private ASNs until the first public ASN

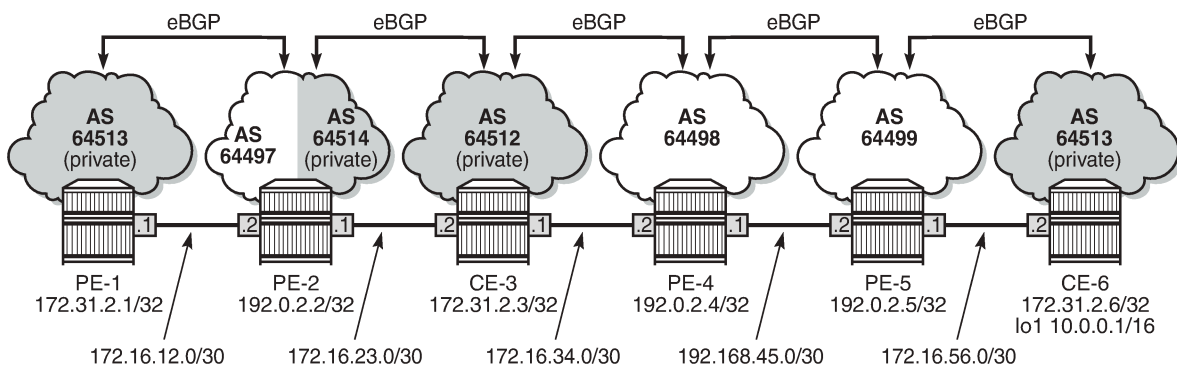


35898

Use case 3: CE-1 and CE-6 in the same private AS

[Figure 90: 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 90: 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
    group "eBGP"
      family ipv4
        neighbor 172.16.12.1
          peer-as 64513
        exit
      neighbor 172.16.23.2
        local-as 64514 private
        peer-as 64512
      exit
    exit
  no shutdown
exit
```

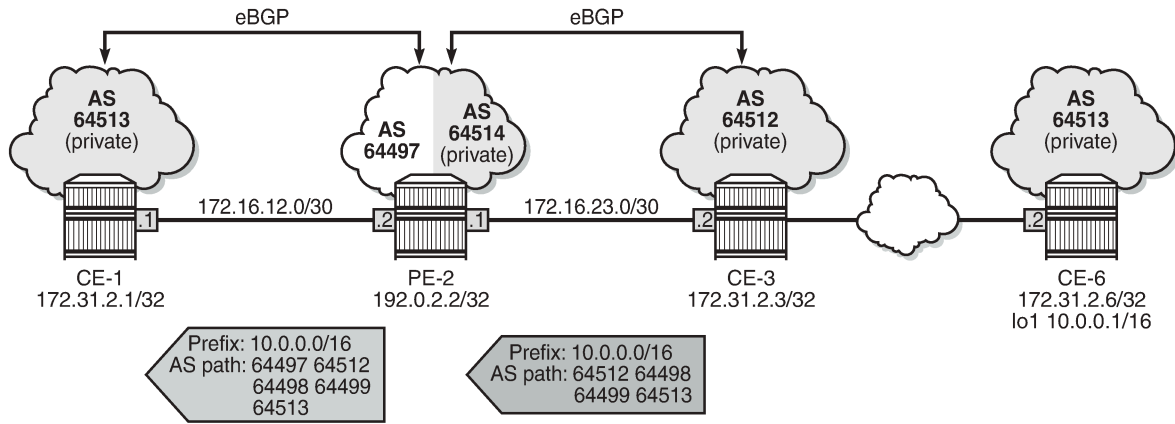
On CE-1, the received route for prefix 10.0.0.0/16 is invalid, because CE-1 detects its own ASN in the AS path attribute, which is considered an AS loop:

```
*A: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: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 91: PE-2 adds its public ASN to the AS path shows the BGP routes received by PE-2 and CE-1.

Figure 91: 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** option, as follows:

```
# on PE-2:
configure
  router Base
  bgp
    split-horizon
    group "eBGP"
      family ipv4
      neighbor 172.16.12.1
        remove-private skip-peer-as
        peer-as 64513
      exit
      neighbor 172.16.23.2
        local-as 64514 private
        peer-as 64512
      exit
    exit
  exit
  no shutdown
exit
```

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: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		-

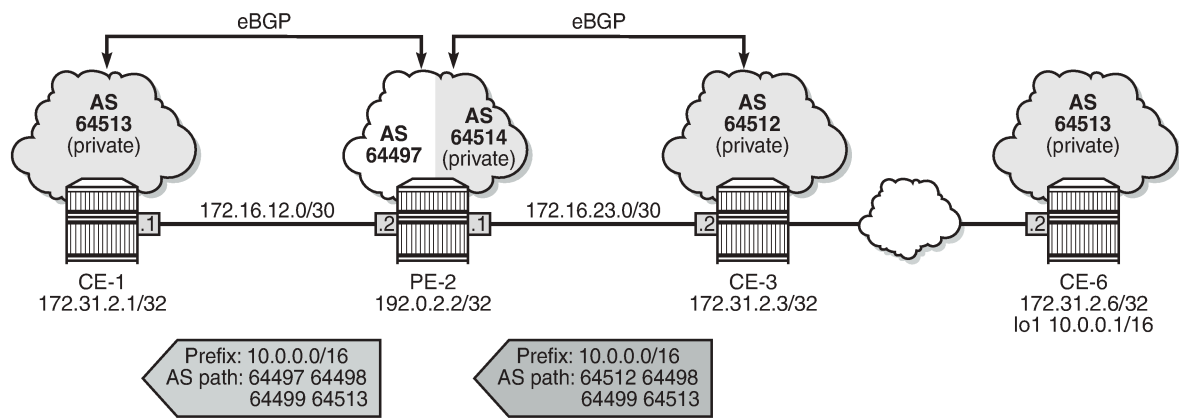
Routes : 1			
=====			

Again, this route is invalid because of the AS loop, as indicated by the flags:

```
*A: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 92: PE-2 removes the private ASNs except peer ASN 64513 shows the BGP routes received by PE-2 and CE-1.

Figure 92: 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** option, as follows:

```
# on PE-2:
configure
router Base
  bgp
    split-horizon
    group "eBGP"
      family ipv4
      neighbor 172.16.12.1
        remove-private skip-peer-as replace
        peer-as 64513
      exit
    neighbor 172.16.23.2
      local-as 64514 private
      peer-as 64512
```

```

exit
exit
no shutdown
exit
    
```

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: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
=====
    
```

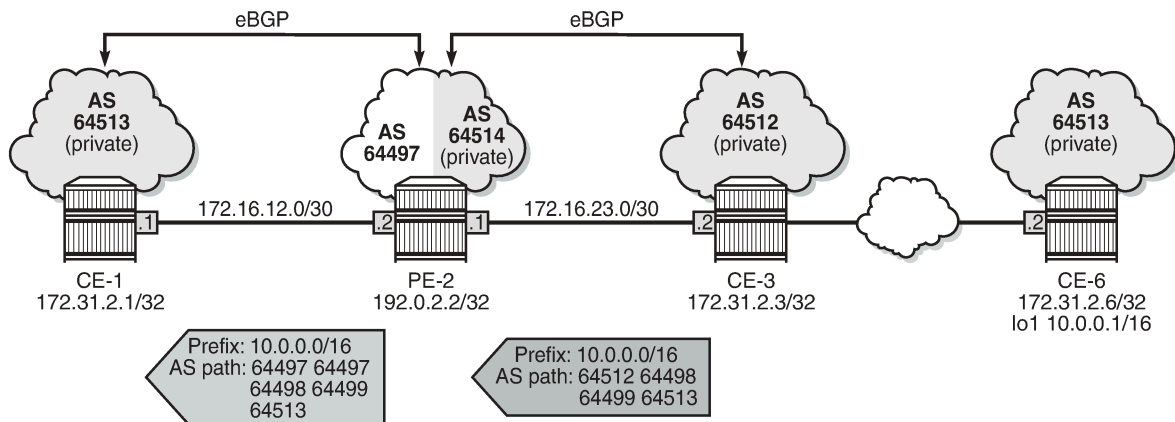
Again, the route is invalid because of the AS loop, as indicated by the flags:

```

*A: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 93: PE-2 replaces the private ASNs except peer ASN 64513 shows the received BGP routes on PE-2 and CE-1.

Figure 93: 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
    exit
```



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
    bgp
      group "eBGP"
        shutdown
        sleep 3
        no shutdown
    exit
```

```
*A: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

The information and configuration in this chapter were originally written for SR OS Release 14.0.R4. The 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/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.

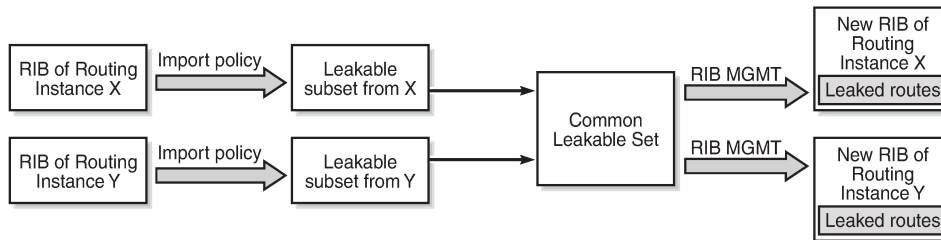
Any BGP route for an IPv4 or IPv6 prefix can be leaked. A BGP route does not have to be the best path or used for forwarding in the source instance in order to be leaked. In SR OS Releases earlier than 19.10.R1, the BGP route had to be valid (that is, the next-hop must be resolved; the AS PATH must not 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.

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 94: BGP route leaking process](#) shows the process of BGP route leaking.

Figure 94: 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.

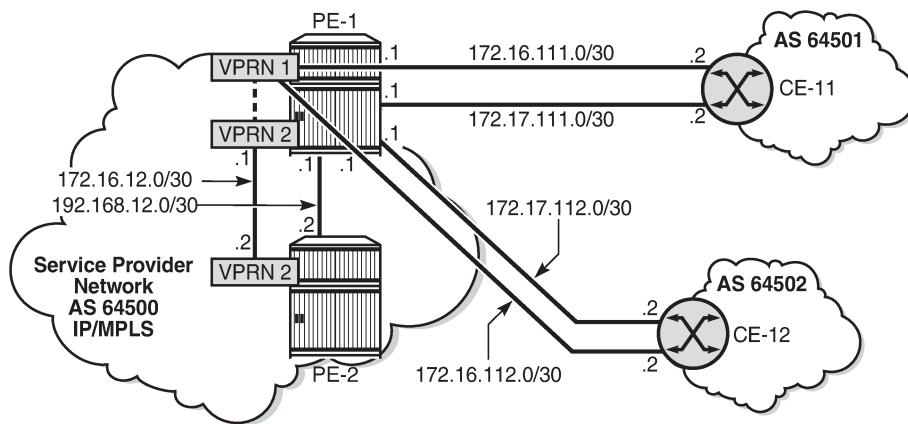
A BGP route leaked into a VPRN can be exported from the VPRN as a VPN-IPv4/v6 route if it matches the VRF export policy. Normal VPN export rules apply: by default, the leaked route is exported if it is the overall best path and it is used as the active route to the destination.

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*.

Configuration

[Figure 95: 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 95: Example topology](#). The interfaces also have IPv6 addresses, which will be shown in [Figure 99: BGP IPv6 route leaking between VPRNs](#) and [Figure 100: 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 95: 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

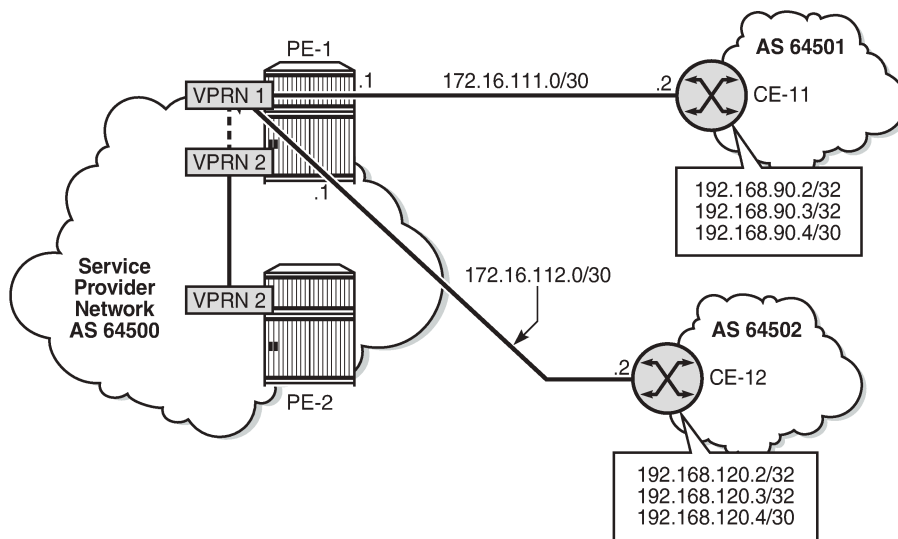
The nodes in the example topology have the following initial configuration:

- Cards, MDAs, ports
- Router interfaces
- IGP (IS-IS or OSPF) between the PEs
- LDP between the PEs
- VPRN 1 on PE-1; 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 96: 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 96: BGP IPv4 route leaking between VPRNs



25965

BGP leaking is disabled by default. The routing table for VPRN 1 on PE-1 includes routes that are learned from CE-11 and CE-12, as follows:

```
*A: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    00h01m28s    0
0
172.16.111.0/30
int-PE-1-CE-11                    Local  Local    00h01m28s    0
0
172.16.112.0/30
int-PE-1-CE-12                    Local  Local    00h01m28s    0
0
192.168.90.2/32
172.16.111.2                       Remote BGP      00h00m07s    170
0
192.168.90.3/32
172.16.111.2                       Remote BGP      00h00m07s    170
0
192.168.90.4/30
172.16.111.2                       Remote BGP      00h00m07s    170
0
192.168.120.2/32
172.16.112.2                      Remote BGP      00h00m05s    170
0
192.168.120.3/32
172.16.112.2                      Remote BGP      00h00m05s    170
0
192.168.120.4/32
172.16.112.2                      Remote BGP      00h00m05s    170
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: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: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  00h01m28s    0
   system                      0
172.16.2.2/32                Remote BGP VPN 00h00m41s   170
   192.0.2.2 (tunneled)         10
172.16.12.0/30               Local  Local  00h01m28s    0
   int-PE-1-PE-2_VPN2          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
=====
```

To configure BGP route leaking, an import policy is required in VPRN 1. The BGP route leaking policy is configured on PE-1, as follows:

```
# on PE-1:
configure
  router Base
    policy-options
      begin
        policy-statement "BGP-Leak-Policy"
          entry 10
            from
              protocol bgp
            exit
            action accept
            bgp-leak
          exit
        exit
      exit
```

```
exit
commit
```

By adding the **action accept bgp-leak**, 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.

The BGP route leaking policy can be applied in VPRN 1 in the general **bgp** context (as is the case here), in the **group** context, or per **neighbor**:

```
# on PE-1:
configure
service
  vprn "VPRN 1"
  bgp
    import "BGP-Leak-Policy"
  exit
```

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: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 detailed output for any route 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: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
Nextthop     : 172.16.111.2
Path Id      : None
From         : 172.16.111.2
Res. Protocol : LOCAL                      Res. Metric   : 0
Res. Nextthop : 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
  router Base
    policy-options
      begin
        policy-statement "Import-Leakable-Routes"
          entry 10
            from
              protocol bgp
            exit
            action accept
          exit
        exit
      exit
    exit
  commit
```

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 "VPRN 2" on PE-1 as follows:

```
# on PE-1:
configure
  service
    vprn "VPRN 2"
```



```

    bgp
      rib-management
        ipv4
          leak-import "Import-Leakable-Routes"
        exit
      exit
    exit
  
```

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: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                                  -
u*>li 192.168.120.2/32                      100      None
      172.16.112.2 (VPRN 1)                 None     0
      64502                                  -
u*>li 192.168.120.3/32                      100      None
      172.16.112.2 (VPRN 1)                 None     0
      64502                                  -
u*>li 192.168.120.4/32                      100      None
      172.16.112.2 (VPRN 1)                 None     0
      64502                                  -
-----
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: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
Nextthop     : 172.16.111.2 (VPRN 1)
Path Id      : None
From         : BGP VPRN 1
Res. Protocol : LOCAL                Res. Metric   : 0
Res. Nextthop : 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                        Dest Class    : 0
Add Paths Send : Default
RIB Priority   : Normal
Last Modified  : 00h02m13s
---snip---
    
```

The route table for VPRN 2 in the neighbor PE-2 contains the leaked routes, as follows:

```

*A:PE-2# 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                      Remote BGP VPN 00h09m36s 170
    192.0.2.1 (tunneled)              10
172.16.2.2/32                      Local  Local   00h10m20s  0
    system                             0
172.16.12.0/30                     Local  Local   00h10m20s  0
    int-PE-2-PE-1_VPN2                 0
192.168.90.2/32                    Remote BGP   00h02m28s 170
    172.16.12.1                         0
192.168.90.3/32                    Remote BGP   00h02m28s 170
    172.16.12.1                         0
192.168.90.4/30                    Remote BGP   00h02m28s 170
    172.16.12.1                         0
192.168.120.2/32                   Remote BGP   00h02m28s 170
    172.16.12.1                         0
192.168.120.3/32                   Remote BGP   00h02m28s 170
    172.16.12.1                         0
192.168.120.4/32                   Remote BGP   00h02m28s 170
    172.16.12.1                         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
=====

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 96: 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
router Base
  policy-options
  begin
  policy-statement "BGP-Leak-Policy"
  entry 10
  from
    protocol bgp
  exit
  action accept
  bgp-leak
  exit
  exit
exit
commit
```

In the preceding example, the BGP route leaking policy was applied in the global **bgp** context in VPRN "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
  group "EBGP_64500to64501_IPv4"
  neighbor 172.16.111.2
  import "BGP-Leak-Policy"
  exit
  exit
exit
```

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. Not all of these are leakable.

```
*A: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                                -
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
=====

```

Some routes are learned from CE-11 and other routes are learned from CE-12, and only the routes imported from CE-11 are leakable. The following command shows which IPv4 BGP routes are marked as leakable in VPRN 1 on PE-1:

```

*A: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
router Base
  policy-options
  begin
  policy-statement "Import-Leakable-Routes"
    entry 10
      from
        protocol bgp
      exit
      action accept
      exit
    exit
  exit
exit
commit
```

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 "Import-Leakable-Routes"
  exit
exit
exit
```

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: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)	100 None	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 shown for route 192.168.90.2/32:

```

*A: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

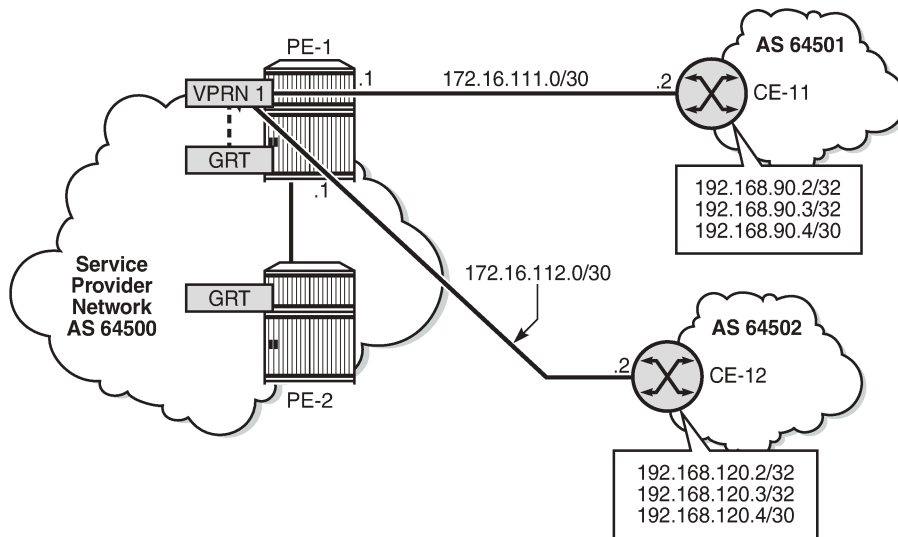
Network       : 192.168.90.2/32
Nextthop     : 172.16.111.2 (VPRN 1)
Path Id      : None
From         : BGP VPRN 1
Res. Protocol : LOCAL                Res. Metric   : 0
Res. Nextthop : 172.16.111.2
Local Pref.   : 100
Aggregator AS : None
Atomic Aggr.  : Not Atomic
AIGP Metric   : None
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None
Fwd Class     : None
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 97: 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. These routes will be marked as leakable and leaked to the GRT.

Figure 97: BGP IPv4 route leaking from VPRN to GRT



25966

The routing table for VPRN 1 in PE-1 contains the BGP routes exported by CE-11 and CE-12, as follows:

```
*A:PE-1# show router 1 route-table
=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]
Type Proto Age Metric Pref
-----
172.16.1.1/32
system Local Local 00h14m59s 0
172.16.111.0/30
int-PE-1-CE-11 Local Local 00h14m59s 0
172.16.112.0/30
int-PE-1-CE-12 Local Local 00h14m59s 0
192.168.90.2/32
172.16.111.2 Remote BGP 00h00m16s 170
192.168.90.3/32
172.16.111.2 Remote BGP 00h00m16s 170
192.168.90.4/30
172.16.111.2 Remote BGP 00h00m16s 170
192.168.120.2/32
172.16.112.2 Remote BGP 00h03m30s 170
192.168.120.3/32
172.16.112.2 Remote BGP 00h03m30s 170
192.168.120.4/32
172.16.112.2 Remote BGP 00h03m30s 170
-----
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 routing table of the base router does not include any of the BGP routes exported by the CEs, as follows:

```
*A: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   00h14m59s  0
      int-PE-1-CE-11                             0
172.17.112.0/30                                  Local  Local   00h14m59s  0
      int-PE-1-CE-12                             0
192.0.2.1/32                                     Local  Local   00h14m59s  0
      system                                       0
192.0.2.2/32                                     Remote  ISIS    00h14m44s  15
      192.168.12.2                               10
192.168.12.0/30                                  Local  Local   00h14m59s  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 BGP routes are marked as leakable after applying the following configuration:

```
# on PE-1:
configure
  router Base
    policy-options
      begin
        policy-statement "BGP-Leak-Policy"
          entry 10
            from
              protocol bgp
            exit
            action accept
            bgp-leak
          exit
        exit
      exit
    exit
  commit
```

This BGP route leaking policy can be applied in the general BGP configuration of VPRN 1, or per BGP group (as is the case here), or per BGP neighbor:

```
# on PE-1:
configure
  service
    vprn "VPRN 1"
      bgp
        group "EBGP_64500to64501_IPv4"
          import "BGP-Leak-Policy"
        exit
      exit
    exit
  exit
```


The following command shows the leakable BGP routes in VPRN 1:

```
*A: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:

```
# PE-1:
configure
  router Base
    policy-options
      begin
      policy-statement "Import-Leakable-Routes"
        entry 10
          from
            protocol bgp
          exit
          action accept
          exit
        exit
      exit
    exit
  commit
```

This import policy is applied in the base router, as follows:

```
# on PE-1:
configure
  router
    bgp
      rib-management
        ipv4
          leak-import "Import-Leakable-Routes"
        exit
      exit
```

```
exit
```

As a result, the leakable BGP routes in VPRN 1 are leaked to the GRT, as follows:

```
*A: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                                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: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              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---
```

The GRT includes the leaked routes, as follows:

```
*A: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  00h23m13s    0
    int-PE-1-CE-11              0
172.17.112.0/30             Local  Local  00h23m13s    0
    int-PE-1-CE-12              0
192.0.2.1/32                Local  Local  00h23m13s    0
    system                       0
192.0.2.2/32                Remote  ISIS   00h22m57s   15
    192.168.12.2                 10
192.168.12.0/30             Local  Local  00h23m13s    0
    int-PE-1-PE-2              0
192.168.90.2/32             Remote  BGP    00h04m49s   170
    172.16.111.2                 0
192.168.90.3/32             Remote  BGP    00h04m49s   170
    172.16.111.2                 0
192.168.90.4/30             Remote  BGP    00h04m49s   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: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   00h22m58s   15
    192.168.12.1                 10
192.0.2.2/32                Local  Local  00h23m06s    0
    system                       0
192.168.12.0/30             Local  Local  00h23m06s    0
    int-PE-2-PE-1              0
192.168.90.2/32             Remote  BGP    00h04m45s   170
    192.168.12.1                 10
192.168.90.3/32             Remote  BGP    00h04m45s   170
    192.168.12.1                 10
192.168.90.4/30             Remote  BGP    00h04m45s   170
```

```

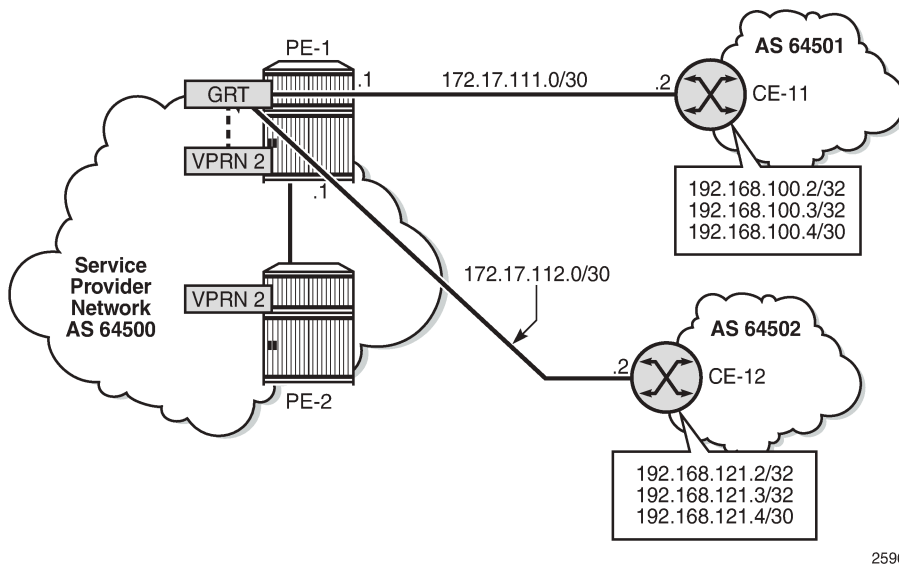
192.168.12.1                                     10
-----
No. of Routes: 6
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

Example 4 - BGP IPv4 route leaking from GRT to VPRN per neighbor

Figure 98: 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.

Figure 98: BGP IPv4 route leaking from GRT to VPRN



25966

The GRT in PE-1 includes BGP routes learned from CE-11 and CE-12, as follows:

```

*A: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   00h25m58s    0
  int-PE-1-CE-11                                0
172.17.112.0/30                                  Local  Local   00h25m58s    0
  int-PE-1-CE-12                                0
192.0.2.1/32                                      Local  Local   00h25m58s    0
  system                                          0
192.0.2.2/32                                      Remote  ISIS    00h25m43s    15
  192.168.12.2                                    10

```

```

192.168.12.0/30          Local  Local  00h25m58s  0
   int-PE-1-PE-2
192.168.100.2/32       Remote BGP    00h00m57s  170
   172.17.111.2
192.168.100.3/32       Remote BGP    00h00m57s  170
   172.17.111.2
192.168.100.4/30       Remote BGP    00h00m57s  170
   172.17.111.2
192.168.121.2/32       Remote BGP    00h01m08s  170
   172.17.112.2
192.168.121.3/32       Remote BGP    00h01m08s  170
   172.17.112.2
192.168.121.4/30       Remote BGP    00h01m08s  170
   172.17.112.2
-----
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 BGP leaking policy is the same as in the preceding examples:

```

# on PE-1:
configure
router Base
  policy-options
  begin
  policy-statement "BGP-Leak-Policy"
  entry 10
  from
  protocol bgp
  exit
  action accept
  bgp-leak
  exit
  exit
exit
commit

```

The BGP route leaking policy is applied on the base router for neighbor CE-11 only, as follows:

```

# on PE-1:
configure
router
  bgp
  group "EBGP_64500to64501_IPv4"
  neighbor 172.17.111.2
  import "BGP-Leak-Policy"
  exit
exit
exit

```

The following command shows that only the routes imported from neighbor CE-11 are marked as leakable in the GRT:

```

*A: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 NextHop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
u*>i	192.168.100.2/32 172.17.111.2 64501	None None	None 0 -
u*>i	192.168.100.3/32 172.17.111.2 64501	None None	None 0 -
u*>i	192.168.100.4/30 172.17.111.2 64501	None None	None 0 -

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
  router Base
    policy-options
      begin
        policy-statement "Import-Leakable-Routes"
          entry 10
            from
              protocol bgp
            exit
            action accept
          exit
        exit
      exit
    exit
  commit
```

This import policy is applied in VPRN 2, as follows:

```
# on PE-1:
configure
  service
    vprn 2
      bgp
        rib-management
          ipv4
            leak-import "Import-Leakable-Routes"
          exit
        exit
      exit
    exit
```

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: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: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
Nexthop       : 172.17.111.2 (Base)
Path Id       : None
From          : BGP Base
Res. Protocol : LOCAL                Res. Metric   : 0
Res. Nexthop  : 172.17.111.2
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 Base
AS-Path       : 64501

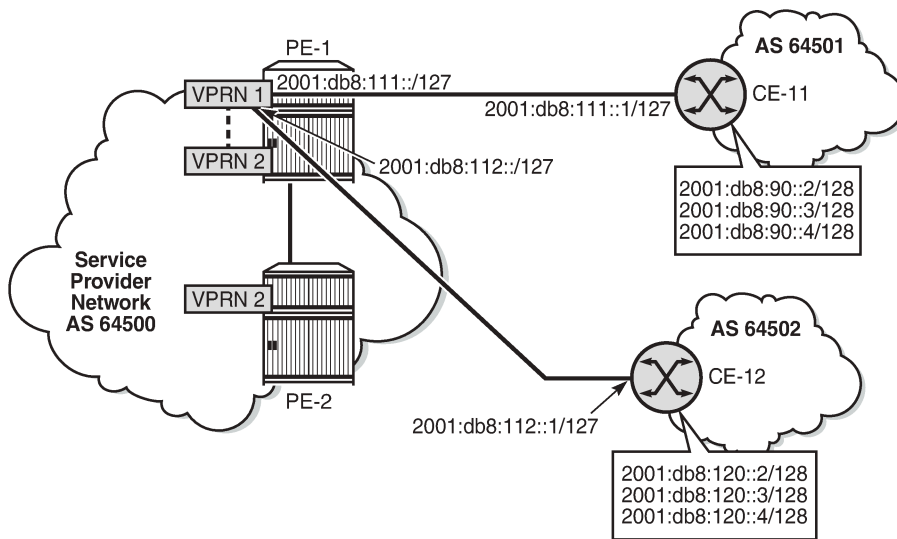
```

---snip---

Example 5 - BGP IPv6 route leaking between VPRNs. Global VPRN BGP configuration.

Figure 99: 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 99: BGP IPv6 route leaking between VPRNs



25968

```
*A:PE-1# show router 1 route-table ipv6
```

```
=====
IPv6 Route Table (Service: 1)
=====
```

Dest Prefix[Flags] Next Hop[Interface Name]	Type	Proto	Age	Metric	Pref
2001:db8::1:1/128 system	Local	Local	00h32m46s	0	0
2001:db8:90::2/128 2001:db8:111::1	Remote	BGP	00h00m44s	0	170
2001:db8:90::3/128 2001:db8:111::1	Remote	BGP	00h00m44s	0	170
2001:db8:90::4/126 2001:db8:111::1	Remote	BGP	00h00m44s	0	170
2001:db8:111::/127 int-PE-1-CE-11	Local	Local	00h32m46s	0	0
2001:db8:112::/127 int-PE-1-CE-12	Local	Local	00h32m46s	0	0
2001:db8:120::2/128 2001:db8:112::1	Remote	BGP	00h00m48s	0	170
2001:db8:120::3/128 2001:db8:112::1	Remote	BGP	00h00m48s	0	170


```

2001:db8:120::4/126          Remote BGP          00h00m48s 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 BGP route leaking policy is the same as for IPv4 routes:

```

# PE-1:
configure
  router Base
    policy-options
      begin
        policy-statement "BGP-Leak-Policy"
          entry 10
            from
              protocol bgp
            exit
            action accept
              bgp-leak
            exit
          exit
        exit
      commit

```

This import policy is applied in the **bgp** context of VPRN 1, as follows:

```

@ on PE-1:
configure
  service
    vprn "VPRN 1"
      bgp
        import "BGP-Leak-Policy"
      exit
    exit

```

With the preceding configuration, 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: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                               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
=====

```

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
router Base
  policy-options
  begin
  policy-statement "Import-Leakable-Routes"
  entry 10
  from
  protocol bgp
  exit
  action accept
  exit
  exit
exit
commit

```

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 2
  bgp
  rib-management
  ipv6
  leak-import "Import-Leakable-Routes"
  exit
exit
exit

```

The following command shows that the VPRN is importing the leaked BGP IPv6 routes from another VPRN instance:

```

*A: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                               Path-Id    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: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
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

```

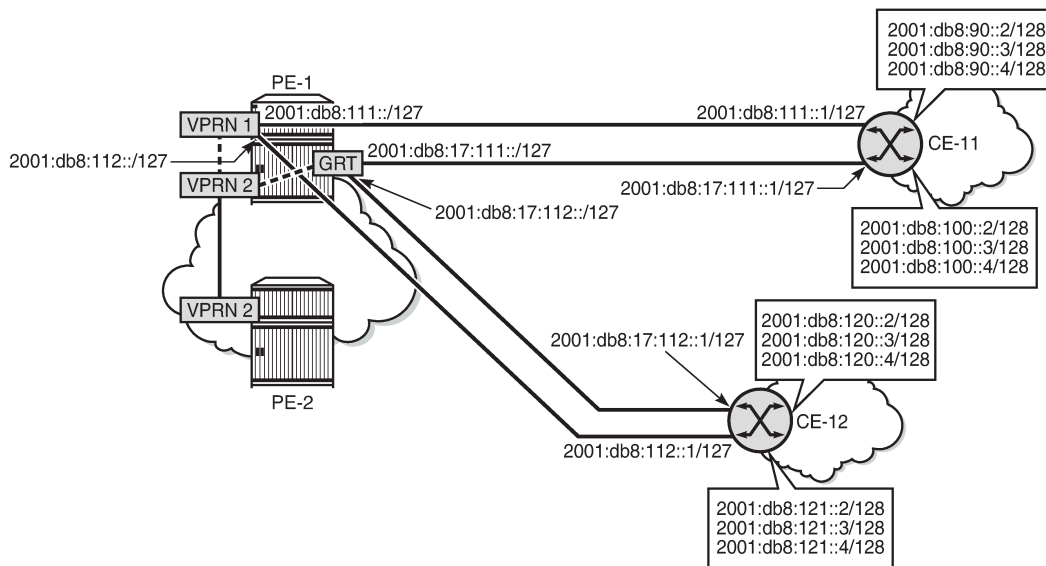
```

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
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 100: 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 100: BGP IPv6 route leaking from GRT and VPRN to VPRN



25969

The IPv6 routing table in the GRT contains routes exported by CE-11 and CE-12, as follows:

```

*A: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::1/128                                  Local  Local   00h42m19s    0
  system                                          0
2001:db8::2/128                                  Remote  ISIS    00h42m04s    15
  fe80::14:1ff:fe01:1-"int-PE-1-PE-2"          10
2001:db8:12::/126                                 Local  Local   00h42m18s    0
  int-PE-1-PE-2                                  0
2001:db8:17:111::/127                             Local  Local   00h42m18s    0
  int-PE-1-CE-11                                  0
    
```

```

2001:db8:17:112::/127          Local  Local  00h42m18s  0
    int-PE-1-CE-12              0
2001:db8:100::2/128           Remote BGP    00h02m54s  170
    2001:db8:17:111::1          0
2001:db8:100::3/128           Remote BGP    00h02m54s  170
    2001:db8:17:111::1          0
2001:db8:100::4/126           Remote BGP    00h02m54s  170
    2001:db8:17:111::1          0
2001:db8:121::2/128           Remote BGP    00h03m03s  170
    2001:db8:17:112::1          0
2001:db8:121::3/128           Remote BGP    00h03m03s  170
    2001:db8:17:112::1          0
2001:db8:121::4/126           Remote BGP    00h03m03s  170
    2001:db8:17:112::1          0
-----
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 IPv6 routing table for VPRN 1 also contains routes exported by CE-11 and CE-12, as follows:

```

*A:PE-1# show router 1 route-table ipv6

=====
IPv6 Route Table (Service: 1)
=====
Dest Prefix[Flags]           Type  Proto  Age      Pref
  Next Hop[Interface Name]   Metric
-----
2001:db8::1:1/128           Local  Local  00h42m18s  0
    system                    0
2001:db8:90::2/128          Remote BGP    00h03m57s  170
    2001:db8:111::1          0
2001:db8:90::3/128          Remote BGP    00h03m57s  170
    2001:db8:111::1          0
2001:db8:90::4/126          Remote BGP    00h03m57s  170
    2001:db8:111::1          0
2001:db8:111::/127          Local  Local  00h42m18s  0
    int-PE-1-CE-11           0
2001:db8:112::/127          Local  Local  00h42m18s  0
    int-PE-1-CE-12           0
2001:db8:120::2/128          Remote BGP    00h03m57s  170
    2001:db8:112::1          0
2001:db8:120::3/128          Remote BGP    00h03m57s  170
    2001:db8:112::1          0
2001:db8:120::4/126          Remote BGP    00h03m57s  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 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
  router Base
    policy-options
      begin
        prefix-list "2001:db8:90::"
          prefix 2001:db8:90::/100 longer
        exit
        prefix-list "2001:db8:120::"
          prefix 2001:db8:120::/100 longer
        exit
      policy-statement "BGP-Leak-Policy_90_120"
        entry 10
          from
            protocol bgp
            prefix-list "2001:db8:90::"
          exit
          action accept
            bgp-leak
          exit
        exit
        entry 20
          from
            protocol bgp
            prefix-list "2001:db8:120::"
          exit
          action accept
            bgp-leak
          exit
        exit
      exit
    exit
  commit
```

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 "BGP-Leak-Policy_90_120"
      exit
    exit
```

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
  router Base
    policy-options
      begin
        prefix-list "2001:db8:100::"
          prefix 2001:db8:100::/100 longer
        exit
        prefix-list "2001:db8:121::"
          prefix 2001:db8:121::/100 longer
```

```

exit
policy-statement "BGP-Leak-Policy_100_121"
  entry 10
    from
      protocol bgp
      prefix-list "2001:db8:100::"
    exit
    action accept
      bgp-leak
    exit
  exit
  entry 20
    from
      protocol bgp
      prefix-list "2001:db8:121::"
    exit
    action accept
      bgp-leak
    exit
  exit
exit
commit

```

This BGP leaking policy is applied for neighbor CE-11 in the base router, as follows. The routes exported by CE-12 will not be marked as leakable.

```

# on PE-1:
configure
  router Base
    bgp
      group "EBGP_64500to64501_IPv6"
        neighbor 2001:db8:17:111::1
          import "BGP-Leak-Policy_100_121"
      exit
    exit

```

The following command shows which routes are marked as leakable in the GRT:

```

*A: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                               LocalPref  MED
      Nexthop (Router)                     Path-Id    IGP Cost
      As-Path                               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: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                               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
  router Base
    policy-options
      begin
        policy-statement "Import-Leakable-Routes"
          entry 10
            from
              protocol bgp
            exit
            action accept
          exit
        exit
      exit
    exit

```



```
commit
```

This import policy is configured for IPv6 routes in VPRN2, as follows:

```
# on PE-1:
configure
  service
    vprn "VPRN 2"
      bgp
        rib-management
          ipv6
            leak-import "Import-Leakable-Routes"
          exit
        exit
      exit
    exit
  exit
```

The following command shows the leaked IPv6 routes in VPRN 2:

```
*A: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 Nextthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
u*>li	2001:db8:90::2/128 2001:db8:111::1 (VPRN 1) 64501	100 None	None 0 -
u*>li	2001:db8:90::3/128 2001:db8:111::1 (VPRN 1) 64501	100 None	None 0 -
u*>li	2001:db8:90::4/126 2001:db8:111::1 (VPRN 1) 64501	100 None	None 0 -
u*>li	2001:db8:100::2/128 2001:db8:17:111::1 (Base) 64501	100 None	None 0 -
u*>li	2001:db8:100::3/128 2001:db8:17:111::1 (Base) 64501	100 None	None 0 -
u*>li	2001:db8:100::4/126 2001:db8:17:111::1 (Base) 64501	100 None	None 0 -
u*>li	2001:db8:120::2/128 2001:db8:112::1 (VPRN 1) 64502	100 None	None 0 -
u*>li	2001:db8:120::3/128 2001:db8:112::1 (VPRN 1) 64502	100 None	None 0 -
u*>li	2001:db8:120::4/126 2001:db8:112::1 (VPRN 1) 64502	100 None	None 0 -

```
-----
```

```
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: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
Nextthop     : 2001:db8:17:111::1 (Base)
Path Id      : None
From         : BGP Base
Res. Protocol : LOCAL                      Res. Metric   : 0
Res. Nextthop : 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: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
Nextthop     : 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
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
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.

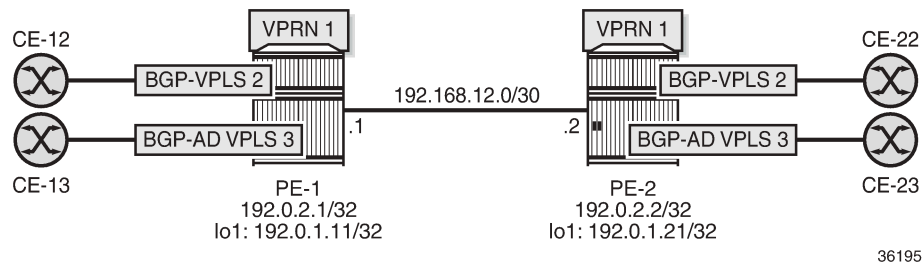
In this chapter, the following use cases are shown:

- 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 101: Example topology shows the example topology with two nodes.

Figure 101: 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
  router Base
    policy-options
      begin
      prefix-list "192.0.1.0/24"
        prefix 192.0.1.0/24 prefix-length-range 32-32
      exit
      community "target:64500:1"
        members "target:64500:1"
      exit
      community "target:64500:2"
        members "target:64500:2"
      exit
      policy-statement "export-bgp"
        entry 10
          from
            prefix-list "192.0.1.0/24"
          exit
          action accept
          exit
        exit
      exit
      policy-statement "export-VPLS2"
        entry 10
          action accept
          community add "target:64500:2"
        exit
      exit
      policy-statement "export-VPRN1"
        entry 10
          action accept
```

```

        community add "target:64500:1"
        next-hop 192.0.1.11
    exit
exit
policy-statement "import-VPLS2"
    entry 10
        from
            community "target:64500:2"
            family l2-vpn
        exit
        action accept
    exit
    default-action drop
exit
policy-statement "import-VPRN1"
    entry 10
        from
            protocol bgp-vpn
            community "target:64500:1"
        exit
        action accept
    exit
    default-action drop
exit
exit
commit

```

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
        bgp
            split-horizon
            next-hop-resolution
            labeled-routes
            transport-tunnel
            family label-ipv4
                resolution-filter
                no ldp
                sr-isis
            exit
            resolution filter
        exit
    exit
exit
group "iBGPv4"
    family vpn-ipv4 label-ipv4
    peer-as 64500
    neighbor 192.0.2.2
        export "export-bgp"
    exit
exit
group "iBGP-L2"
    family l2-vpn
    type internal
    local-address 192.0.1.11

```

```

neighbor 192.0.1.21
exit
exit

```

The service configuration on PE-1 is as follows:

```

# on PE-1:
configure
  service
    pw-template 1 name "PW1" create
    exit
    vprn 1 name "VPRN 1" customer 1 create
    vrf-import "import-VPRN1"
    vrf-export "export-VPRN1"
    route-distinguisher 64500:1
    auto-bind-tunnel
      resolution-filter
        bgp # default
    exit
    resolution filter
    exit
    vrf-target target:64500:1
    interface "lo1" create
    address 172.31.1.1/32
    loopback
    exit
    bgp
      next-hop-resolution
      use-bgp-routes
    exit
    exit
    no shutdown
  exit
  vpls 2 name "BGP-VPLS 2" customer 1 create
  bgp
    route-distinguisher 64500:2
    vsi-export "export-VPLS2"
    vsi-import "import-VPLS2"
    route-target export target:64500:2 import target:64500:2
    pw-template-binding 1 import-rt "target:64500:2"
    exit
  exit
  bgp-vpls
    max-ve-id 100
    ve-name "PE-1"
    ve-id 1
    exit
    no shutdown
  exit
  sap 1/2/1:2 create
  exit
  no shutdown
exit
vpls 3 name "BGP-AD VPLS 3" customer 1 create
bgp
  route-distinguisher 64500:3
  route-target export target:64500:3 import target:64500:3
  pw-template-binding 1
  exit
exit
bgp-ad
  vpls-id 64500:3
  vsi-id

```

```

        prefix 192.0.1.11
        exit
        no shutdown
    exit
    sap 1/2/1:3 create
    exit
    no shutdown
exit

```

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: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: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
  Nextthop (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: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: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                               Label
      Nexthop   VeId                               LocalPref
      As-Path   BaseOffset                               vplsLabelBase
-----
u*>i VPLS         -                               0
      64500:2    -                               -
      192.0.1.21 2                               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
  router Base
    policy-options
      begin
        policy-statement "import-VPRN1"
          entry 10
            from
              protocol bgp-vpn
              community "target:64500:1"
            exit
            action accept
              local-preference 200
            exit
          exit
        default-action drop
      exit
    exit
  commit
```

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

```

Change the import policy back to the original configuration, as follows:

```

# on PE-1:

```

```
configure
router Base
  policy-options
  begin
  policy-statement "import-VRPN1"
  entry 10
  from
  protocol bgp-vpn
  community "target:64500:1"
  exit
  action accept
  no local-preference
  exit
  exit
  exit
  commit
```

The **mp-bgp-keep** 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: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: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: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: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:PE-1# clear router bgp neighbor 192.0.2.2 soft-route-refresh # BGP-LU, BGP-VPN
*A: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](#)

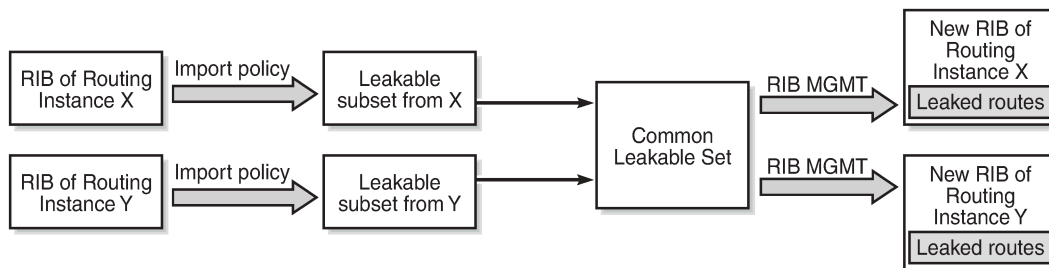
Applicability

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 102: BGP route leaking process between BGP routing instances X and Y](#) shows the BGP route leaking process between BGP routing instances.

Figure 102: 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** 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** 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 enabled, 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** 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.

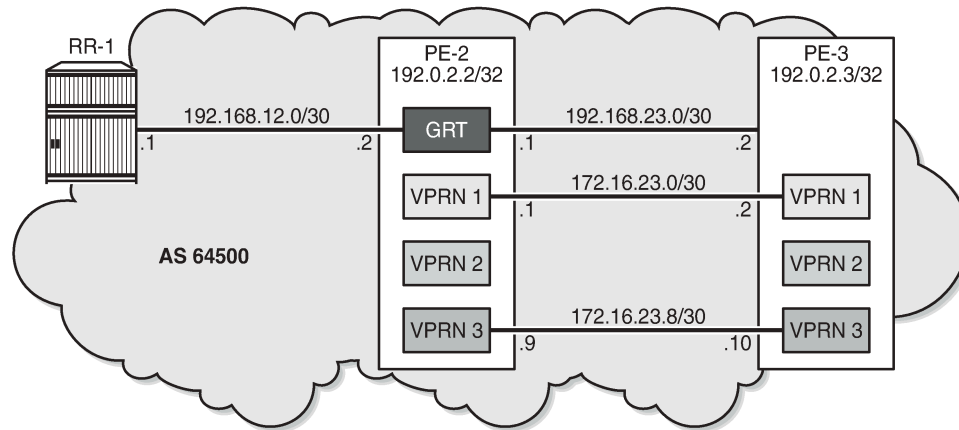
If a leaked BGP route is resolved by a VPRN, the VPRN can re-advertise the route to VPRN BGP peers or export the route as a VPN-IP route. However, if a leaked route is resolved over a BGP-VPN route, it can only be exported as a VPN-IP route if **allow-bgp-vpn-export** is enabled in the VPRN.

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 103: Example topology](#) shows the example topology with an RR and two PEs.

Figure 103: 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
    interface "int-PE-2-RR-1"
      address 192.168.12.2/30
      port 1/1/c1/3:100
      no shutdown
    exit
    interface "int-PE-2-PE-3"
      address 192.168.23.1/30
      port 1/1/c1/1:100
      no shutdown
    exit
    interface "system"
      address 192.0.2.2/32
      no shutdown
    exit
    autonomous-system 64500
    mpls-labels
      sr-labels start 32000 end 32999
    exit
    isis
      area-id 49.0001
      advertise-router-capability area
      segment-routing
        prefix-sid-range global
        no shutdown
      exit
      interface "system"
        ipv4-node-sid index 2
      exit
    interface "int-PE-2-PE-3"
```

```

        interface-type point-to-point
        exit
        no shutdown
    exit

```

A BGP session is established between RR-1 and the base router on PE-2. The BGP configuration on PE-2 is as follows:

```

# on PE-2:
configure
  router Base
    bgp
      split-horizon
      group "iBGP"
        family ipv4
          peer-as 64500
          neighbor 192.168.12.1
        exit
      exit
    exit
  exit

```

RR-1 advertises BGP routes with different communities for the different VPRNs on PE-2:

- prefix 10.14.0.0/16 with community "target:64501:1" for VPRN 1
- prefix 10.24.0.0/16 with community "target:64501:2" for VPRN 2
- prefix 10.34.0.0/16 with community "target:64501:3" for VPRN 3

PE-2 receives the following BGP routes from RR-1:

```

*A:PE-2# show router bgp neighbor 192.168.12.1 received-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     10.14.0.0/16                           100        None
      10.13.0.1                               None        0
      64501                                    -
i     10.24.0.0/16                           100        None
      10.23.0.1                               None        0
      No As-Path                               -
i     10.34.0.0/16                           100        None
      10.33.0.1                               None        0
      64503                                    -
-----
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: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
  router Base
    policy-options
      begin
        prefix-list "10.0.0.0/8"
          prefix 10.0.0.0/8 longer
        exit
      policy-statement "leak-10.x"
        entry 10
          from
            prefix-list "10.0.0.0/8"
          exit
          action accept
            bgp-leak
          exit
        exit
      exit
    exit
  commit
exit
bgp
  group "iBGP"
    family ipv4
    peer-as 64500
    neighbor 192.168.12.1
      import "leak-10.x"
    exit
  exit
exit
```

The routes are now marked as leakable:

```
*A:PE-2# show router bgp routes hunt | match Flags
Flags      : Invalid IGP Nexthop-Unresolved Leakable
Flags      : Invalid IGP Nexthop-Unresolved Leakable
Flags      : Invalid IGP Nexthop-Unresolved Leakable
```

```
*A:PE-2# show router bgp routes ipv4 leakable
=====
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 Nexthop (Router)	LocalPref Path-Id	MED IGP Cost
------	-----------------------------	----------------------	-----------------

	As-Path		Label
i	10.14.0.0/16	100	None
	10.13.0.1	None	0
	64501		-
i	10.24.0.0/16	100	None
	10.23.0.1	None	0
	No As-Path		-
i	10.34.0.0/16	100	None
	10.33.0.1	None	0
	64503		-

Routes : 3			
=====			

Even though the routes are marked as leakable, these BGP routes with unresolved next-hop are only leaked from the base router to a VPRN context when the command **allow-unresolved-leaking** is configured in the **BGP next-hop-resolution** context of the base router, as shown later in the examples.

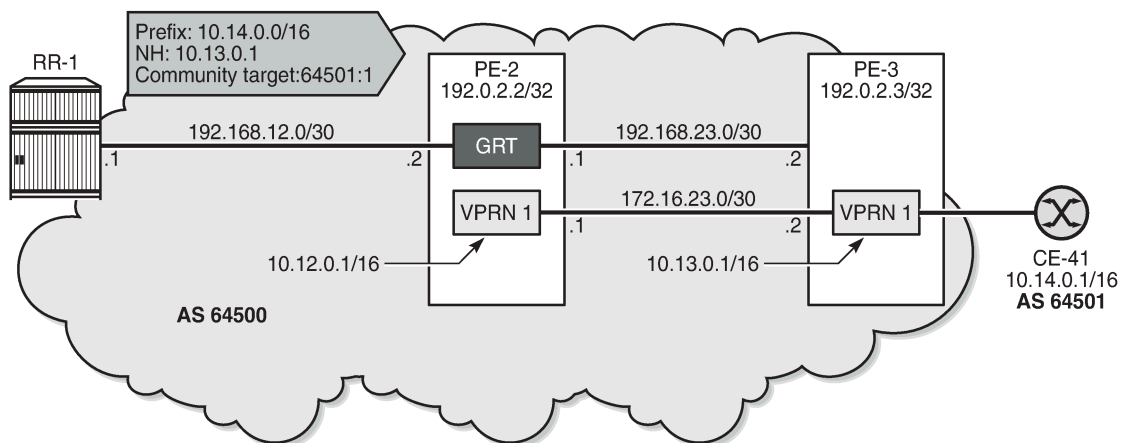
The following use cases are shown:

- BGP route 10.14.0.0/16 leaked to VPRN 1 with BGP next-hop resolved using IS-IS
- BGP route 10.24.0.0/16 leaked to VPRN 2 with BGP next-hop resolved using VPN-IP
- BGP route 10.34.0.0/16 leaked to VPRN 3 with BGP next-hop resolved using eBGP

Use case 1: BGP route leaked to VPRN 1 with next-hop resolved using IS-IS

Figure 104: 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 104: 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 1 name "VPRN 1" customer 1 create
    autonomous-system 64500
    route-distinguisher 64500:1
    vrf-target target:64500:1
    interface "lo1" create
      address 10.13.0.1/32
      loopback
    exit
    interface "int-VPRN1-PE-3-PE-2" create
      address 172.16.23.2/30
      sap 1/1/cl/2:1 create
    exit
  exit
  interface "int-VPRN3-PE-3-CE-41" create
    address 172.16.34.1/30
    sap 1/1/cl/1:1 create
  exit
  static-route-entry 10.14.0.0/16
    next-hop 172.16.34.2
    no shutdown
  exit
  isis 0
    area-id 49.0001
    interface "lo1"
      interface-type point-to-point
      no shutdown
    exit
    interface "int-VPRN1-PE-3-PE-2"
      interface-type point-to-point
      no shutdown
    exit
  no shutdown
exit

```

On PE-2, the route table for VPRN 1 shows the following IS-IS route for prefix 10.13.0.1/32:

```
*A: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   00h10m43s    0
   lo1
10.13.0.1/32                       Remote ISIS   00h10m13s    15
   172.16.23.2
172.16.23.0/30                       Local  Local   00h10m43s    0
   int-VPRN1-PE-2-PE-3
-----
No. of Routes: 3
---snip---
=====

```

PE-2 receives the following BGP route from RR-1 in the base routing instance with community "target:64500:1":

```
*A: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                               Path-Id    Label
-----
i     10.14.0.0/16                           100        None
      10.13.0.1                               None       0
      64501                                    -          -
-----
Routes : 1
=====
```

This route is leakable:

```
*A: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
  router Base
    policy-options
      begin
        community "target:64500:1"
          members "target:64500:1"
        exit
      policy-statement "leak-import-1"
        entry 10
          from
            community "target:64500:1"
          exit
          action accept
          exit
        exit
        default-action drop
        exit
      exit
    commit
  exit
exit
service
  vprn "VPRN 1"
    autonomous-system 64500
    route-distinguisher 64500:1
    vrf-target target:64500:1
    bgp
```

```

        rib-management
        ipv4
        leak-import "leak-import-1"
    exit
exit
exit
exit
exit
exit

```

By default, the base router does not leak unresolved routes, so the list of leaked BGP routes in VPRN 1 remains empty:

```

*A: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 following 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
  exit
exit

```

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: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:PE-2# show router 1 route-table

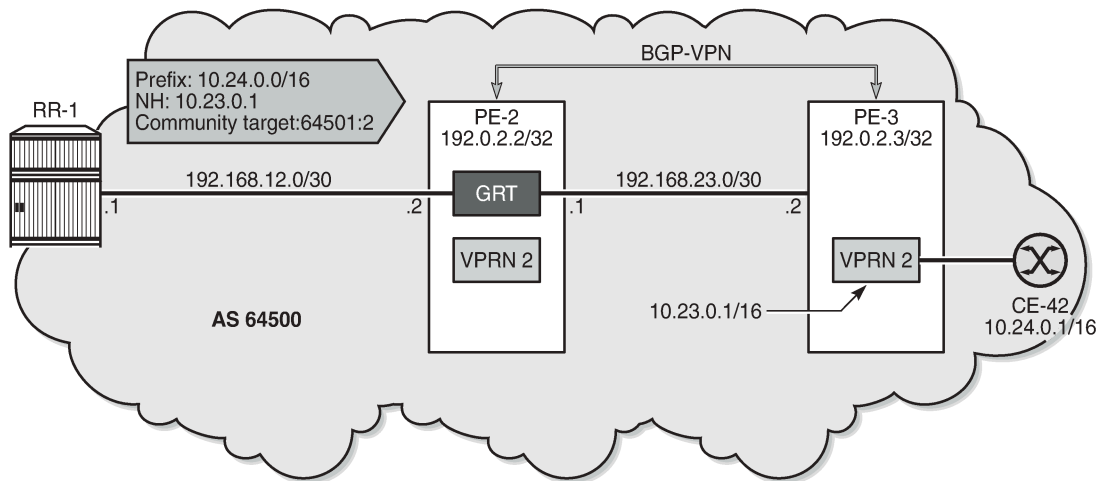
=====
Route Table (Service: 1)
=====
Dest Prefix[Flags]                        Type   Proto   Age           Pref
Next Hop[Interface Name]                 Metric
-----
10.12.0.1/32                               Local  Local   00h17m49s    0
      lo1                                   0
10.13.0.1/32                               Remote  ISIS    00h17m19s    15
      172.16.23.2                          10
10.14.0.0/16                               Remote BGP    00h00m17s    170
      172.16.23.2                          10
172.16.23.0/30                             Local  Local   00h17m49s    0
      int-VPRN1-PE-2-PE-3                  0
-----
No. of Routes: 4
---snip---
=====

```

Use case 2: BGP route leaked to VPRN 2 with next-hop resolved using VPN-IP

Figure 105: 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 105: 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 2 name "VPRN 2" customer 1 create
  autonomous-system 64500
  route-distinguisher 64500:2
  auto-bind-tunnel
  resolution-filter
  sr-isis
  exit
  resolution filter
  exit
  vrf-target target:64500:2
  interface "lo1" create
  address 10.23.0.1/32
  loopback
  exit
  no shutdown
  exit
```

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
    group "iBGP-VPN"
    family vpn-ipv4
    peer-as 64500
    neighbor 192.0.2.2
    exit
```

```
exit
exit
```

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: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   00h21m55s    0
  lo1                               0
10.23.0.1/32                     Remote BGP VPN 00h20m27s  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: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: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
router Base
policy-options
```

```

begin
community "target:64500:2"
  members "target:64500:2"
exit
policy-statement "leak-import-2"
  entry 10
    from
      community "target:64500:2"
    exit
    action accept
    exit
  exit
  default-action drop
  exit
exit
commit
exit
service
  vprn "VPRN 2"
    autonomous-system 64500
    route-distinguisher 64500:2
    auto-bind-tunnel
      resolution-filter
        sr-isis
      exit
      resolution filter
    exit
    vrf-target target:64500:2
    bgp
      rib-management
        ipv4
          leak-import "leak-import-2"
          exit
        exit
        no shutdown
      exit
      no shutdown
    exit
  exit

```

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: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: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"
      autonomous-system 64500
      route-distinguisher 64500:2
      auto-bind-tunnel
        resolution-filter
          sr-isis
        exit
      resolution filter
    exit
  vrf-target target:64500:2
  bgp
    next-hop-resolution
      use-bgp-routes      # for BGP and BGP-VPN routes
    exit
  rib-management
    ipv4
      leak-import "leak-import-2"
    exit
  exit
  no shutdown
exit
no shutdown
exit
```

When the next-hop can be resolved using a VPN-IPv4 route, the leaked route becomes used, valid, and best in VPRN 2:

```
*A: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
-----
u*>li 10.24.0.0/16              100        None
      10.23.0.1 (Base)      None        10
      No As-Path                -
-----
```

```
Routes : 1
=====
```

```
*A:PE-2# show router 2 bgp routes hunt | match Flags
Flags          : Used Valid Best IGP Leaked In-RTM
```

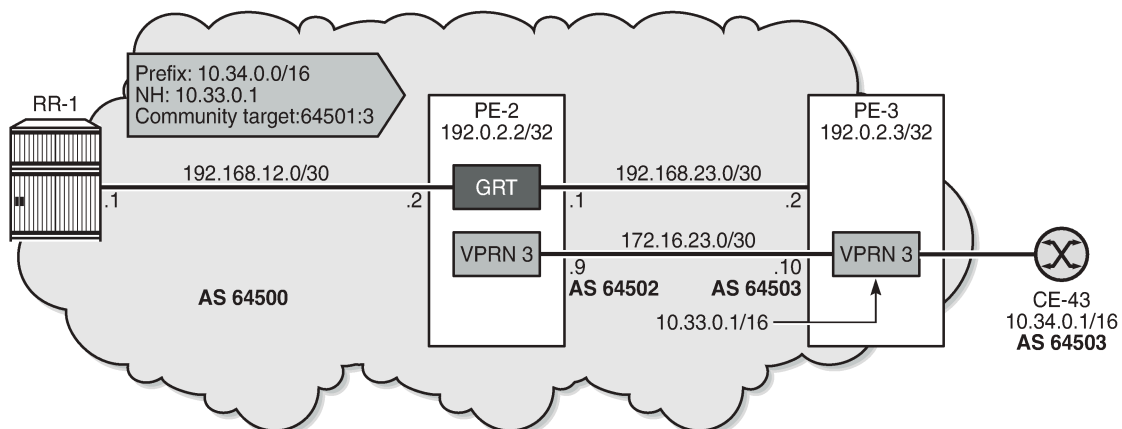
The route table for VPRN 2 on PE-2 now includes a BGP route for prefix 10.24.0.0/16:

```
*A:PE-2# show router 2 route-table
=====
Route Table (Service: 2)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]          Type  Proto  Age           Pref
Metric
-----
10.22.0.1/32
  lo1                               Local Local   00h38m32s    0
10.23.0.1/32
  192.0.2.3 (tunneled:SR-ISIS:524290) Remote BGP VPN 00h37m03s   170
10 10
10.24.0.0/16
  192.0.2.3 (tunneled:SR-ISIS:524290) Remote BGP 00h09m01s   170
10 10
-----
No. of Routes: 3
---snip---
```

Use case 3: BGP route leaked to VPRN 3 with next-hop resolved using eBGP

Figure 106: 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 106: 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
router Base
  policy-options
  begin
  prefix-list "10.33.0.0/16"
    prefix 10.33.0.0/16 longer
  exit
  policy-statement "export_10.33"
    entry 10
      from
        prefix-list "10.33.0.0/16"
      exit
      to
        protocol bgp
      exit
      action accept
      exit
    exit
  exit
  commit
exit
service
  vprn 3 name "VPRN 3" customer 1 create
  autonomous-system 64503
  route-distinguisher 64503:3
  vrf-target target:64500:3
  interface "lo1" create
    address 10.33.0.1/32
    loopback
  exit
  interface "int-VPRN3-PE-3-PE-2" create
    address 172.16.23.10/30
    sap 1/1/c1/2:3 create
  exit
  interface "int-VPRN3-PE-3-CE-43" create
    address 172.16.34.9/30
    sap 1/1/c1/1:3 create
  exit
  static-route-entry 10.34.0.0/16
    next-hop 172.16.34.10
    no shutdown
  exit
  bgp
    router-id 10.33.0.1
    split-horizon
    group "eBGP"
      peer-as 64502
      neighbor 172.16.23.9
      export "export_10.33"
    exit
  exit
  no shutdown
exit
no shutdown
```

```
exit
exit
```

The route table for VPRN 3 on PE-2 contains the loopback address from VPRN 3 on PE-3:

```
*A:PE-2# show router 3 route-table
=====
Route Table (Service: 3)
=====
Dest Prefix[Flags]                Type  Proto  Age           Pref
Next Hop[Interface Name]          Metric
-----
10.32.0.1/32                      Local Local  00h41m32s    0
    lo1                             0
10.33.0.1/32                    Remote BGP  00h40m33s   170
    172.16.23.10                 0
172.16.23.8/30                   Local Local  00h41m32s    0
    int-VPRN3-PE-2-PE-3             0
-----
No. of Routes: 3
---snip---
```

PE-2 receives the following BGP route with community "target:64500:3" from RR-1:

```
*A:PE-2# show router bgp routes community target:64500:3
=====
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     10.34.0.0/16          100        None
      10.33.0.1           None        0
      64503                -
-----
Routes : 1
=====
```

This route is leakable, but the next-hop 10.33.0.1 cannot be resolved in the base router of PE-2:

```
*A: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: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 Nexthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
u*>i	10.33.0.1/32 172.16.23.10 64503	None None	None 0 -

```
-----
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
  router Base
    policy-options
      begin
        community "target:64500:3"
          members "target:64500:3"
        exit
      policy-statement "leak-import-3"
        entry 10
          from
            community "target:64500:3"
          exit
          action accept
        exit
      exit
    default-action drop
  exit
exit
exit
```

This **leak-import** policy is applied in VPRN 3 and the **BGP next-hop-resolution** is set to **use-bgp-routes**:

```
# on PE-2:
configure
  service
    vprn "VPRN 3"
      autonomous-system 64502
      route-distinguisher 64502:3
      vrf-target target:64500:3
      bgp
        next-hop-resolution
          use-bgp-routes      # for BGP and BGP-VPN routes
        exit
        rib-management
          ipv4
            leak-import "leak-import-3"
          exit
        exit
      exit
    exit
  exit
```


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: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:PE-2# show router 3 route-table
=====
Route Table (Service: 3)
=====
Dest Prefix[Flags]                               Type  Proto  Age           Pref
  Next Hop[Interface Name]                       Metric
-----
10.32.0.1/32                                     Local  Local   00h46m21s    0
      lo1                                         0
10.33.0.1/32                                     Remote BGP    00h45m22s   170
      172.16.23.10                               0
10.34.0.0/16                                     Remote BGP 00h00m05s   170
      172.16.23.10                               0
172.16.23.8/30                                   Local  Local   00h46m21s    0
      int-VPRN3-PE-2-PE-3                         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

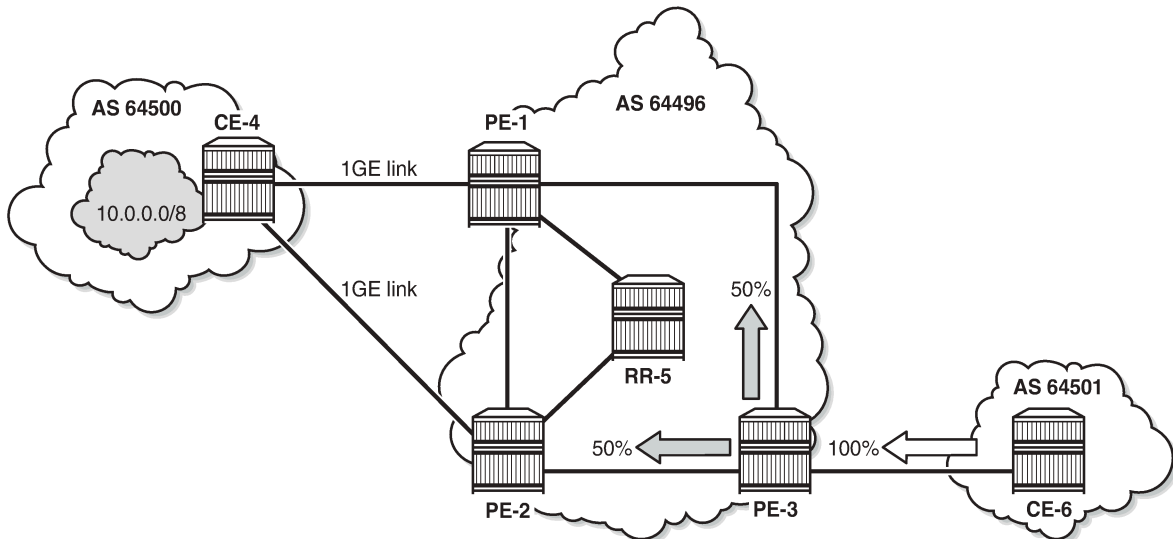
The information and configuration in this chapter was originally based on SR OS Release 15.0.R4. The CLI in the current edition is based on SR OS Release 23.3.R2.

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 107: 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.

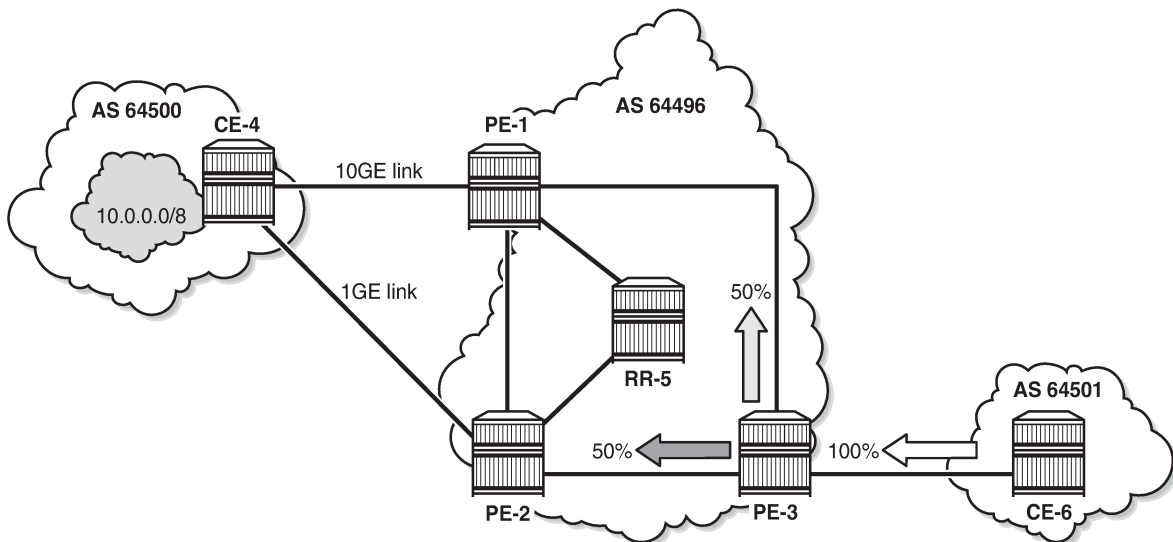
Figure 107: Standard ECMP - Equal Bandwidth Links



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 108: 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.

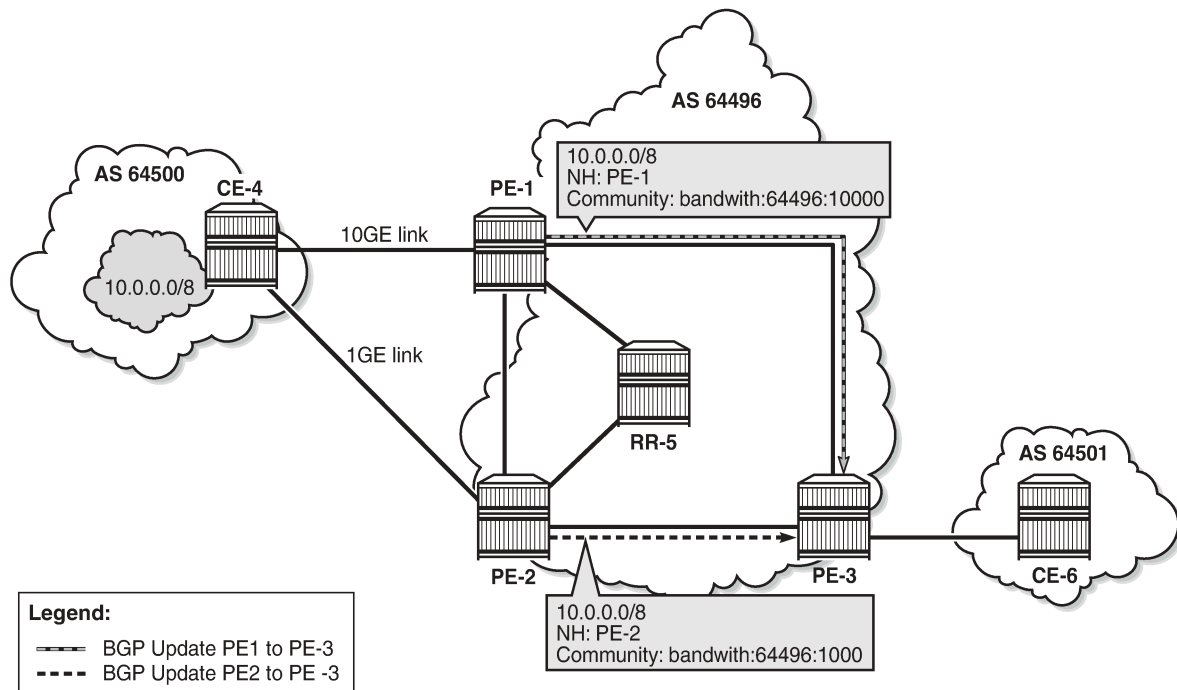
Figure 108: Standard ECMP - Unequal Bandwidth Links



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 109: 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.

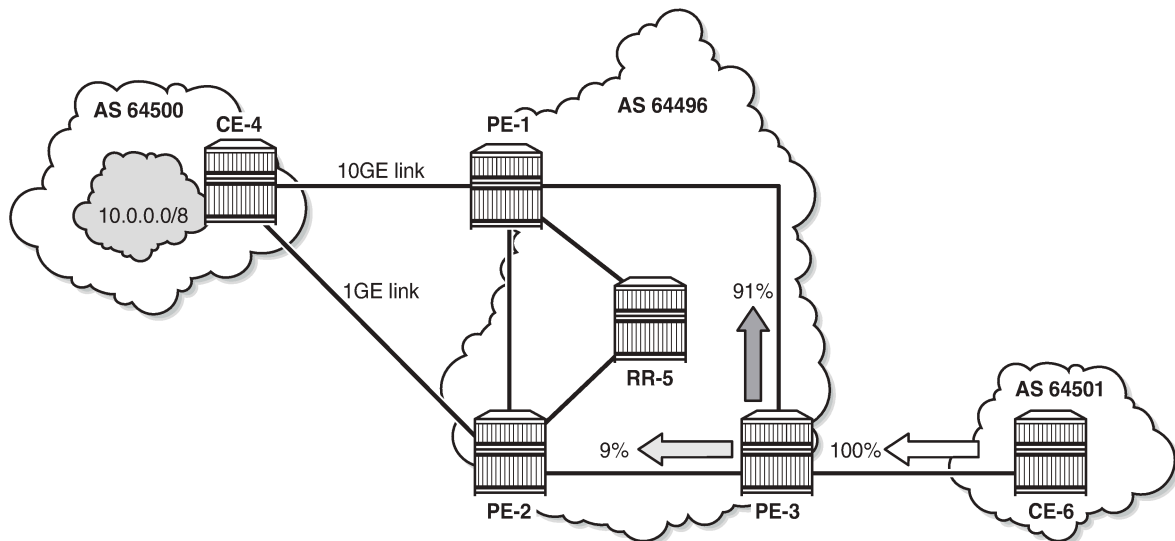
Figure 109: Link Bandwidth Extended Community Advertisement



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 110: 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.

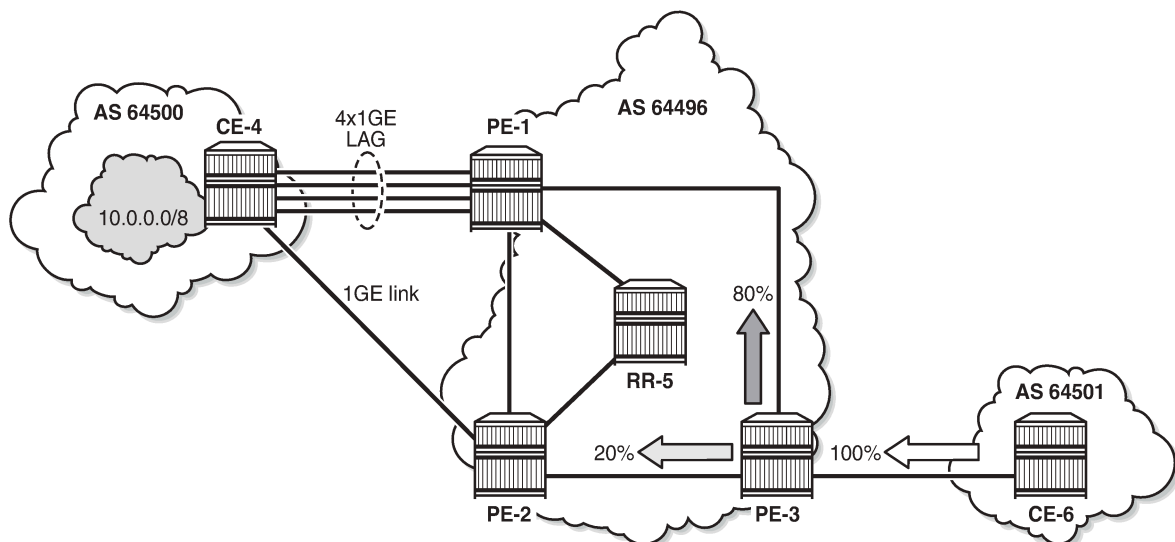
Figure 110: Weighted ECMP - Unequal Bandwidth Links



26857

Figure 111: 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.

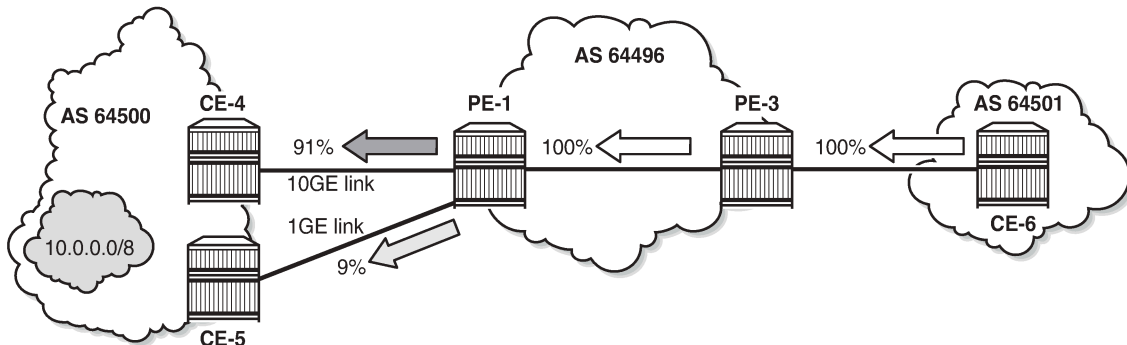
Figure 111: Weighted ECMP - Link Aggregation Group



26858

Figure 112: 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.

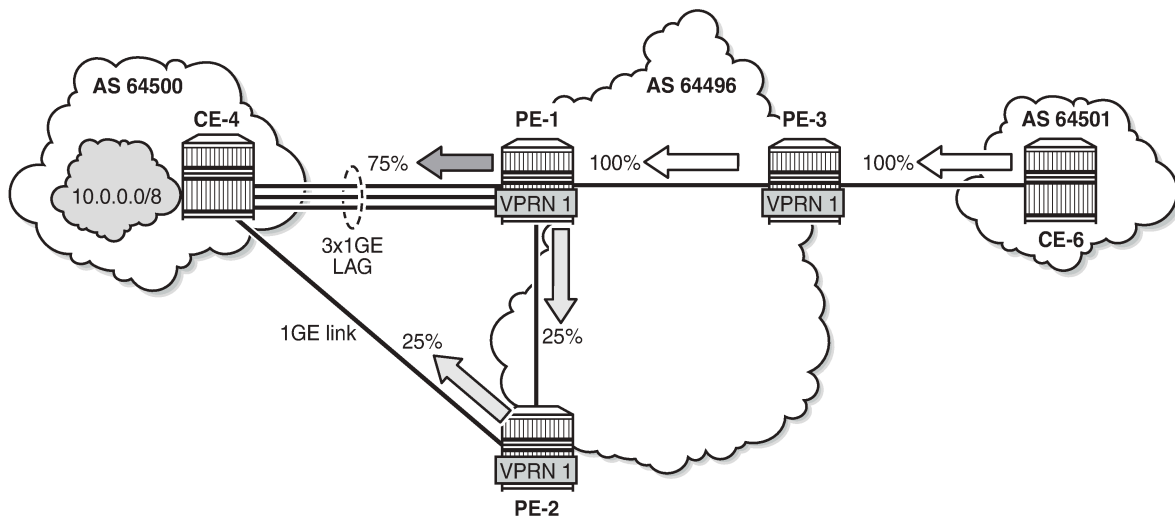
Figure 112: Standard ECMP - Unequal Bandwidth Links with eBGP



26859

Figure 113: 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 113: 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:

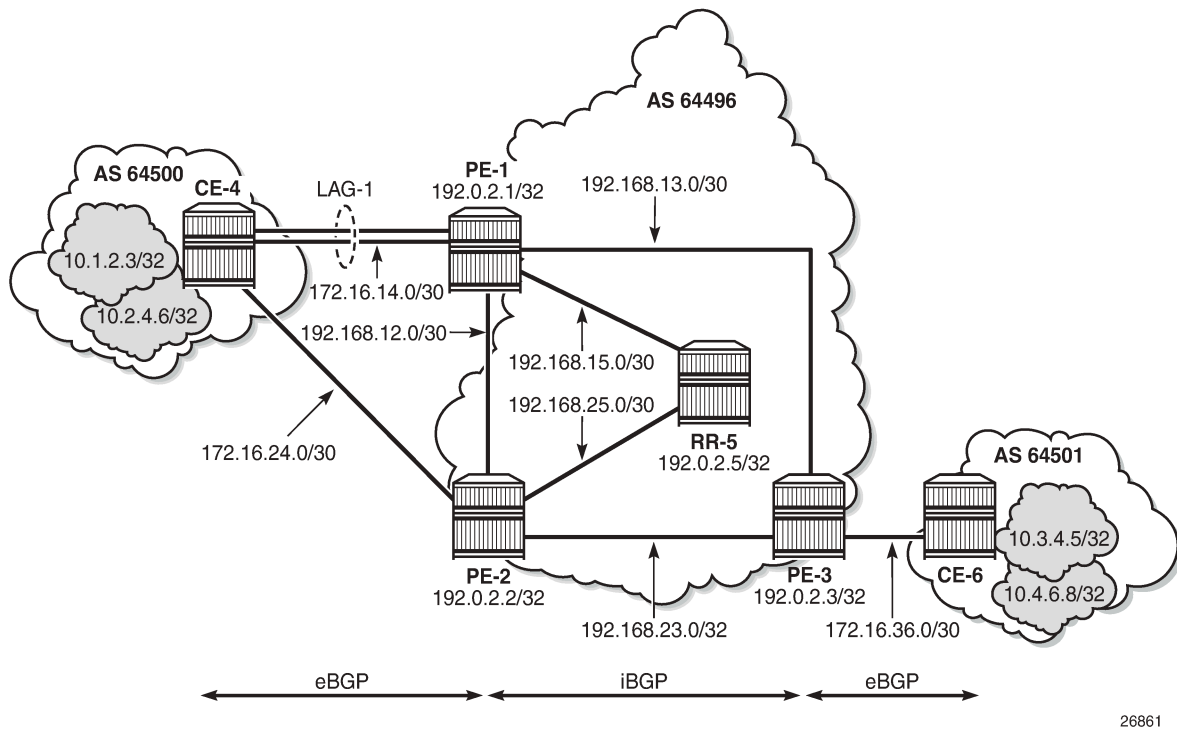
- [BGP Weighted ECMP for IPv4 Family using the link-bandwidth command](#)
- [BGP Weighted ECMP for IPv4 Family using BGP Import Policy](#)

[Figure 114: 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 114: 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"
  address 10.1.2.3/32
  loopback
  no shutdown
exit
interface "int-loopback-2"
address 10.2.4.6/32
loopback
```



```

        no shutdown
    exit
    autonomous-system 64500
    policy-options
        begin
        prefix-list "10.0.0.0/8"
            prefix 10.0.0.0/8 longer
        exit
        policy-statement "policy-export-bgp"
            entry 10
                from
                    prefix-list "10.0.0.0/8"
                exit
                action accept
            exit
        exit
    exit
    commit
exit
bgp
    rapid-withdrawal
    split-horizon
    group "eBGP"
        export "policy-export-bgp"
        peer-as 64496
        neighbor 172.16.14.1
        exit
        neighbor 172.16.24.1
        exit
    exit
    no shutdown
exit
exit all

```

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
            rapid-withdrawal
            split-horizon
            group "eBGP"
                peer-as 64500
                neighbor 172.16.14.2
            exit
        exit
        group "iBGP"
            family ipv4
            next-hop-self
            peer-as 64496
            add-paths
                ipv4 send 2 receive
            exit
            neighbor 192.0.2.5
        exit
    exit
    no shutdown
exit

```

```
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
      rapid-withdrawal
      split-horizon
      group "iBGP"
        family ipv4
          cluster 5.5.5.5
          peer-as 64496
          add-paths
            ipv4 send 2 receive
          exit
        neighbor 192.0.2.1
        exit
        neighbor 192.0.2.2
        exit
        neighbor 192.0.2.3
        exit
      exit
    no shutdown
  exit
exit all
```

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: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                             1          10
      64500                                   -
*i    10.1.2.3/32                             100        None
      192.0.2.2                             11         10
      64500                                   -
u*>i  10.2.4.6/32                             100        None
```

```

192.0.2.1          2          10
64500             -
*i 10.2.4.6/32    100         None
192.0.2.2        12          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 ecmp 2

configure router bgp multi-path maximum-paths 2

```

As a result, PE-3 installs the routes from PE-2 as active, in addition to those from PE-1:

```

*A: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                               1          10
      64500                                    -
u*>i 10.1.2.3/32                             100        None
      192.0.2.2                               11         10
      64500                                    -
u*>i 10.2.4.6/32                             100        None
      192.0.2.1                               2          10
      64500                                    -
u*>i 10.2.4.6/32                             100        None
      192.0.2.2                               12         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: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     00h00m40s    170
    192.168.13.1                                10
10.1.2.3/32                                     Remote BGP     00h00m40s    170
    192.168.23.1                                10
10.2.4.6/32                                     Remote BGP     00h00m40s    170
    192.168.13.1                                10
10.2.4.6/32                                     Remote BGP     00h00m40s    170
    192.168.23.1                                10
10.3.4.5/32                                     Remote BGP     00h05m12s    170
    172.16.36.2                                 0
10.4.6.8/32                                     Remote BGP     00h05m12s    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 (do not forget to add the keyword "expression" after the match statement).

```
*A:PE-3# show router bgp routes 10.1.2.3/32 hunt brief | match "^Nexthop |Community"
expression
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:PE-3# show router fib 1 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:PE-3# show router bgp routes 10.1.2.3/32 hunt brief | match "^Nexthop |Community"
expression
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:PE-3# show router fib 1 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:CE-6# ping 10.1.2.3 source 10.3.4.5 size 1200 count 100000 rapid
```

Second session:

```
*A:CE-6# ping 10.1.2.3 source 10.3.4.5 size 1200 count 100000 rapid
```

Third session:

```
*A:CE-6# ping 10.1.2.3 source 10.4.6.8 size 1200 count 100000 rapid
```

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: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	301	201
	375128	250128
	0	~0.00 0 ~0.00
1/1/c5/1	1	1
	128	128
	0	~0.00 0 ~0.00
Totals	302	202
	375256	250256
	0	~0.00 0 ~0.00

```

-----
At time t = 9 sec (Mode: Rate)
-----
1/1/c2/1      301                201
               375128            250128
               0                ~0.00  0                ~0.00
1/1/c5/1      1
               128              128
               0                ~0.00  0                ~0.00
-----
Totals        302                202
               375256            250256
               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: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                125000
Packets               0                 100
Errors                0                 0
Bits                  0                1000000
Utilization (% of port capacity) 0.00             ~0.00
-----
At time t = 9 sec (Mode: Rate)
-----
Octets                0                125000
Packets               0                 100
Errors                0                 0
Bits                  0                1000000
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 bgp group "eBGP" link-bandwidth no add-to-received-ebgp

```

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
  router Base
    policy-options
      begin

```

```

prefix-list "10.0.0.0/8"
  prefix 10.0.0.0/8 longer
exit
community "bandwidth-4G" members "bandwidth:64500:4000"
policy-statement "policy-import-bandwidth-4G"
  entry 10
    from
      prefix-list "10.0.0.0/8"
    exit
    action accept
      community add "bandwidth-4G"
    exit
  exit
exit
commit
exit all

```

The policy is applied on PE-1 for the eBGP group in the import direction:

```

# on PE-1:
configure router bgp group "eBGP" import "policy-import-bandwidth-4G"

```

The following policy is configured on PE-2 to manually add the Link Bandwidth Extended Community "bandwidth:64500:2000" to routes received from CE-4:

```

# on PE-2:
configure
  router Base
    policy-options
      begin
        prefix-list "10.0.0.0/8"
          prefix 10.0.0.0/8 longer
        exit
        community "bandwidth-2G" members "bandwidth:64500:2000"
        policy-statement "policy-import-bandwidth-2G"
          entry 10
            from
              prefix-list "10.0.0.0/8"
            exit
            action accept
              community add "bandwidth-2G"
            exit
          exit
        exit
      exit
    commit
  exit all

```

The policy is applied on PE-2 for the eBGP group in the import direction:

```

# on PE-2:
configure router bgp group "eBGP" import "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:PE-3# show router bgp routes 10.1.2.3/32 hunt brief | match "^Nexthop |Community"
  expression
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:PE-3# show router fib 1 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 CLI in the current edition corresponds to SR OS Release 20.7.R1.

Overview

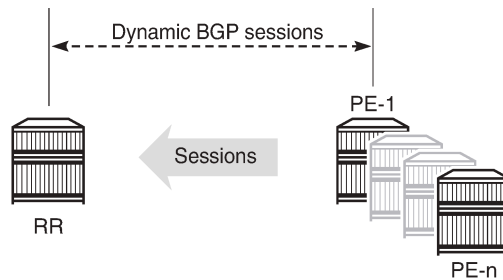
SR OS supports static and dynamic BGP sessions, where the static sessions are initiated toward explicitly configured non-passive neighbors, which are identified through an IPv4 or IPv6 address.

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 115: 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 115: 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

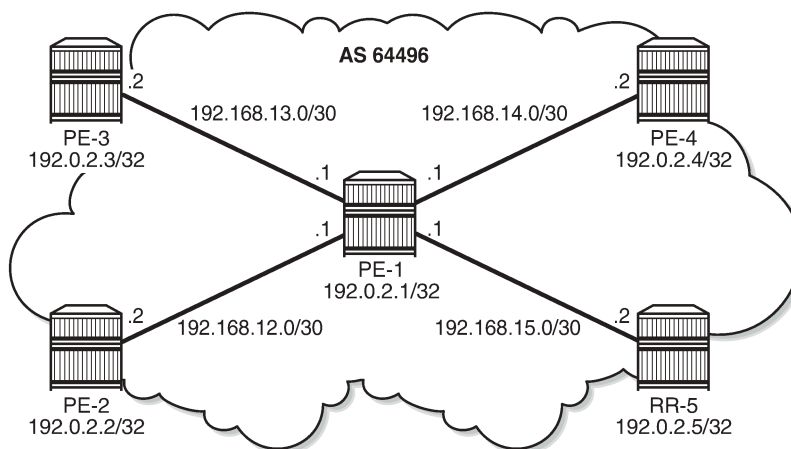
Figure 116: Dynamic BGP peers shows the example topology, and has the following characteristics:

- 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 116: 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
```

```

group "iBGP"
  peer-as 64496
  neighbor 192.0.2.5
  exit
exit
no shutdown
exit

```

The BGP configuration on the other route reflector clients is the same as on PE-1.

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
      dynamic-neighbor-limit 20
      group "iBGP"
        cluster 5.5.5.5
        peer-as 64496
        dynamic-peer-limit 10
        dynamic-neighbor
          match
            prefix 192.0.2.0/24
            allowed-peer-as 64496
          exit
        exit
      exit
    exit
  no shutdown
exit

```

Dynamic neighbors are shown with the "D" flag, as follows:

```

*A: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: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
    bgp
      group "iBGP"
        cluster 5.5.5.5
        peer-as 64496
        dynamic-neighbor
          match
            prefix 192.0.2.0/24
            allowed-peer-as 64496
          exit
        exit
      exit
    neighbor 192.0.2.1
      keepalive 20
      hold-time 60
    exit
  exit
no shutdown
exit
```

Therefore, the properties of BGP group iBGP are as follows:

```
*A: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 Accept
Export Policy   : None Specified - Default Accept
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: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: moved from higher state ACTIVE to lower
state IDLE due to event 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: remote end closed connection"

71 2020/08/19 16:41:37.225 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 CONFIG_CHG"

70 2020/08/19 16:41:37.224 CEST WARNING: BGP #2039 Base Peer 1: 192.0.2.1
"(ASN 64496) VR 1: Group iBGP: Peer 192.0.2.1: moved from higher state CONNECT
to lower state IDLE due to event CONFIG_CHG"

69 2020/08/19 16:41:37.214 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 CONFIG_CHG"

68 2020/08/19 16:41:37.214 CEST WARNING: BGP #2039 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: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: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
    exit
```

A hard reset of a running BGP session will result in that BGP session being torn down, as follows:

```
*A: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"

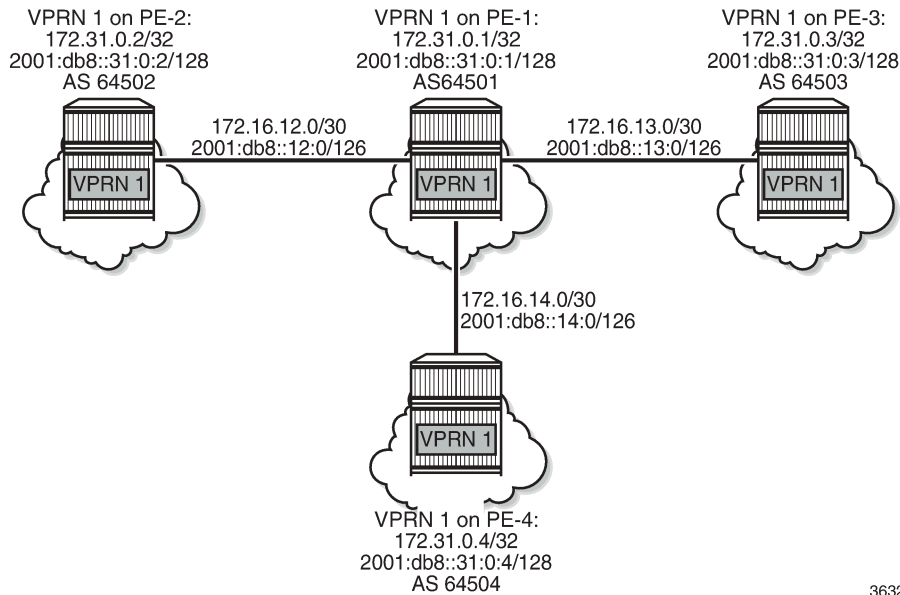
78 2020/08/19 17:12:39.574 CEST WARNING: BGP #2039 Base Peer 1: 192.0.2.4
"(ASN 64496) VR 1: Group iBGP: Peer 192.0.2.4: moved from higher state ESTABLISHED
to lower state IDLE due to event ADMIN_RESET_HARD"

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 117: 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 117: 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 1 name "VPRN 1" customer 1 create
      autonomous-system 64502
      router-id 172.31.0.2
      route-distinguisher 1:1
      vrf-target target:1:1
      interface "int-VPRN1-PE-2-PE-1" create
        address 172.16.12.2/30
        ipv6
          address 2001:db8::12:2/126
        exit
        sap 1/1/1:1 create
        exit
      exit
    interface "system" create
      address 172.31.0.2/32
      ipv6
        address 2001:db8::31:0:2/128
      exit
    loopback
  exit
  bgp
    router-id 172.31.0.2
    split-horizon
    group "eBGPv4"
      family ipv4
      next-hop-self
      peer-as 64501
      neighbor 172.16.12.1
```

```

        export "exp-vprn-1-v4"
    exit
exit
group "eBGPv6"
    family ipv6
    next-hop-self
    peer-as 64501
        neighbor 2001:db8::12:1
        export "exp-vprn-1-v6"
    exit
exit
exit
exit
no shutdown

```

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 1 name "VPRN 1" customer 1 create
        bgp
            router-id 172.31.0.1
            split-horizon
            group "eBGPv4"
                family ipv4
                next-hop-self
                export "exp-vprn-1-v4"
                dynamic-neighbor-limit 10
                dynamic-neighbor
                    match
                        prefix 172.16.0.0/16
                        allowed-peer-as 64502
                        allowed-peer-as 64504
                    exit
                exit
            exit
        exit
    group "eBGPv6"
        family ipv6
        next-hop-self
        export "exp-vprn-1-v6"
        dynamic-neighbor-limit 10
        dynamic-neighbor
            match
                prefix 2001:db8::/107
                allowed-peer-as 64502 max 64503
            exit
        exit
    exit
exit
exit

```

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: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: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  : None Specified - Default Accept
Export Policy   : exp-vprn-1-v4
                : Default Accept
---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  : None Specified - Default Accept
Export Policy  : exp-vprn-1-v6
                : Default Accept
---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 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 classic CLI, for backward compatibility reasons, this approach is not followed by default (**no ebgp-default-reject-policy**). This insecure default to advertise and receive all routes is not compliant with RFC 8212, *Default External BGP (EBGP) Route Propagation Behavior without Policies*. The secure behavior must be enabled using the **ebgp-default-reject-policy** command, which 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:

```
*A:PE-2# configure router bgp group "eBGP" ebgp-default-reject-policy
- no ebgp-default-reject-policy
- ebgp-default-reject-policy [import] [export]
<import>           : keyword
<export>           : keyword
```

The eBGP default reject policy is the last policy in a policy chain.



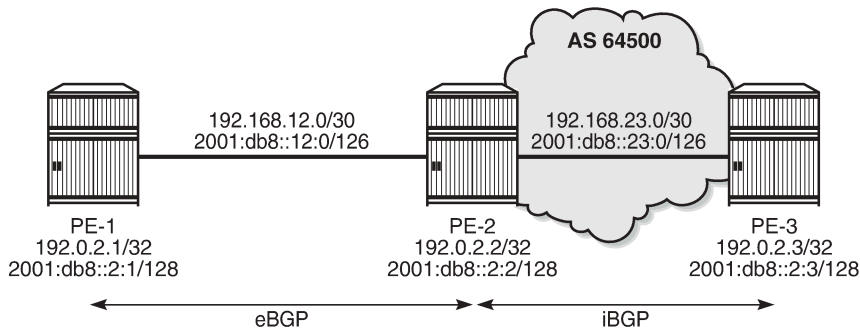
Note:

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 118: 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 118: Example topology



36517

The initial configuration includes:

- Cards, MDAs, ports
- Router interfaces
- SR-ISIS on PE-2 and PE-3 in AS 64500

Default in classic CLI: no eBGP default reject policy

On PE-1, BGP is configured as follows:

```
# on PE-1:
configure
router Base
  bgp
    split-horizon
    group "eBGP"
      local-as 64501
      peer-as 64500
      neighbor 192.168.12.2
        family ipv4 ipv6 label-ipv4 label-ipv6
        export "export-10.1" "export-10.2" "export-10.131"
          "export-10.132"
    exit
  exit
```

On PE-2, BGP is configured as follows:

```
# on PE-2:
configure
router Base
  bgp
    split-horizon
    next-hop-resolution
```



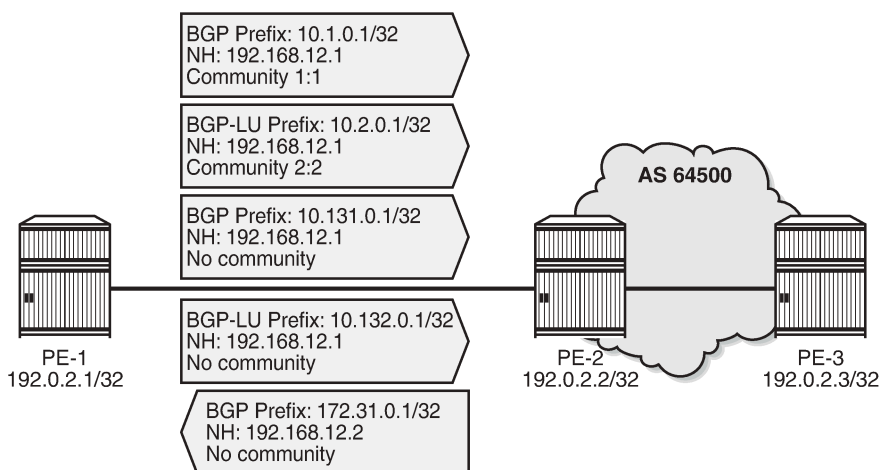
```

labeled-routes
  transport-tunnel
    family label-ipv4
      resolution-filter
        no ldp
        sr-isis
      exit
    resolution filter
  exit
exit
exit
exit
group "eBGP"
  local-as 64500
  peer-as 64501
  neighbor 192.168.12.1
    family ipv4 ipv6 label-ipv4 label-ipv6
    export "export-bgp"
  exit
exit
group "iBGP-IPv4"
  family ipv4 label-ipv4
  peer-as 64500
  neighbor 192.0.2.3
    next-hop-self
  exit
exit
group "iBGP-IPv6"
  family ipv6 label-ipv6
  peer-as 64500
  neighbor 2001:db8::2:3
    next-hop-self
  exit
exit
exit

```

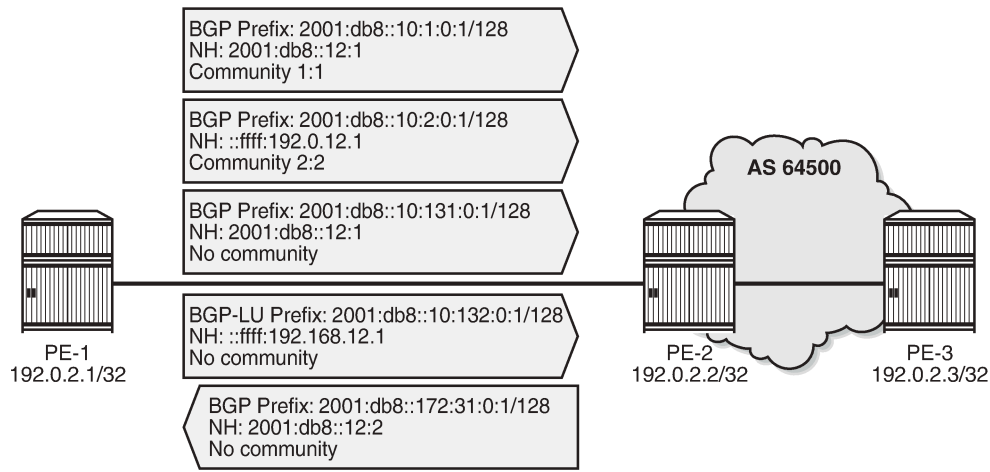
Figure 119: Advertised BGP and BGP-LU IPv4 routes and Figure 120: Advertised BGP and BGP-LU IPv6 routes show the advertised BGP and BGP Labeled Unicast (BGP-LU) routes between PE-1 and PE-2:

Figure 119: Advertised BGP and BGP-LU IPv4 routes



36518

Figure 120: Advertised BGP and BGP-LU IPv6 routes



36519

By default, in classic CLI, no eBGP default reject policy is used. When no eBGP import-policy is configured on PE-2, any route received from an eBGP peer is accepted, as follows:

```
*A:PE-2# show router bgp neighbor 192.168.12.1 | match "Import Policy"
Import Policy      : None Specified - Default Accept
```

In addition, 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:PE-2# show router bgp neighbor 192.0.2.3 | match "Export Policy"
Export Policy      : None Specified - Default Accept
```

The following BGP summary on PE-2 shows that all routes received from eBGP peer 192.168.12.1 are received, accepted and advertised to PE-3:

```
*A: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 0/0/2 (IPv4)
              12   0
              0/0/2 (Lbl-IPv4)
192.168.12.1
Def. Instance  64501     16   0 00h01m55s 2/2/1 (IPv4)
              11   0
              2/2/1 (IPv6)
              2/2/0 (Lbl-IPv4)
              2/2/0 (Lbl-IPv6)

2001:db8::2:3
```

```
Def. Instance 64500      7  0 00h01m34s 0/0/2 (IPv6)
                12  0                0/0/2 (Lbl-IPv6)
-----
```

The following output shows that both received BGP routes are used:

```
*A: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                                     -
u*>i  10.131.0.1/32                            None       None
      192.168.12.1                             None       0
      64501                                     -
-----
Routes : 2
=====
```

In a similar way, two received routes are active for the **ipv6**, **label-ipv4**, and **label-ipv6** address families.

EBGP default reject policy for import and export

On PE-1 and PE-2, the eBGP default reject policy is configured in the group "eBGP", both for import and export, as follows:

```
# on PE-1, PE-2:
configure
  router Base
    bgp
      group "eBGP"
        ebgp-default-reject-policy import export
      exit
```

Both PE-1 and PE-2 have export policies configured and the same prefixes will be advertised. However, the received routes will be rejected because no import policies are configured:

```
*A:PE-2# show router bgp neighbor 192.168.12.1 | match "Import Policy"
Import Policy      : None Specified - Default Reject
*A:PE-2# show router bgp neighbor 192.168.12.1 | match "Export Policy"
Export Policy      : export-bgp
```

The following BGP summary on PE-2 shows that the same number of routes is received from eBGP peer 192.168.12.1, but these routes are rejected:

```
*A: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      38   0 00h17m29s 0/0/0 (IPv4)
                44   0           0/0/0 (Lbl-IPv4)
192.168.12.1
Def. Instance 64501      48   0 00h17m50s 2/0/1 (IPv4)
                43   0           2/0/1 (IPv6)
                2/0/0 (Lbl-IPv4)
                2/0/0 (Lbl-IPv6)
2001:db8::2:3
Def. Instance 64500      38   0 00h17m29s 0/0/0 (IPv6)
                45   0           0/0/0 (Lbl-IPv6)
-----
```

The following shows that the received BGP routes are invalid:

```
*A: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:PE-2# show router bgp routes hunt | match Flags
Flags          : Invalid IGP Rejected
Flags          : Invalid IGP Rejected

*A:PE-2# show router bgp routes ipv6 hunt | match Flags
```

```
Flags          : Invalid IGP Rejected
Flags          : Invalid IGP Rejected

*A:PE-2# show router bgp routes label-ipv4 hunt | match Flags
Flags          : Invalid IGP Rejected
Flags          : Invalid IGP Rejected

*A: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 accepts incoming routes with communities "1:1" or "2:2":

```
# on PE-2:
configure
  router Base
    policy-options
      begin
        community "1:1"
          members "1:1"
        exit
        community "2:2"
          members "2:2"
        exit
      policy-statement "import-1:1-2:2"
        entry 10
          from
            community "1:1"
          exit
          action accept
          exit
        exit
        entry 20
          from
            community "2:2"
          exit
          action accept
          exit
        exit
      exit
    exit
  commit
exit
bgp
  group "eBGP"
    import "import-1:1-2:2"
  exit
```

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: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: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 0/0/1 (Lbl-IPv4)
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 0/0/1 (Lbl-IPv6)
-----

```

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: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: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: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: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 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 (Layer 3 Services)*, 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
  bgp
    next-hop-resolution
      labeled-routes
        allow-static
    exit
  exit
exit
```

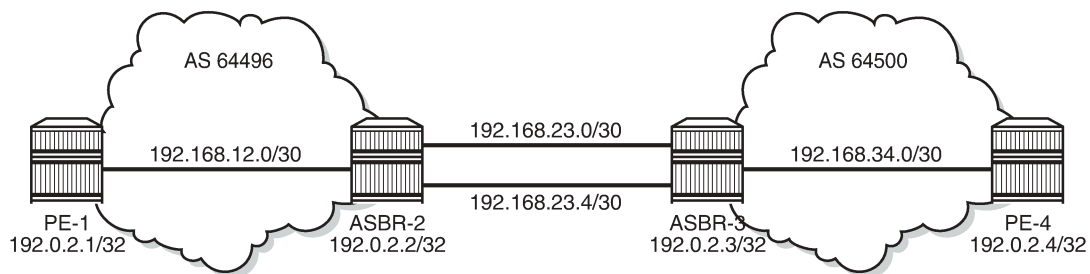
Even with this feature enabled, the system will first try to resolve the BGP next-hop to LDP or RSVP LSPs before the IP route table is attempted. The option is supported for the following address families:

- Labeled IPv4 routes
- VPN-IPv4 and VPN-IPv6 routes

Configuration

Figure 121: 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 121: Example topology



26311

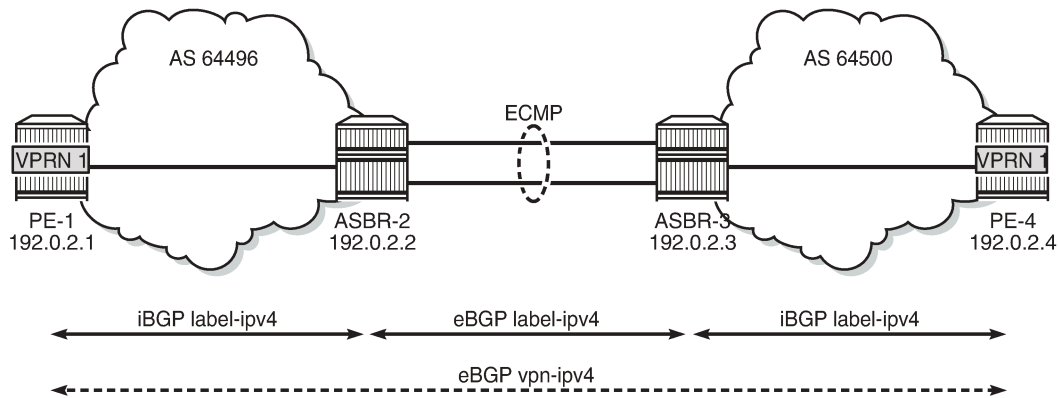
The initial configuration on the nodes includes the following:

- 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 122: 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 122: 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
  autonomous-system 64496
  bgp
    split-horizon
    group "iBGP"
      export "export-bgp"
      peer-as 64496
      neighbor 192.0.2.2
        family label-ipv4
      exit
    exit
  exit
```

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
router
  policy-options
    begin
    prefix-list "PE-sys"
      prefix 192.0.2.0/28 prefix-length-range 32-32
    exit
    policy-statement "export-bgp"
      entry 10
        from
          protocol direct
          prefix-list "PE-sys"
        exit
        action accept
      exit
    exit
  exit
exit
commit
```

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
router
  autonomous-system 64496
  ecmp 2
  bgp
    multi-path
      maximum-paths 2 ebgp 2
    exit
    split-horizon
    group "eBGP"
      peer-as 64500
      neighbor 192.0.2.3
        family label-ipv4
        advertise-inactive
      exit
    exit
    group "iBGP"
      peer-as 64496
      neighbor 192.0.2.1
        family label-ipv4
      exit
    exit
  exit
exit
```

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** 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. This way, no export policy is required for the eBGP session between ASBRs. 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
  bgp
    group "eBGP_multihop"
      family vpn-ipv4
      peer-as 64500
      local-address 192.0.2.1
      neighbor 192.0.2.4
        multihop 10
        vpn-apply-export
        export "EBGP-VPN-IPv4"
      exit
    exit
  exit
```

The export policy "EBGP-VPN-IPv4" is not required in this example, but usually some export policy would be used.

On PE-1, VPRN 1 is configured with loopback address 10.1.1.1/32, as follows:

```
# on PE-1:
configure
service
  vprn 1 name "VPRN 1" customer 1 create
  route-distinguisher 64496:1
  auto-bind-tunnel
  resolution-filter
  ldp
  exit
  resolution filter
exit
vrf-target target:64496:1
interface "loopback" create
  address 10.1.1.1/32
  loopback
exit
no shutdown
```

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*, 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: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: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: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                               0
192.0.2.2/32                      Remote  ISIS  00h10m40s  15
  192.168.12.2                       10
192.168.12.0/30                   Local  Local  00h10m50s   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
=====
```

```
*A: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: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
  static-route-entry 192.0.2.3/32
    next-hop 192.168.23.2
    no shutdown
  exit
  next-hop 192.168.23.6
  no shutdown
  exit
exit
    
```

The routing table in ASBR-2 contains two routes toward ASBR-3, as follows:

```

*A: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: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: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 Port     : 0
Local AS     : 64496           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: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
    
```

```
i      192.0.2.4/32          None      None
      192.0.2.3           None      0
      64500                None      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: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
  ldp
  interface-parameters
    interface "int-ASBR-2-ASBR-3_1st" dual-stack
      ipv4
        no shutdown
      exit
    no shutdown
```



```

        exit
        interface "int-ASBR-2-ASBR-3_2nd" dual-stack
        ipv4
            no shutdown
        exit
        no shutdown
    exit
exit
exit

```

ASBR-2 now has a valid, best, and used route for prefix 192.0.2.4/32, as follows:

```

*A: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 has a valid route for prefix 192.0.2.4/32, as follows:

```

*A: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                                    524282
-----
Routes : 1
=====

```

The following routing table shows that PE-1 has a BGP labeled route toward PE-4:

```
*A: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   00h43m23s    0
  system                                           0
192.0.2.2/32                                       Remote  ISIS    00h43m13s    15
  192.168.12.2                                     10
192.0.2.4/32                                       Remote  BGP_LABEL 00h17m57s    170
  192.0.2.2 (tunneled)                               10
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: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: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
-----
u*>i 64500:1:10.2.2.2/32                     None     None
      192.0.2.4                             None     0
      64500                                  None     524284
-----
Routes : 1
=====
    
```

The routing table for VPRN 1 on PE-1 includes a BGP-VPN route to PE-4, as follows:

```

*A: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   00h44m02s    0
  loopback
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
  ldp
  interface-parameters
    interface "int-ASBR-2-ASBR-3_1st" shutdown
    no interface "int-ASBR-2-ASBR-3_1st"
    interface "int-ASBR-2-ASBR-3_2nd" shutdown
    no interface "int-ASBR-2-ASBR-3_2nd"
  exit
    
```

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
  bgp
    next-hop-resolution
    labeled-routes
      allow-static
    exit
  exit
exit

```

On ASBR-2, the labeled IPv4 route for prefix 192.0.2.4/32 is now valid, best, and used, as follows:

```

*A: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: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: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                                           0
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: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: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: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 and configuration in this chapter are based on SR OS Release 20.7.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

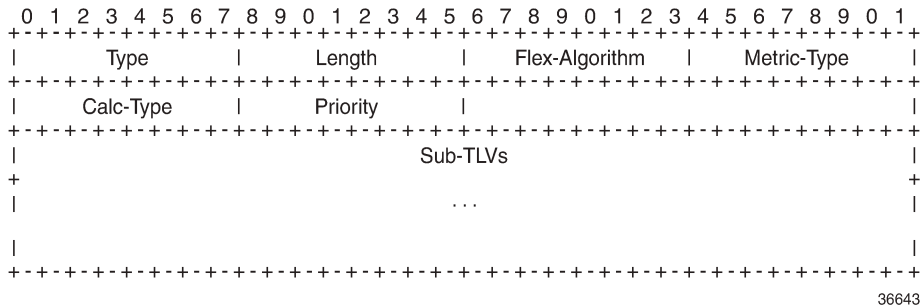
A FAD is the construct that identifies how a path for a Flex-Algorithm will be computed, and consists of three components:

- 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 123: IS-IS FAD sub-TLV](#).

Figure 123: 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

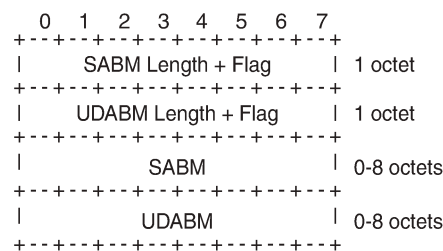
The ASLA sub-TLV contains Link Attribute sub-sub-TLVs, the format of which matches the existing formats defined in RFC 5305, RFC 7308, and RFC 8570. The Application-Specific SRLG TLV encodes link identifier sub-TLVs, such as IPv4/IPv6 Interface address, IPv4/IPv6 Neighbor address, and Link Local/Remote 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 124: Application Identifier Bit Mask](#) is used by both the ASLA sub-TLV and the Application-Specific SRLG TLV.

Figure 124: 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

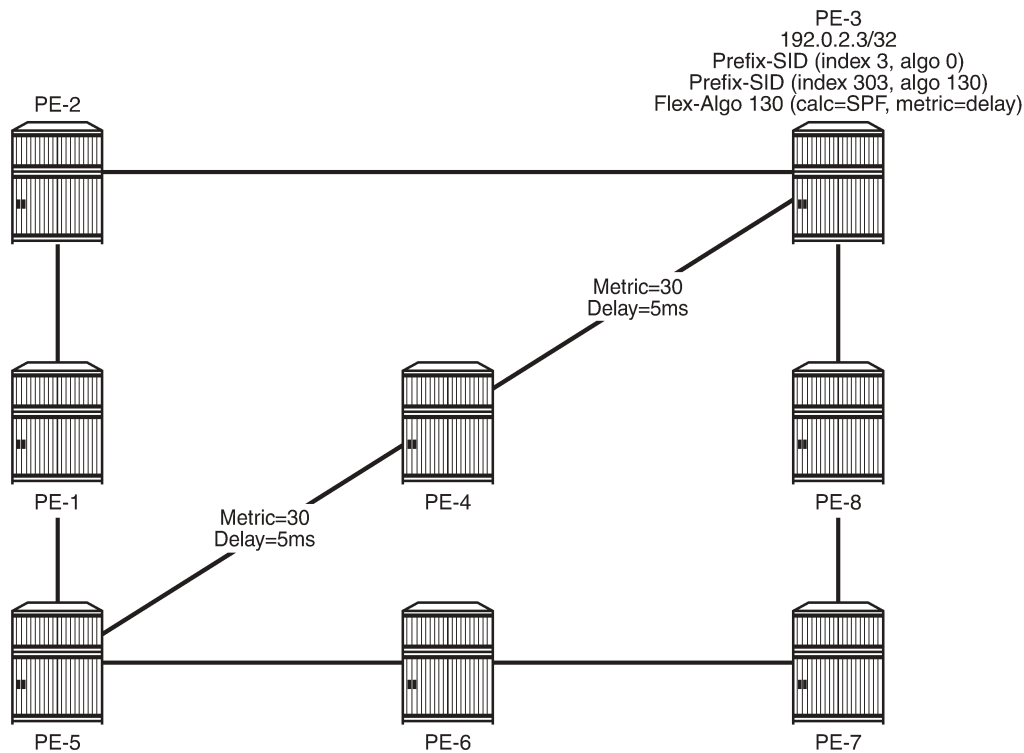
A router may use various algorithms when calculating reachability to other nodes or prefixes attached to those nodes. RFC 8667, *IS-IS extensions for SR*, describes the use of the SR-Algorithm sub-TLV (carried as part of the Router Capabilities TLV) to advertise the algorithms that the router can support. 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 125: 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 125: Flexible Algorithm example in an SR-MPLS domain



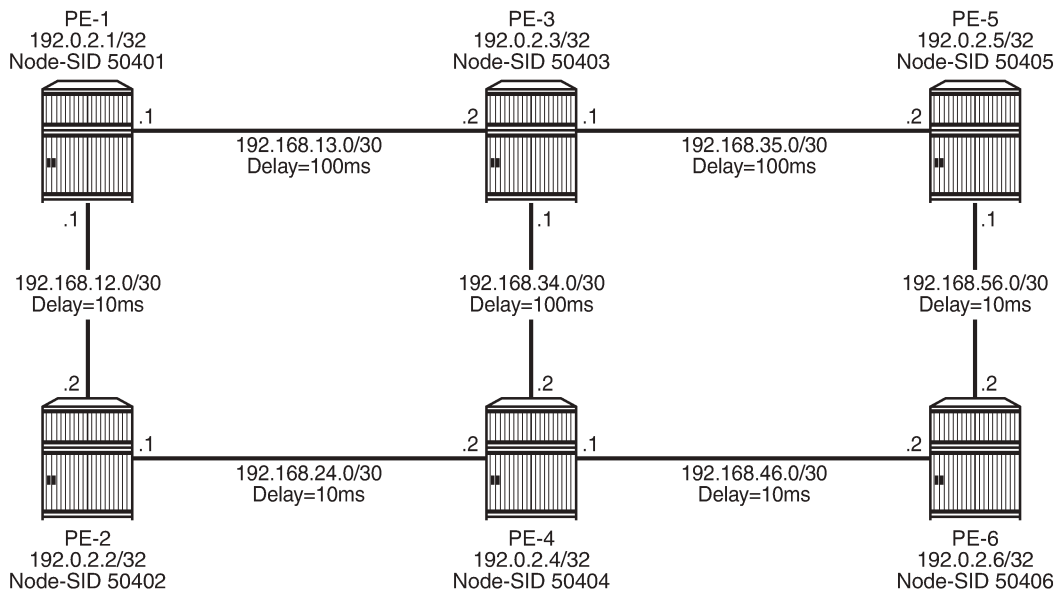
36645

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 126: 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 126: 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 126: 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
    exit
  exit
  no shutdown
exit
interface "int-PE-1-PE-3"
  if-attribute
    delay
      static 100000
  exit
exit
no shutdown
exit
```

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
    exit
  exit
```

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:PE-1# show router isis database detail PE-1.00-00

=====
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
    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
  router Base
    flexible-algorithm-definitions
      flex-algo "FlexAlgo-128" create
        description "FlexAlgo-128-Delay-Metric"
        metric-type delay
        no shutdown
      exit
    exit
  exit

```

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
        flex-algo 128
          advertise "FlexAlgo-128"
          participate
          loopfree-alternates
        exit
      exit
    no shutdown
  exit
exit
```

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
        flex-algo 128
          participate
          loopfree-alternates
        exit
      exit
    no shutdown
  exit
exit
```

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: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"
ipv4-node-sid label 50405
passive
flex-algo 128
ipv4-node-sidlabel 54405
exit
no shutdown
exit
exit
exit

```

When configured, the additional Prefix SID advertisement can be viewed in the PE-5 advertised IS-IS LSP.

```

*A: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: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: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-route-entry 172.16.0.1/32
      indirect 192.0.2.5
        tunnel-next-hop
          resolution-filter
            sr-isis
          exit
          flex-algo 128
          resolution filter
        exit
      no shutdown
    exit
  exit
```

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:PE-1# show router route-table 172.16.0.1/32
=====
Route Table (Router: Base)
=====
Dest Prefix[Flags]
Next Hop[Interface Name]          Type   Proto   Age           Pref
                                   Metric
-----
172.16.0.1/32
  192.0.2.5 (tunneled:SR-ISIS:524298)  Remote Static  01h29m00s  5
                                   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"
      hop 1 sid-label 54405
      no shutdown
    exit
  lsp "PE-1-PE-5-SR-TE-FlexAlgo128" sr-te
    to 192.0.2.5
    max-sr-labels 1 additional-frr-labels 2
    primary "FlexAlgo-128"
    no shutdown
  exit
exit
exit
```

First, verification is obtained that the SR-TE LSP is administratively and operationally up.

```
*A: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: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 1 name "VPRN 1" customer 1 create
  auto-bind-tunnel
  resolution-filter
    sr-te
  exit
  resolution filter
  exit
  no shutdown
exit

```

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: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 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
    if-attribute
      admin-group "blue" value 10
    exit
  interface "int-PE-1-PE-3"
    if-attribute
      admin-group "blue"
      delay
        static 100000
    exit
  exit
  no shutdown

```

```
exit
```

**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
  router Base
    flexible-algorithm-definitions
      flex-algo "FlexAlgo-129" create
        description "FlexAlgo-129-AG-Blue"
        exclude
          admin-group "blue"
        exit
      no shutdown
    exit
  exit
exit
```

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
          advertise "FlexAlgo-129"
          participate
          loopfree-alternates
        exit
      exit
    no shutdown
  exit
exit
```

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"
      ipv4-node-sid label 50405
      passive
      flex-algo 129
        ipv4-node-sid label 54415
      exit
    no shutdown
  exit
exit

```

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:PE-1# show router isis prefix-sids algo 129
```

```
=====
Rtr Base ISIS Instance 0 Prefix/SID Table
=====
```

Prefix	SID	Lvl/Typ Algo	SRMS MT	AdvRtr Flags
192.0.2.5/32	4415	2/Int. 129	N 0	PE-5 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:PE-1# show router tunnel-table 192.0.2.5/32 protocol isis
```

```
=====
IPv4 Tunnel Table (Router: Base)
=====
```

Destination Color	Owner	Encap	TunnelId	Pref	Nexthop	Metric
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

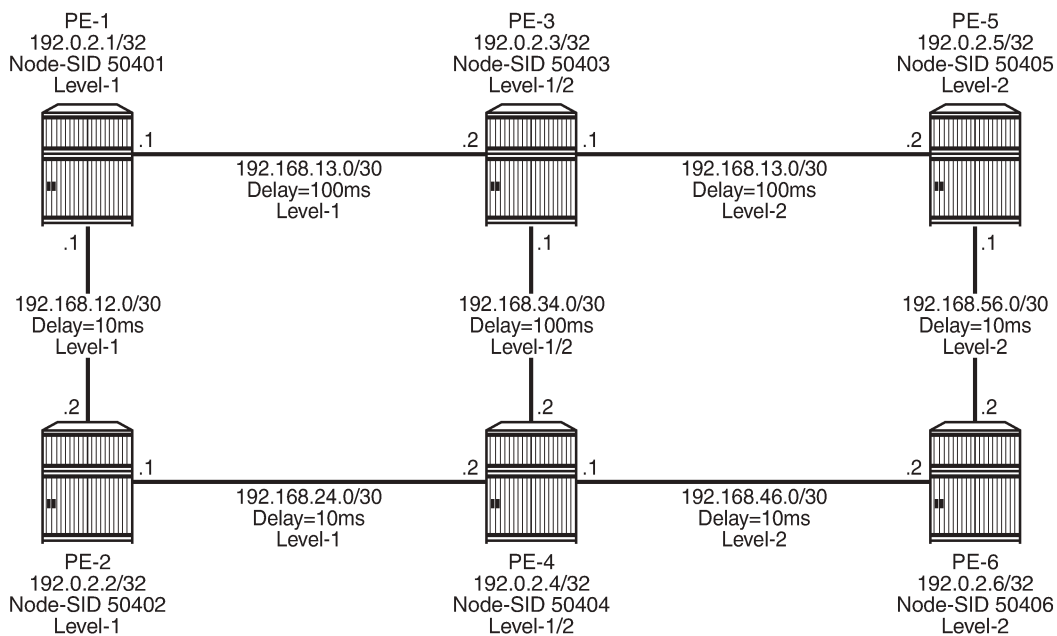
```
-----
Flags: B = BGP or MPLS backup hop available
        L = Loop-Free Alternate (LFA) hop available
        E = Inactive best-external BGP route
        k = RIB-API or Forwarding Policy backup hop
=====
```

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 127: 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 127: 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: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:PE-1# show router tunnel-table 192.0.2.5/32 protocol isis

=====
IPv4 Tunnel Table (Router: Base)
=====
```


Destination Color	Owner	Encap	TunnelId	Pref	Nexthop	Metric
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](#)

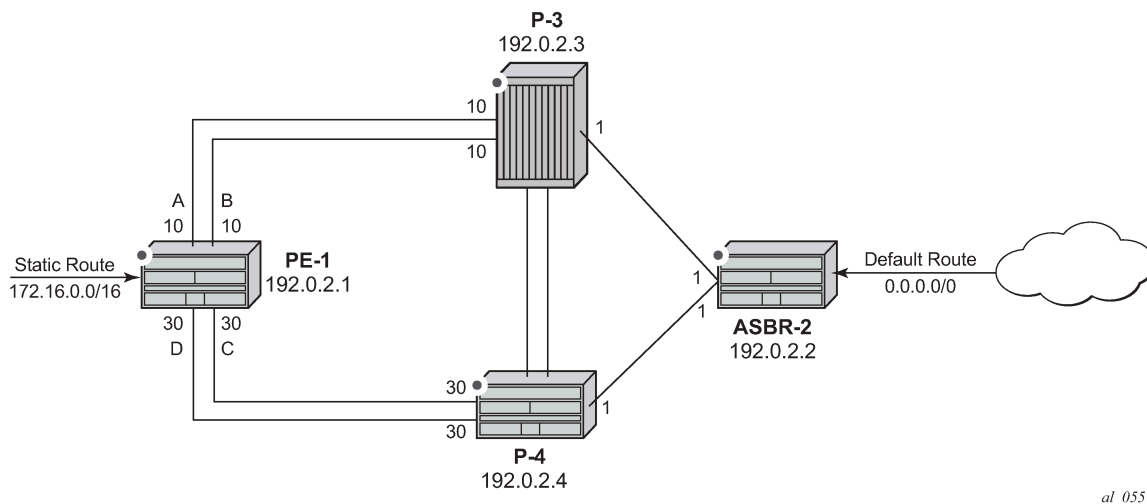
Applicability

The chapter was initially written for SR OS Release 11.0.R6. However, the CLI in the current edition is based on 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.

Figure 128: Link bundle schematic



Consider the network in [Figure 128: 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). 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 128: 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 PE-1 and P-3, and PE-1 and P-4, is load balanced across the two links.

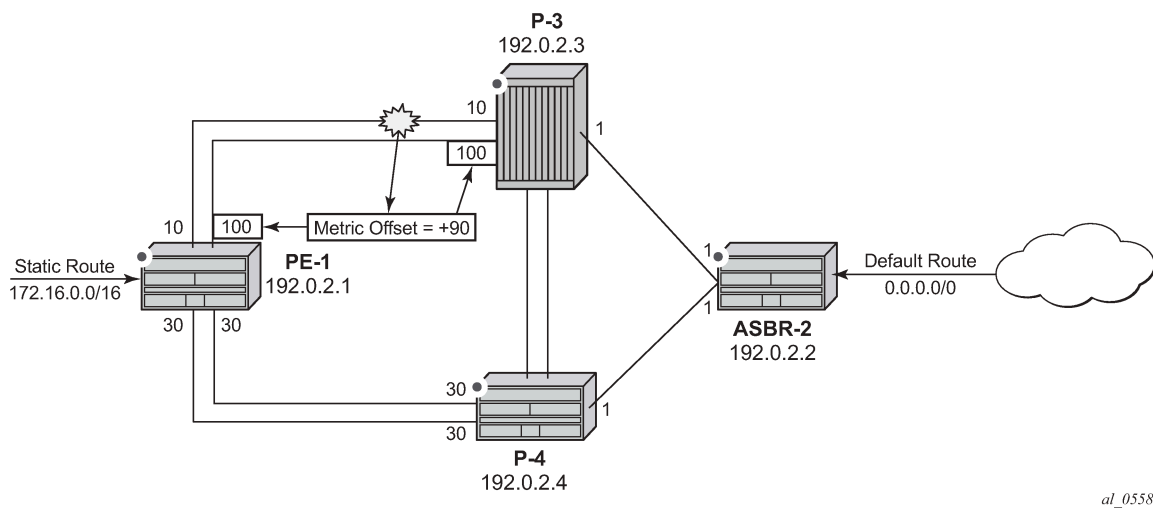
A default route is injected into the ASBR-2 router and re-distributed via a policy statement into IS-IS, so that traffic flowing from PE-1 to the ASBR is resolved by this route. Traffic flows between PE-1 and ASBR-2, using the path with the lowest IS-IS metric, via P-3 with a metric of 11. The second 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.

Figure 129: Effect of single link failure on bundle group



Using [Figure 129: 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. To illustrate the change in metrics, a default static route is configured on ASBR-2 and re-distributed 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).

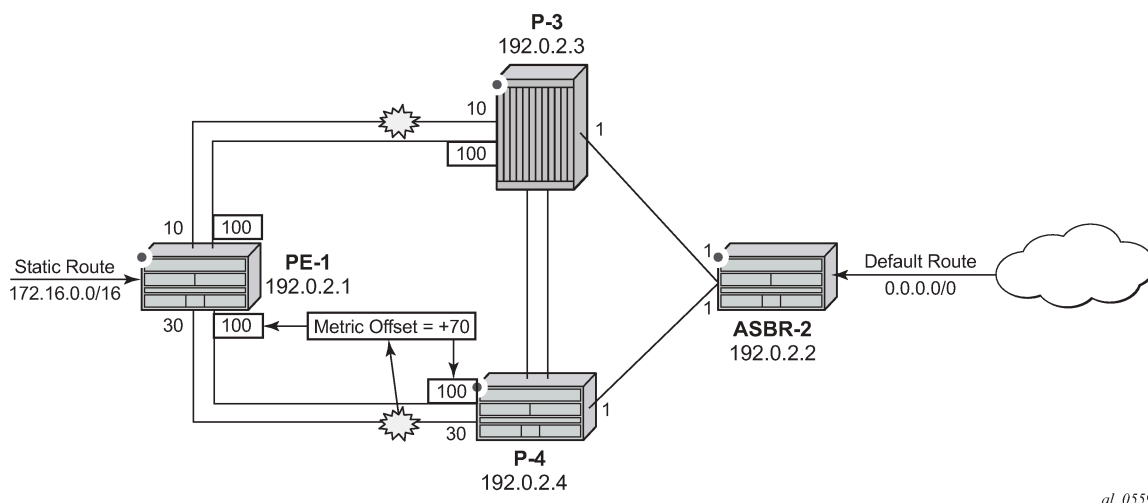
The path between PE-1 and ASBR-2 via P-4 now has the lowest IS-IS metric, and any affected routers within the IS-IS area will try and re-route the traffic based on the new metric.

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 130: Double link failure](#).

Figure 130: Double link failure



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 128: Link bundle schematic](#) and link status is Up (U) or Down (D).

Table 5: Status of the links A, B, C, and D

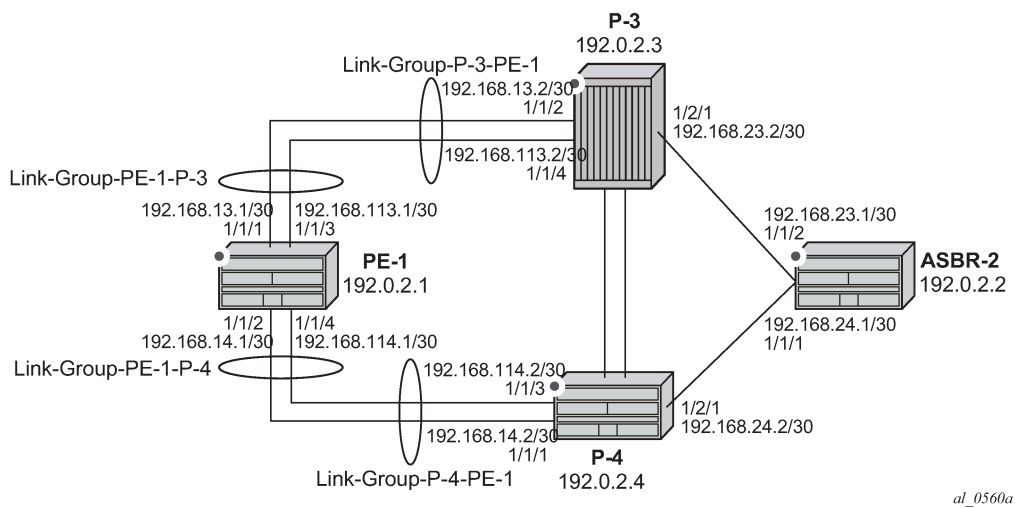
ABCD Status	A (metric,status)	B (metric,status)	C (metric,status)	D (metric,status)
UUUU	10 Transmit	10 Transmit	30 Idle	30 Idle
UDUU	100 Idle	Down	30 Transmit	30 Transmit

ABCD Status	A (metric,status)	B (metric,status)	C (metric,status)	D (metric,status)
UDUD	100 Transmit	Down	100 Transmit	Down
UUUD	10 Transmit	10 Transmit	100 Idle	Down

Configuration

The example topology is shown in [Figure 131: Example topology](#).

Figure 131: Example topology



al_0560a

The PE-1 router configuration commands are as follows.

```
# on PE-1:
configure
router
  interface "int-PE-1-P-3-1"
    address 192.168.13.1/30
    port 1/1/1
  exit
  interface "int-PE-1-P-3-2"
    address 192.168.113.1/30
    port 1/1/3
  exit
  interface "int-PE-1-P-4-1"
    address 192.168.14.1/30
    port 1/1/2
  exit
  interface "int-PE-1-P-4-2"
    address 192.168.114.1/30
    port 1/1/4
  exit
  interface "system"
    address 192.0.2.1/32
  exit
```

ecmp 2

The IP router configuration for the remaining routers can be derived from [Figure 131: 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
  isis
    level-capability level-1
    area-id 49.0001
    advertise-passive-only
    level 1
      wide-metrics-only
    exit
    interface "system"
      passive
    exit
    interface "int-PE-1-P-3-1"
      interface-type point-to-point
      level 1
      metric 10
    exit
    exit
    interface "int-PE-1-P-3-2"
      interface-type point-to-point
      level 1
      metric 10
    exit
    exit
    interface "int-PE-1-P-4-1"
      interface-type point-to-point
      level 1
      metric 30
    exit
    exit
    interface "int-PE-1-P-4-2"
      interface-type point-to-point
      level 1
      metric 30
    exit
  exit
no shutdown
```

The IS-IS configuration for the remaining routers can be derived from [Figure 131: 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
  static-route-entry 0.0.0.0/0
    black-hole
    no shutdown
  exit
```

```
# on PE-1, ASBR-2:
configure
```

```

router
  policy-options
  begin
    policy-statement "STATIC-ISIS"
      entry 10
        from
          protocol static
        exit
        to
          level 1
        exit
        action accept
          metric set "@igp@"
        exit
      exit
    exit
  commit
exit
isis 0
  export "STATIC-ISIS"
exit

```

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
    isis 0
      link-group "Link-Group-PE-1-P-3"
        level 1
          ipv4-unicast-metric-offset 90
          member "int-PE-1-P-3-1"
          member "int-PE-1-P-3-2"
          revert-members 2
          oper-members 2
        exit
      exit
    exit
  exit

```

Similarly, the IS-IS link group for PE-1 for the interfaces toward P-4 is:

```

# on PE-1:
configure
  router
    isis 0
      link-group "Link-Group-PE-1-P-4"
        level 1
          ipv4-unicast-metric-offset 70
          member "int-PE-1-P-4-1"

```

```

        member "int-PE-1-P-4-2"
        revert-members 2
        oper-members 2
    exit
exit
exit

```

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 of the P-routers is as follows.

```

# on P-3:
configure
router
  isis 0
    level-capability level-1
    area-id 49.0001
    advertise-passive-only
    level 1
      wide-metrics-only
    exit
    interface "system"
      passive
    exit
    interface "int-P-3-PE-1-1"
      interface-type point-to-point
      level 1
      metric 10
    exit
  exit
  interface "int-P-3-PE-1-2"
    interface-type point-to-point
    level 1
    metric 10
  exit
  interface "int-P-3-ASBR-2"
    interface-type point-to-point
    level 1
    metric 1
  exit
  link-group "Link-Group-P-3-PE-1"
    level 1

```



```

        ipv4-unicast-metric-offset 90
        member "int-P-3-PE-1-1"
        member "int-P-3-PE-1-2"
        revert-members 2
        oper-members 2
    exit
exit
no shutdown
exit

```

```

# on P-4:
configure
router
isis
    level-capability level-1
    area-id 49.0001
    advertise-passive-only
    level 1
        wide-metrics-only
    exit
    interface "system"
        passive
    exit
    interface "int-P-4-PE-1-1"
        interface-type point-to-point
        level 1
            metric 30
        exit
    exit
    interface "int-P-4-PE-1-2"
        interface-type point-to-point
        level 1
            metric 30
        exit
    exit
    interface "int-P-4-ASBR-2"
        interface-type point-to-point
        level 1
            metric 1
        exit
    exit
    link-group "Link-Group-P-4-PE-1"
        level 1
            ipv4-unicast-metric-offset 70
            member "int-P-4-PE-1-1"
            member "int-P-4-PE-1-2"
            revert-members 2
            oper-members 2
        exit
    exit
no shutdown
exit

```

An overview of all link groups can be shown using the following commands, in this case on node PE-1.

The link group status is as follows:

```

*A: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		
Link-Group-PE-1-P-3	2	2	2	L1	normal
Link-Group-PE-1-P-4	2	2	2	L1	normal

The output for the individual link group members is as follows:

For "Link-Group-PE-1-P-3" at PE-1:

```
*A:PE-1# show router isis link-group-member-status level 1 "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:PE-1# show router isis link-group-member-status level 1 "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: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:P-3# show router isis link-group-member-status level 1 "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
=====
```

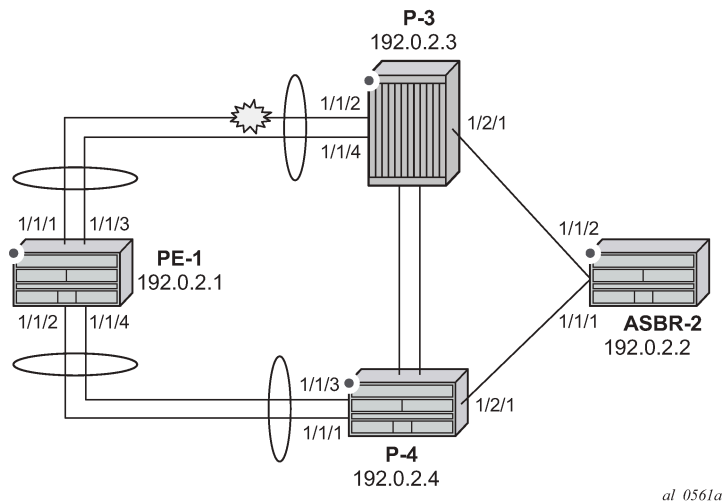
Routing table PE-1

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: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 132: 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 per [Figure 132: Link failure](#).

```
# on P-3:
configure
```

```
port 1/1/2
shutdown
```

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: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: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: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
=====
```

By examining the IS-IS database on PE-1, it can be seen 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: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   : 905
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

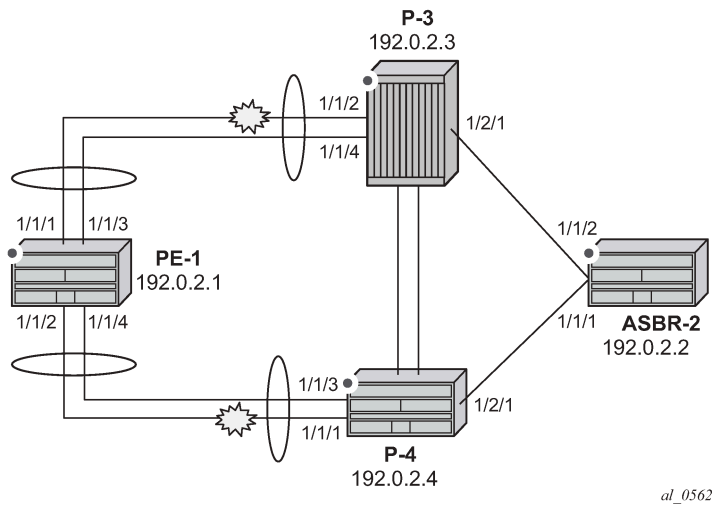
TLVs :
Area Addresses:
  Area Address : (3) 49.0001
Supp Protocols:
  Protocols    : IPv4
IS-Hostname   : PE-1
Router ID     :
  Router ID    : 192.0.2.1
I/F Addresses :
  I/F Address  : 192.0.2.1
  I/F Address  : 192.168.13.1
  I/F Address  : 192.168.14.1
  I/F Address  : 192.168.113.1
  I/F Address  : 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 133: 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
shutdown
```

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: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
```

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: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    : 0xb                Checksum   : 0xd372    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** 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 **disable-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)

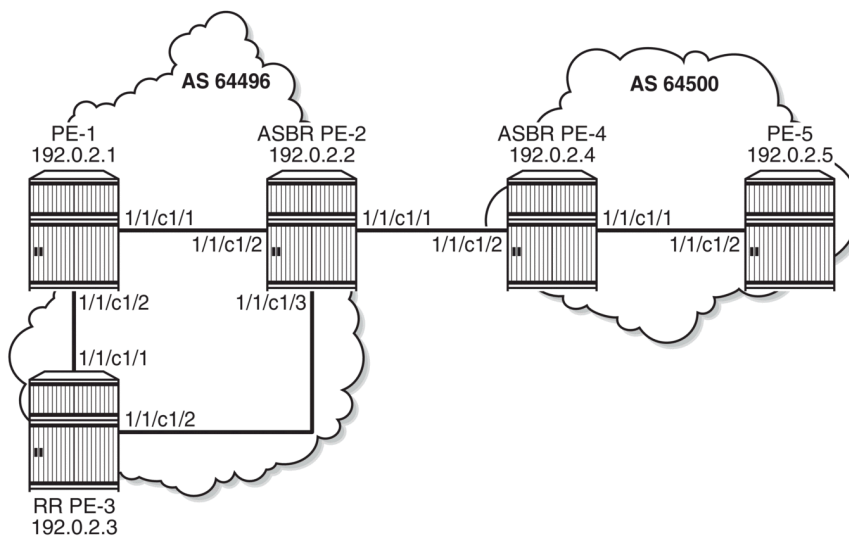
If the BGP NH is an IPv4 address, the TTM selects the tunnel table entry that matches the address prefix with the lowest preference and allowed by the applicable resolution filter. If the preference is the same, the tunnel table entry with the best metric is chosen, and so on.

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 134: Example topology](#) shows the example topology with three routers in AS 64496 and two routers in AS 64500.

Figure 134: 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
router
  policy-options
  begin
  prefix-list "PE-sys"
    prefix 192.0.2.0/28 prefix-length-range 32-32
  exit
  policy-statement "export-bgp"
  entry 10
    from
      protocol direct
      prefix-list "PE-sys"
    exit
    to
      protocol bgp-label
    exit
  action accept
  exit
  exit
exit
commit
exit all
```

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
  bgp
  split-horizon
  group "iBGP"
  peer-as 64496
  advertise-inactive
  cluster 192.0.2.3
  neighbor 192.0.2.1
    family vpn-ipv4 vpn-ipv6 label-ipv4
  exit
  neighbor 192.0.2.2
    family label-ipv4
  exit
  exit
exit
exit all
```

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**) where these routes are used. The initial BGP configuration on PE-2 is as follows:

```
# on PE-2:
configure
router
  bgp
    split-horizon
    group "iBGP"
      peer-as 64496
      family label-ipv4
      advertise-inactive
      neighbor 192.0.2.3
    exit
  exit
  group "eBGP4_local"
    family label-ipv4
    advertise-inactive
    neighbor 192.168.24.2
      peer-as 64500
    exit
  exit
exit all
```

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:PE-2# configure router bgp next-hop-resolution labeled-routes
*A:PE-2>config>router>bgp>next-hop-res>lbl-routes# info detail | match "allow-static"
no allow-static
*A:PE-2>config>router>bgp>next-hop-res>lbl-routes# info detail | match "family label-ipv4"
post-lines 16
    family label-ipv4
      resolution-filter
        ldp
        ---snip---
        no bgp
        ---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: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 00h03m04s 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** is enabled), 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
  interface "loopback"
  address 10.0.0.2/32
  loopback
  exit
  static-route-entry 10.0.0.4/32
  next-hop 192.168.24.2
  no shutdown
  exit
  exit
  exit all
```

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
  bgp
  next-hop-resolution
  labeled-routes
  allow-static
  exit
  exit
  group "eBGP4_static"
  neighbor 10.0.0.4
  peer-as 64500
  family label-ipv4
  advertise-inactive
  local-address 10.0.0.2
  exit
  exit
  exit all
```

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
  static-route-entry 192.0.2.4/32
  next-hop 192.168.24.2
  preference 25
  no shutdown
  exit
```

```

exit
ldp
  interface-parameters
    interface "int-PE-2-PE-4" dual-stack
      ipv4
        no shutdown
      exit
      no shutdown
    exit
  exit
exit
bgp
  group "eBGP4_tunnel"
    neighbor 192.0.2.4
      peer-as 64500
      family label-ipv4
      advertise-inactive
    exit
  exit
exit all

```

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:PE-2# show router fib 1 192.0.2.5/32
=====
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: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                                   524284
*i    192.0.2.5/32                           None       None
      10.0.0.4                               None       1
      64500                                   524284
*i    192.0.2.5/32                           None       None
      192.0.2.4                              None       1
      64500                                   524284
-----
Routes : 3
=====

```

Table 6: 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 6: 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: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 Resolved Next Hop Admin-tag-policy	FibProg Colored FlexAlgo	Metric Ref. Count Last Mod.
10.0.0.4	5	STATIC
10.0.0.4/32	N	1
192.168.24.2	N	0
--	--	00h18m40s
192.0.2.3	15	ISIS
192.0.2.3/32	N	10
192.168.23.2	N	0
--	--	00h25m47s
192.0.2.4	25	STATIC
192.0.2.4/32	N	1
192.168.24.2	N	0
--	--	00h18m07s
192.168.24.2	0	LOCAL
192.168.24.0/30	N	0
192.168.24.2	N	0
--	--	00h25m47s

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
  bgp
    group "eBGP4_local"
      shutdown
    exit all
```

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:PE-2# show router fib 1 192.0.2.5/32

=====
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: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                    524284
*i    192.0.2.5/32              None        None
      192.0.2.4                None        1
      64500                    524284
-----
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: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
--                       --    00h30m21s
192.0.2.3               15    ISIS
192.0.2.3/32            N     10
192.168.23.2            N     0
--                       --    00h37m28s
192.0.2.4               25    STATIC
192.0.2.4/32            N     1
192.168.24.2            N     0
--                       --    00h29m48s
-----
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
  static-route-entry 10.0.0.4/32
    next-hop 192.168.24.2
  shutdown
exit all
```

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:PE-2# show router fib 1 192.0.2.5/32

=====
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: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                               Path-Id    Label
-----
u*>i  192.0.2.5/32                            None       None
      192.0.2.4                            None       1
      64500                                  None       524284
i     192.0.2.5/32                            None       None
      10.0.0.4                              None       0
      64500                                  None       524284
-----
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: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
--          --         --
--          --         00h22m54s
192.0.2.3      15        ISIS
192.0.2.3/32   N         10
192.168.23.2   N         0
--          --         01h05m18s
192.0.2.4      25        STATIC
192.0.2.4/32   N         1
192.168.24.2  N         0
--          --         00h57m38s
-----
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
    static-route-entry 10.0.0.4/32
        next-hop 192.168.24.2
        no shutdown
    exit all
```

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
    bgp
        next-hop-resolution
            labeled-routes
                no allow-static
    exit all
```

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: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 00h01m22s 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:PE-2# show router fib 1 192.0.2.5/32

=====
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: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                   524284
i     192.0.2.5/32          None       None
      10.0.0.4                None       0
      64500                   524284
```

```
-----
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: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
  --                          --     00h05m07s
192.0.2.3                   15    ISIS
  192.0.2.3/32               N      10
  192.168.23.2               N      0
  --                          --     01h26m05s
192.0.2.4                   25   STATIC
  192.0.2.4/32                N      1
  192.168.24.2              N      0
  --                          --     01h18m25s
-----
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
  bgp
    group "eBGP4_local"
      no shutdown
    exit all
```

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 enabling **rr-use-route-table**. The following error is raised when attempting to configure **rr-use-route-table** without **disable-route-table-install**:

```
*A:PE-3# configure router bgp next-hop-resolution labeled-routes rr-use-route-table
INFO: BGP #1001 Configuration failed because of inconsistent values - BGP [VR 1] route-table-
for-label-routes cannot be set unless disable-route-table-install is set!
```

The command **disable-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
  bgp
    disable-route-table-install
    split-horizon
    next-hop-resolution
      labeled-routes
        rr-use-route-table
    exit
  exit
  group "iBGP"
    peer-as 64496
    family vpn-ipv4 vpn-ipv6 label-ipv4
    advertise-inactive
    cluster 192.0.2.3
    neighbor 192.0.2.1
    exit
    neighbor 192.0.2.2
    exit
  exit
exit all
```

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: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                LocalPref  MED
      Nexthop (Router)      Path-Id    IGP Cost
      As-Path                Path-Id    Label
-----
*>i  192.0.2.5/32            100        None
      192.0.2.2              None        10
      64500                   524281
-----
Routes : 1
=====
```

The following labeled BGP route has NH 192.0.2.2, which is resolved to an IS-IS route:

```
*A: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
  --                                       --     01h29m52s
-----
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: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                                     524281
-----
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: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:PE-1# show router fp-tunnel-table 1

```

```

=====
IPv4 Tunnel Table Display

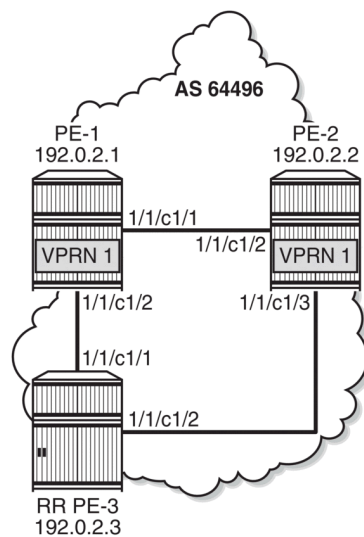
Legend:
label stack is ordered from bottom-most to top-most
B - FRR Backup
=====
Destination                                Protocol  Tunnel-ID
Lbl/SID                                     Intf/Tunnel
NextHop
Lbl/SID (backup)
NextHop (backup)
-----
192.0.2.2/32                                LDP      -
524287
192.168.12.2                                BGP      1/1/c1/1:100
192.0.2.5/32                               BGP      -
524281
192.0.2.2                                  LDP
-----
Total Entries : 2
=====

```

NH resolution for iBGP VPN-IPv4/v6 routes

Figure 135: VPRN 1 in AS 64496 shows that VPRN 1 is configured on PE-1 and PE-2 in AS 64496.

Figure 135: 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
  bgp

```



```

split-horizon
group "iBGP"
peer-as 64496
export "export-bgp"
neighbor 192.0.2.3
    family vpn-ipv4 vpn-ipv6
exit
exit
exit all
    
```

On PE-1, VPRN 1 is configured as follows. The configuration on PE-2 is similar.

```

# on PE-1:
configure
service
vprn 1 name "VPRN 1" customer 1 create
bgp-ipvpn
mpls
route-distinguisher 64496:1
vrf-target target:64496:1
auto-bind-tunnel
resolution filter
resolution-filter
ldp
exit
exit
no shutdown
exit
interface "loopback1" create
loopback
address 1.1.1.1/32
ipv6
address 2001:db8::1:1:1:1/128
exit
exit
no shutdown
exit
exit all
    
```

Even though only LDP is explicitly configured in the auto-bind tunnel resolution filter, the resolution filter allows LDP and BGP tunnels:

```

*A:PE-1# configure service vprn 1
*A:PE-1>config>service>vprn$ info detail | match "auto-bind-tunnel" post-lines 19
    auto-bind-tunnel
        resolution-filter
            ---snip---
            ldp
            ---snip---
            bgp
            ---snip---
    
```

VPRN 1 is only configured on nodes in AS 64496, so only LDP transport tunnels are used. The following tunnel table on PE-2 shows that an LDP tunnel toward PE-1 is available:

```

*A:PE-2# 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	192.168.12.1	10

---snip---

PE-2 receives the following BGP VPN-IPv4 route with route distinguisher (RD) 64496:1 used in VPRN 1:

```
*A: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                               524283
-----
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: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:524283 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: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                               LocalPref MED
  Nexthop (Router)                         Path-Id   IGP Cost
  As-Path                                   As-Path   Label
-----
u*>i 64496:1:2001:db8::1:1:1/128           100      None
      ::ffff:192.0.2.1                     None     10
      No As-Path                           As-Path   524283
-----
Routes : 1
=====

```

The following IPv6 FIB for VPRN 1 shows that a LDP tunnel is used to reach NH 192.0.2.1:

```

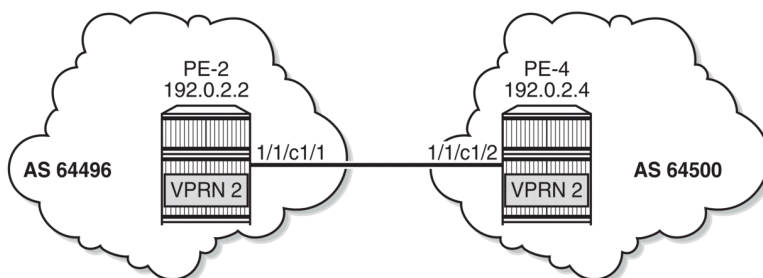
*A:PE-2# show router 1 fib 1 ipv6
=====
FIB Display
=====
Prefix [Flags]                               Protocol
  NextHop
-----
2001:db8::1:1:1/128                          BGP_VPN
  192.0.2.1 (VPRN Label:524283 Transport:LDP)
2001:db8::2:2:2:1/128                        LOCAL
  2001:db8::2:2:2:1 (loopback1)
-----
Total Entries : 2
=====

```

NH resolution for inter-AS VPRN model B

Figure 136: VPRN 2 in AS 64496 and in AS 64500 shows that VPRN 2 is configured in AS 64496 and in AS 64500.

Figure 136: 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 2 name "VPRN 2" customer 1 create
    bgp-ipvpn
      mpls
        route-distinguisher 2:2
        vrf-target target:2:2
        auto-bind-tunnel
          resolution filter
          resolution-filter
        ldp
      exit
    exit
    no shutdown
  exit
  interface "loopback2" create
  loopback
    address 2.2.2.2/32
  ipv6
    address 2001:db8::2:2:2:2/128
  exit
  no shutdown
exit
exit all
```

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
    bgp
      enable-inter-as-vpn
      split-horizon
      rapid-update vpn-ipv4 vpn-ipv6 label-ipv4
      next-hop-resolution
        labeled-routes
          allow-static
        exit
      exit
      group "eBGP4_local"
        neighbor 192.168.24.2
        peer-as 64500
        family vpn-ipv4 vpn-ipv6
      exit
      group "eBGP4_static"
        neighbor 10.0.0.4
        peer-as 64500
        family vpn-ipv4 vpn-ipv6
        local-address 10.0.0.2
      exit
      group "eBGP4_tunnel"
        neighbor 192.0.2.4
        peer-as 64500
```

```

        family vpn-ipv4 vpn-ipv6
        exit
    exit
    no shutdown
exit all

```

VPN IP NH resolved to local route

PE-2 has three BGP VPN-IPv4 routes for prefix 4.4.4.2/32. The used route is NH 192.168.24.2, which is a local route.

```

*A: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
      192.168.24.2                          None       0
      64500                                   524285
*i    2:2:4.4.4.2/32                          None       None
      10.0.0.4                               None       1
      64500                                   524285
*i    2:2:4.4.4.2/32                          None       None
      192.0.2.4                              None       1
      64500                                   524285
-----
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: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: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                                None       524285
*i    2:2:2001:db8::4:4:4:2/128              None       None
      ::ffff:10.0.0.4                      None       1
      64500                                None       524285
*i    2:2:2001:db8::4:4:4:2/128              None       None
      ::ffff:192.0.2.4                    None       1
      64500                                None       524285
-----
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
  bgp
    group "eBGP4_local"
      shutdown
  exit all
```

On PE-2, the static route with the best preference is toward 10.0.0.4:

```
*A: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      524285
*>i 2:2:4.4.4.2/32                          None      None
    192.0.2.4                                  None      1
    64500                                       None      524285
-----
Routes : 2
=====

```

On PE-2, NH 10.0.0.4 is resolved to 192.168.24.2:

```

*A: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
  --                                     --    00h27m46s
192.0.2.4                                25     STATIC
  192.0.2.4/32                            N      1
  192.168.24.2                           N      0
  --                                     --    01h41m04s
-----
Next Hops : 2
=====

```

This resolved NH 192.168.24.2 is the NH for prefix 4.4.4.2/32 in the FIB:

```

*A: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: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                                     524285
*>i   2:2:2001:db8::4:4:4:2/128                None       None
      ::ffff:192.0.2.4                          None       1
      64500                                     524285
-----
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
  bgp
    next-hop-resolution
    labeled-routes
      no allow-static
  exit all

```

On PE-2, the used VPN-IPv4 route toward 4.4.4.2/32 has NH 192.0.2.4:

```

*A: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 Nexthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
u*>i	2:2:4.4.4.2/32 192.0.2.4 64500	None None	None 1 524285
*i	2:2:4.4.4.2/32 10.0.0.4 64500	None None	None 1 524285

Routes : 2

The tunnel table on PE-2 shows that an LDP tunnel is available toward 192.0.2.4/32:

```
*A: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: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:524285 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:PE-2# show router 2 fib 1 ipv6

=====
FIB Display
=====
Prefix [Flags]          Protocol
  NextHop
-----
2001:db8::2:2:2:2/128  LOCAL
=====
```

```

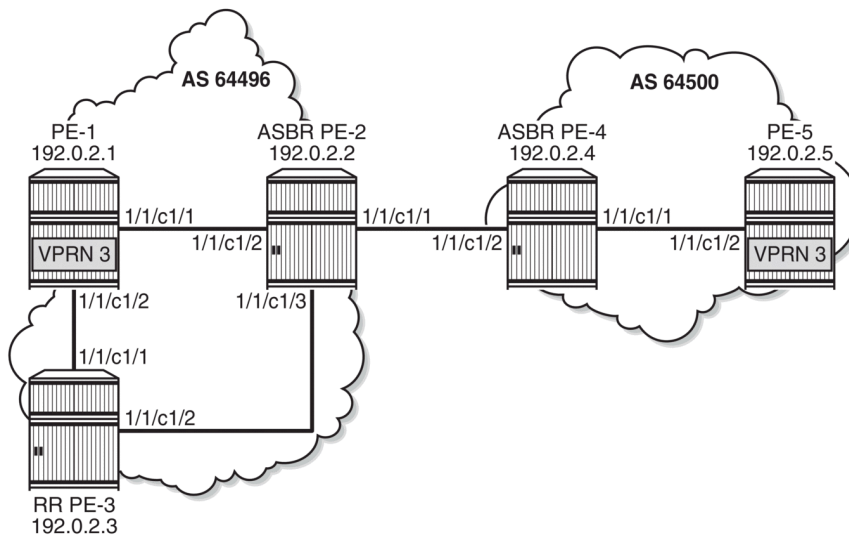
2001:db8::2:2:2:2 (loopback2)
2001:db8::4:4:4:2/128
192.0.2.4 (VPRN Label:524285 Transport:LDP)
-----
Total Entries : 2
=====

```

NH resolution for inter-AS VPRN model C

Figure 137: 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 137: 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
  bgp
    split-horizon
    rapid-update vpn-ipv4 vpn-ipv6 label-ipv4
    group "iBGP"
      peer-as 64496
      export "export-bgp"
      neighbor 192.0.2.3
        family vpn-ipv4 vpn-ipv6 label-ipv4
    exit
  exit
  group "eBGP_multihop"
    peer-as 64500
    neighbor 192.0.2.5
      family vpn-ipv4 vpn-ipv6

```

```

        local-address 192.0.2.1
        multihop 10
    exit
exit
exit
exit all

```

The BGP configuration on RR PE-3 is as follows. The RR is configured with **disable-route-table-install**, 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
  bgp
    disable-route-table-install
    split-horizon
    next-hop-resolution
      labeled-routes
        rr-use-route-table
    exit
  exit
  group "iBGP"
    peer-as 64496
    advertise-inactive
    cluster 192.0.2.3
    neighbor 192.0.2.1
      family vpn-ipv4 vpn-ipv6 label-ipv4
    exit
    neighbor 192.0.2.2
      family label-ipv4
    exit
  exit
exit
exit all

```

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
  bgp
    split-horizon
    rapid-update vpn-ipv4 vpn-ipv6 label-ipv4
    group "iBGP"
      peer-as 64496
      family label-ipv4
      advertise-inactive
      neighbor 192.0.2.3
    exit
  exit
  group "eBGP4_local"
    family label-ipv4
    advertise-inactive
    neighbor 192.168.24.2
      peer-as 64500
    exit
  exit
exit
exit all

```

On PE-1, VPRN 3 is configured as follows. The configuration is similar on PE-5.

```
# on PE-1:
configure
service
  vprn 3 name "VPRN 3" customer 1 create
  bgp-ipvpn
  mpls
    route-distinguisher 3:3
    vrf-target target:3:3
    auto-bind-tunnel
    resolution filter
    resolution-filter
    ldp
  exit
  exit
  no shutdown
exit
interface "loopback3" create
  loopback
  address 1.1.1.3/32
  ipv6
    address 2001:db8::1:1:1:3/128
  exit
exit
no shutdown
exit
exit all
```

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:PE-1# configure service vprn 3
*A:PE-1>config>service>vprn# info detail | match "auto-bind-tunnel" post-lines 19
  auto-bind-tunnel
  resolution-filter
  ---snip---
  ldp
  ---snip---
  bgp
  ---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: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                  None      524285
-----
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: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  262147  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: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:524285 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: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                              524285
*>i 3:3:5.5.5.3/32                          100      None
    192.0.2.5                               None     0
    64500                                    524285
-----
Routes : 2
=====

```

On RR PE-3, NH 192.0.2.1 is resolved using the RTM:

```

*A: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
--             --      02h04m45s
192.0.2.2      15     ISIS
192.0.2.2/32   N       10
192.168.23.1   N       0
--             --      02h04m45s
-----
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 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 7: Policy chaining versus policy logical expressions](#) shows a comparison between examples of policy chaining and policy logical expressions.

Table 7: Policy chaining versus policy logical expressions

Policy chaining example	Policy logical expressions example
<code>configure router bgp import "A" "B" "C"</code>	<code>configure router bgp import "[A]OR[B]"</code>

Policy chaining example	Policy logical expressions example
<p>For each route, policy A is evaluated first.</p> <ul style="list-style-type: none"> If policy A matches the route with action next-policy, then apply any route modifications and continue to evaluate policy B, and so on. When the route is matched in a policy with action accept or drop/reject, the evaluation is completed. 	<p>Several logical operators can be used. This example shows an OR relationship 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>



Note:

In SR OS Release 14.0.R4, and later, the **action drop** replaces the **action reject**. The difference is that with **action drop**, it is possible to modify attributes in the same way as for **action accept**. This behavior is useful when a NOT operation makes a false expression true and the attributes are required. In a similar way, it is possible that an OR operation is true, even though the first policies that were evaluated were false. Routes are accepted when the final result is true and all policies that were evaluated will modify the route attributes. Examples of this behavior are included in the [Configuration](#) section.

To configure policy chaining that may or may not include a policy logical expression, the syntax is:

```
# on PE-1:
configure router "Base" bgp export ?

*A:PE-1# configure router "Base" bgp export
- export <plcy-or-long-expr> [<plcy-or-expr> [<plcy-or-expr>...(up to 14 max)]]
- no export

<plcy-or-long-expr> : <policy-name> | <long-expr>
                    <policy-name> - [64 chars max]
                    <long-expr>   - [255 chars max]
<plcy-or-expr>    : <policy-name> | <expr>
                    <policy-name> - [64 chars max]
                    <expr>        - [64 chars max]
```

When the **import/export** 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** may or may not be enclosed in double quotes. When not enclosed in double quotes, the operand must not be separated from the policies. 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. When separated, the logical expression must be enclosed in double quotes.
- 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. When a policy is enclosed in square brackets, the square brackets are part of the policy.

```
# on PE-1:
configure router "Base" bgp import A
(leads to: "A")
```

```
configure router "Base" bgp import "A"
(leads to: "A")
```

```
configure router "Base" bgp import [A]
(leads to: "[A]")
```

```
configure router "Base" bgp import "[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 import " A "
WARNING: CLI Policy " A " does not exist.
```

A policy that is enclosed in double quotes must not be enclosed in square brackets.

```
# on PE-1:
configure router "Base" bgp import ["A"]
*A:PE-1>config>router>bgp#   import ["A"]
                               ^
Error: Invalid syntax.
```

A logical expression is accepted only when each policy is enclosed in square brackets and not in double quotes, while the logical expression itself may or may not be enclosed in double quotes.

```
# on PE-1:
configure router "Base" bgp import [A]AND[B]
(leads to: "[A]AND[B]")
```

```
configure router "Base" bgp import "[A]AND[B]"
(leads to: "[A]AND[B]")
```

In a logical expression, spaces before and after the operand are allowed only when the logical expression is enclosed in double quotes.

```
# on PE-1:
configure router "Base" bgp import "[A] AND [B]"
(leads to: "[A] AND [B]")
```

The following message is raised when, in a logical expression that is not enclosed in double quotes, there are spaces before and after the operand.

```
# on PE-1:
configure router "Base" bgp import [A] AND [B]
```

```
WARNING: CLI Policy "AND" does not exist.
INFO: BGP #1001 Configuration failed because of inconsistent values - BGP [VR 1] Policy stmts
expression format error
```

In a logical expression, spaces before and after the square brackets that enclose a policy are allowed only when the logical expression is enclosed in double quotes.

```
# on PE-1:
configure router "Base" bgp import " [A] AND [B] "
(leads to: " [A] AND [B] ")
```

The following message is raised when, in a logical expression that is not enclosed in double quotes, there are spaces before and after the square brackets that enclose a policy.

```
# on PE-1:
configure router "Base" bgp import [A] AND [B]
WARNING: CLI Policy "AND" does not exist.
INFO: BGP #1001 Configuration failed because of inconsistent values - BGP [VR 1] Policy stmts
expression format error
```

In a logical expression, spaces inside the square brackets that enclose a policy are allowed only when the logical expression is enclosed in double quotes.

```
# on PE-1:
configure router "Base" bgp import "[ A ]AND[ B ]"
(leads to: "[ A ]AND[ B ]")
```

The following message is raised when there are spaces inside the square brackets that enclose a policy in a logical expression that is not enclosed in double quotes.

```
configure router "Base" bgp import [ A ]AND[ B ]
WARNING: CLI Policy "[" does not exist.
WARNING: CLI Policy "]" does not exist.
INFO: BGP #1001 Configuration failed because of inconsistent values - BGP [VR 1] Policy stmts
expression format error
```

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 import ["A"]AND["B"]
*A:PE-1>config>router>bgp# import ["A"]AND["B"]
      ^
Error: Invalid syntax.
```

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 import "A"AND"B"
*A:PE-1>config>router>bgp# import "A"AND"B"
      ^
Error: Invalid syntax.
```

The following message is raised when there are double quotes inside the square brackets that enclose a policy in a logical expression that is enclosed in double quotes.

```
# on PE-1:
configure router "Base" bgp import "["A"]AND["B"]"
*A:PE-1>config>router>bgp#      import "["A"]AND["B"]"
                                     ^
Error: Invalid syntax.
```

The following message is raised when a policy is enclosed in double quotes instead of square brackets in a logical expression that is enclosed in double quotes.

```
# on PE-1:
configure router "Base" bgp import ""A"AND"B""
*A:PE-1>config>router>bgp#      import ""A"AND"B""
                                     ^
Error: Invalid syntax.
```

The operand in a logical expression must be in uppercase.

```
# on PE-1:
configure router "Base" bgp import "[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 import "[A]and[B]"
INFO: BGP #1001 Configuration failed because of inconsistent values - BGP [VR 1] Policy stmts
expression format error
```

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 import "(((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 import "((((P1]AND[P2])OR[P3])AND[P4])OR[P5])AND[P6]"
INFO: BGP #1001 Configuration failed because of inconsistent values - BGP [VR 1] Policy stmts
expression format error
```

The following message is raised when the parentheses are not balanced.

```
# on PE-1:
configure router "Base" bgp import "(((P1]AND[P2])OR[P3])AND[P4]"
INFO: BGP #1001 Configuration failed because of inconsistent values - BGP [VR 1] Policy stmts
expression format error
```

A logical expression accepts a maximum of 16 policies.

```
# on PE-1:
configure router "Base" bgp import "[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]")
```

The following message is raised when there are too many policies in the logical expression.

```
# on PE-1:
configure router "Base" bgp import "[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]AND[P17]"
INFO: BGP #1001 Configuration failed because of inconsistent values - BGP [VR 1] Policy stmts expression format error
```

When the import/export command has multiple values, the values are separated with spaces and ordered in a **policy chain**. The policy chain accepts between 1 and 15 values. In the policy chain:

- each value is either a policy or a logical expression and must be formatted in the same way as the single value above.
- only 1 logical expression is allowed.
- policies and the logical expression must not be repeated.
- a logical expression with a length that does not exceed 64 characters may be in any location. Otherwise, the logical expression with a length that does not exceed 255 characters must be in the first location.

A policy chain must not be enclosed in square brackets.

```
# on PE-1:
configure router "Base" bgp import A B
(leads to: "A" "B")
```

The following message is raised when the policy chain is enclosed in square brackets.

```
# on PE-1:
configure router "Base" bgp import [A B]
WARNING: CLI Policy "[A" does not exist.
WARNING: CLI Policy "B]" does not exist.
```

A policy chain is accepted with and without double quotes enclosing the policies.

```
# on PE-1:
configure router "Base" bgp import A B
(leads to: "A" "B")
```

```
configure router "Base" bgp import "A" B
(leads to: "A" "B")
```

```
configure router "Base" bgp import A "B"
(leads to: "A" "B")
```

```
configure router "Base" bgp import "A" "B"
(leads to: "A" "B")
```

A policy chain can contain policies and a logical expression.

```
# on PE-1:
configure router "Base" bgp import A B [C]OR[A]
(leads to: "A" "B" "[C]OR[A]")
```

In a policy chain, policies and the logical expression must not be repeated. The following message is raised when the policy chain has duplicate policies or logical expressions.

```
# on PE-1:
configure router "Base" bgp import A A "A"
INFO: BGP #1001 Configuration failed because of inconsistent values - BGP [VR 1] Policy stmts
should be unique and set in order!
```

A policy chain accepts a maximum of 15 policies.

```
# on PE-1:
configure router "Base" bgp import 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 import P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15 P16
*A:PE-1>config>router>bgp# import "P1" "P2" "P3" "P4" "P5" "P6" "P7" "P8" "P9" "P10" "P11"
"P12" "P13" "P14" "P15" P16
^
Error: Invalid parameter.
```

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 import "[A]AND[B]" C "[B]OR[C]"
INFO: BGP #1001 Configuration failed because of inconsistent values - BGP [VR 1] Policy stmts
expression format error
```

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 import A B "[C]AND[A]"
(leads to: "A" "B" "[C]AND[A]")
```

```
configure router "Base" bgp import A "[C]AND[A]" B
(leads to: "A" "[C]AND[A]" "B")
```

```
configure router "Base" bgp import "[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 import
"[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 import 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]"
*A:PE-1>config>router>bgp# import "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]"
Error: String too long
```

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 import
"[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]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]A
ND[A]AND[B]AND[C]AND[A]AND[B]AND[C]AND[A]AND[B]AND[C]"
A B
*A:PE-1>config>router>bgp# import
"[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]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]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]AND[B]AND[C]"
A B
Error: String too long
```

Route policy logical expressions

Logical expressions are evaluated to be true or false. [Table 8: Boolean values for the policy actions](#) shows the mapping of policy actions to Boolean values.

Table 8: Boolean values for the policy actions

Policy action	Boolean value
Accept	True
Next-entry	True
Next-policy	True
Reject	False
Drop	False



Note:

The policy **action drop** replaces **action reject** in Release 14.0.R4, and later. The **action drop** supports route attribute modifications while **action reject** does not. SR OS automatically converts reject actions to drop actions.

[Table 9: Actions for the logical operators](#) shows the evaluation actions for the logical operators NOT, OR, and AND.

Table 9: 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 10: 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 dropped when the entire expression is false.

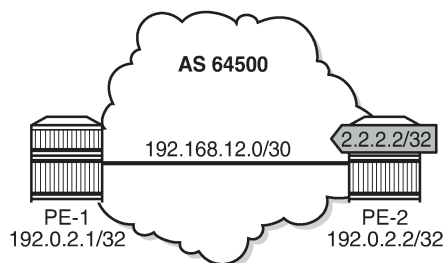
Table 10: Mapping the final result of an expression to a policy action

Final result	Action
True	accept, next-entry, or next-policy (depending on the last entry evaluated)
False	drop/reject

Configuration

[Figure 138: Example topology](#) shows the example topology including the advertised route.

Figure 138: 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
  router "Base"
    policy-options
      begin
        community "A"
          members "1:1"
        exit
        community "B"
          members "2:2"
        exit
        community "C"
          members "3:3"
        exit
        policy-statement "A"
          entry 10
            action accept
              community add "A"
              local-preference 110
            exit
          exit
        exit
        policy-statement "B"
          entry 10
            action accept
              community add "B"
              local-preference 220
            exit
          exit
        exit
        policy-statement "C"
          entry 10
            action next-policy
              community add "C"
            exit
          exit
        exit
      commit
    exit all
```


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 "C" "A" "B"
    exit all
```

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: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
Path Id       : None
From          : 192.0.2.2
Res. Protocol : ISIS                Res. Metric   : 10
Res. NextHop  : 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
      import "[A]AND[B]"
    exit all
```

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:PE-1# show router bgp routes hunt brief | match "Local Pref."
Local Pref.      : 220                               Interface Name : int-PE-1-PE-2
```

```
*A: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:PE-1# show router bgp routes hunt brief | match "Local Pref."
Local Pref.      : 110                               Interface Name : int-PE-1-PE-2
```

```
*A: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
      import "[A]OR[B]"
    exit all
```

```
*A:PE-1# show router bgp routes hunt brief | match "Local Pref."
Local Pref.      : 110                               Interface Name : int-PE-1-PE-2
```

```
*A: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
    import "[B]OR[A]"
  exit all
```

```
*A:PE-1# show router bgp routes hunt brief | match "Local Pref."
Local Pref.      : 220                               Interface Name : int-PE-1-PE-2
```

```
*A: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
    import "NOT[A]AND[B]"
  exit all
```

```
*A:PE-1# show router bgp routes hunt brief
```

```
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
---snip---
-----
RIB In Entries
-----
Network      : 2.2.2.2/32
Nexthop      : 192.0.2.2
Path Id      : None
From         : 192.0.2.2
Res. Protocol : ISIS                Res. Metric   : 10
Res. Nexthop : 192.168.12.2
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       : No Cluster Members
Originator Id : None                 Peer Router Id : 192.0.2.2
Fwd Class     : None                 Priority       : None
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
      import "NOT[A]OR[B]"
    exit all
```

```
*A:PE-1# show router bgp routes hunt brief
```

```
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
---snip---
-----
RIB In Entries
-----
Network      : 2.2.2.2/32
Nexthop      : 192.0.2.2
Path Id      : None
From         : 192.0.2.2
Res. Protocol : ISIS                      Res. Metric   : 10
Res. Nexthop : 192.168.12.2
Local Pref. : 220                      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   : 2:2 1:1
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---
```

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
      import "[B] OR NOT[A]"
    exit all
```

```
*A:PE-1# show router bgp routes hunt brief
```

```
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
---snip---
-----
RIB In Entries
-----
Network      : 2.2.2.2/32
Nexthop      : 192.0.2.2
Path Id      : None
From         : 192.0.2.2
Res. Protocol : ISIS                      Res. Metric   : 10
```

```

Res. Nexthop   : 192.168.12.2
Local Pref.    : 220
Aggregator AS  : None
Atomic Aggr.   : Not Atomic
AIGP Metric    : None
Connector      : None
Community      : 2:2
Cluster        : No Cluster Members
Originator Id  : None
Fwd Class      : None
Flags          : Used Valid Best IGP In-RTM
---snip---
Interface Name : int-PE-1-PE-2
Aggregator     : None
MED            : None
IGP Cost       : 10
Peer Router Id : 192.0.2.2
Priority        : None
    
```

Table 11: Assigned LP and communities for the import logical expressions summarizes the results for these different scenarios.

Table 11: 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
      import "[C]AND[A]AND[B]"
    exit all
    
```

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:PE-1# show router bgp routes hunt brief
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
---snip---
-----
RIB In Entries
-----
Network      : 2.2.2.2/32
Nexthop      : 192.0.2.2
Path Id      : None
From         : 192.0.2.2
Res. Protocol : ISIS                      Res. Metric   : 10
Res. Nexthop : 192.168.12.2
Local Pref. : 220                      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   : 2:2 1:1 3:3
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---
```

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
      import "[C]AND[A]OR[B]"
    exit all
```

```
*A:PE-1# show router bgp routes hunt brief
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
---snip---
-----
RIB In Entries
-----
Network      : 2.2.2.2/32
Nexthop      : 192.0.2.2
Path Id      : None
From         : 192.0.2.2
Res. Protocol : ISIS                      Res. Metric   : 10
Res. Nexthop : 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
```

```
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---
```

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
      import "[C]OR[A]OR[B]"
    exit all
```

```
*A:PE-1# show router bgp routes hunt brief
```

```
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
---snip---
-----
RIB In Entries
-----
Network      : 2.2.2.2/32
Nexthop      : 192.0.2.2
Path Id      : None
From         : 192.0.2.2
Res. Protocol : ISIS                Res. Metric   : 10
Res. Nexthop : 192.168.12.2
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   : 3:3
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---
```

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
      import "[C]OR[A]OR[B]" "D"
    exit all
```

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
    import "[C]OR[A]AND[B]"
  exit all
```

```
*A:PE-1# show router bgp routes hunt brief
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
---snip---
-----
RIB In Entries
-----
Network      : 2.2.2.2/32
Nexthop      : 192.0.2.2
Path Id      : None
From         : 192.0.2.2
Res. Protocol : ISIS           Res. Metric   : 10
Res. Nexthop : 192.168.12.2
Local Pref. : 220           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   : 2:2 3:3
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---
```

Table 12: Assigned LP and communities for the import logical expressions summarizes the results for these different scenarios.

Table 12: 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
  router "Base"
    policy-options
      begin
        community "D"
          members "4:4"
        exit
        community "E"
          members "5:5"
        exit
        community "F"
          members "6:6"
        exit
        community "G"
          members "7:7"
        exit
        policy-statement "D"
          entry 10
            from
              community "D"
            exit
            action accept
              local-preference 4
            exit
          exit
          default-action drop
          exit
        exit
        policy-statement "E"
          entry 10
            from
              community "E"
            exit
            action accept
              local-preference 5
            exit
          exit
          default-action drop
          exit
        exit
        policy-statement "F"
          entry 10
            from
              community "F"
            exit
            action accept
              local-preference 6
            exit
          exit
          default-action drop
          exit
        exit
        policy-statement "G"
          entry 10
            from
              community "G"
            exit
            action accept
```

```

                local-preference 7
            exit
        exit
        default-action drop
    exit
exit
commit
exit all
    
```

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
      import "([D]AND[E])OR([F]AND[G])"
    exit all
    
```

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:PE-1# show router bgp policy-test "([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:PE-1# show router bgp policy-test "[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:PE-1# show router bgp policy-test "[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:PE-1# show router bgp routes hunt brief
=====
      BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
---snip---
-----
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.  : 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    : 5:5
Cluster      : No Cluster Members
Originator Id : None                      Peer Router Id : 192.0.2.2

```

```
Fwd Class      : None          Priority       : None
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:PE-1# show router bgp policy-test "([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:PE-1# show router bgp routes hunt brief
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
---snip---
-----
RIB In Entries
-----
Network       : 2.2.2.2/32
Nexthop       : 192.0.2.2
Path Id       : None
From          : 192.0.2.2
Res. Protocol : ISIS                Res. Metric   : 10
Res. Nexthop  : 192.168.12.2
Local Pref. : 5                    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   : 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
    import "([D]OR[E])AND([F]OR[G])"
  exit all
```

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:PE-1# show router bgp policy-test "([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
-----
Routes : 1
=====
```

The route was rejected and, therefore, no policy was executed. The LP kept its default value of 100:

```
*A:PE-1# show router bgp routes hunt brief
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
---snip---
-----
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. : 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   : 5:5 4:4
Cluster       : No Cluster Members
Originator Id : None                      Peer Router Id : 192.0.2.2
```

```
Fwd Class      : None          Priority       : None
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:PE-1# show router bgp policy-test "([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 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:PE-2# show router bgp routes hunt brief
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
---snip---
-----
RIB In Entries
-----
Network       : 1.1.1.1/32
Nexthop       : 192.0.2.1
Path Id       : None
From          : 192.0.2.1
Res. Protocol : ISIS          Res. Metric   : 10
Res. Nexthop  : 192.168.12.1
Local Pref.   : 7             Interface Name : int-PE-2-PE-1
Aggregator AS : None         Aggregator    : None
Atomic Aggr.  : Not Atomic   MED           : None
AIGP Metric   : None         IGP Cost      : 10
Connector     : None
Community     : 5:5 7:7
Cluster       : No Cluster Members
```

```
Originator Id : None           Peer Router Id : 192.0.2.1
Fwd Class    : None           Priority       : None
Flags       : Used Valid Best IGP In-RTM
---snip---
```

Table 13: Assigned LP for the import logical expressions summarizes the results for these different scenarios.

Table 13: 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
  router "Base"
    policy-options
      begin
        community "X"
          members "10:10"
        exit
        community "Y"
          members "11:11"
        exit
        policy-statement "X"
          entry 10
            from
              community "X"
            exit
            action accept
              community add "Y"
            exit
          exit
        exit
      policy-statement "Y"
```

```

        entry 10
        from
            community "Y"
        exit
        action accept
            community remove "X" "Y"
            local-preference 9
        exit
    exit
exit
commit
exit all

```

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:PE-1# show router bgp routes hunt brief
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
---snip---
-----
RIB In Entries
-----
Network       : 2.2.2.2/32
NextHop       : 192.0.2.2
Path Id       : None
From          : 192.0.2.2
Res. Protocol : ISIS                Res. Metric   : 10
Res. NextHop  : 192.168.12.2
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   : 10:10
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---

```

The import policy "[X]AND[Y]" is configured on PE-1:

```

# on PE-1:
configure
    router "Base"
        bgp
            import "[X]AND[Y]"
        exit all

```

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:PE-1# show router bgp routes hunt brief
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
---snip---

```



```
-----  
RIB In Entries  
-----  
Network       : 2.2.2.2/32  
Nexthop       : 192.0.2.2  
Path Id       : None  
From          : 192.0.2.2  
Res. Protocol : ISIS                Res. Metric   : 10  
Res. Nexthop  : 192.168.12.2  
Local Pref. : 9                    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       : 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---
```

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

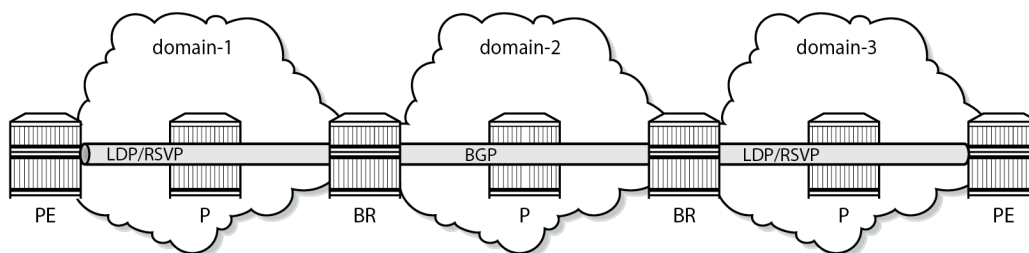
This chapter was initially written based on SR OS Release 15.0.R5, but the CLI in the current edition is based on SR OS Release 23.7.R2.

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 139: Stitching RSVP/LDP tunnels to BGP tunnels](#).

Figure 139: 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

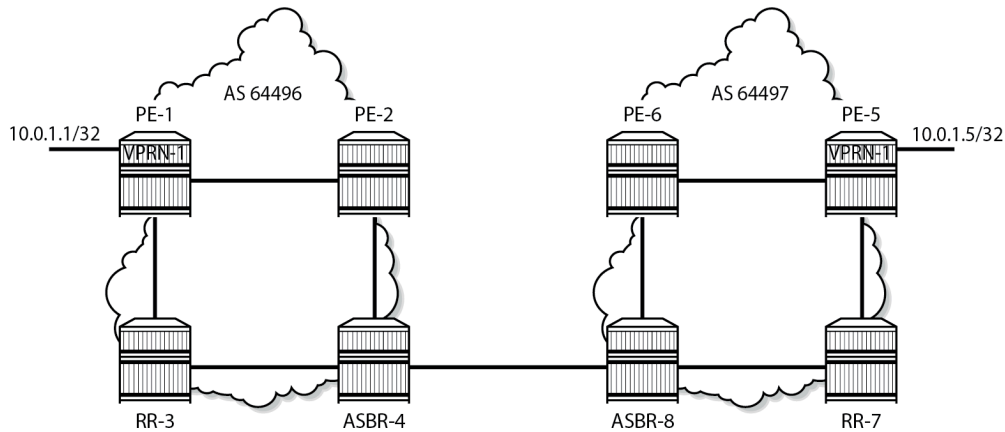
Redistributing /32 blackhole static routes does not require the **advertise-label pop** modifier; the label-IPv4 route is always advertised to the peer AS, and popped by the data plane.

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 140: 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 140: 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
router Base
  policy-options
  begin
  prefix-list "sys"
    prefix 192.0.2.0/29 prefix-length-range 32-32
  exit
  policy-statement "exp-sys"
    entry 10
      from
        protocol direct
        prefix-list "sys"
      exit
    action accept
```

```
        exit
      exit
    exit
  commit
exit
autonomous-system 64496
bgp
  loop-detect discard-route
  split-horizon
  group "IBGP"
    peer-as 64496
    neighbor 192.0.2.3
      family vpn-ipv4
    exit
    neighbor 192.0.2.4
      family label-ipv4
      export "exp-sys"
    exit
  exit
exit
exit
```

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
router Base
  autonomous-system 64496
  bgp
    loop-detect discard-route
    disable-route-table-install
    split-horizon
    group "EBGP-VPN"
      peer-as 64497
      local-address 192.0.2.3
      neighbor 192.0.2.7
        family vpn-ipv4
        multihop 10
        vpn-apply-export
        vpn-apply-import
      exit
    exit
    group "IBGP-VPN"
      cluster 192.0.2.3
      peer-as 64496
      neighbor 192.0.2.1
        family vpn-ipv4
      exit
      neighbor 192.0.2.2
        family vpn-ipv4
      exit
    exit
  exit
  no shutdown
exit
```

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** 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** 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
    rib-management
      label-ipv4
        route-table-import "imp-RR-pfxs"
      exit
    exit
  group "EBGP-label"
    export "exp-RR"
    advertise-inactive
    neighbor 192.168.48.2
      family label-ipv4
      peer-as 64497
    exit
  exit
  group "IBGP-label"
    next-hop-self
    cluster 192.0.2.4
    peer-as 64496
    neighbor 192.0.2.1
      family label-ipv4
    exit
    neighbor 192.0.2.2
      family label-ipv4
    exit
  exit
  no shutdown
exit
```

The *RR-pfxs* prefix list is the exact /32 address of RR-3. The *imp-RR-pfxs* policy in ASBR-4 matches the *RR-pfxs* prefix list in entry 10 with action accept and the **advertise-label pop** modifier. The *exp-RR* policy is used to advertise the RR prefix to the peer AS. The system prefixes of PE-1 and PE-2 are advertised by the PEs themselves. The policies on ASBR-4 are defined as follows:

```
# on ASBR-4:
configure
router Base
  policy-options
    begin
    prefix-list "RR-pfxs"
      prefix 192.0.2.3/32 exact
    exit
  policy-statement "exp-RR"
    entry 10
      from
        prefix-list "RR-pfxs"
      exit
      action accept
      exit
    exit
  exit
```

```

policy-statement "imp-RR-pfxs"
  entry 10
    from
      prefix-list "RR-pfxs"
    exit
    action accept
      advertise-label pop
    exit
  exit
exit
exit
commit
    
```

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-route-entry 192.0.2.7/32
      indirect 192.0.2.4
      tunnel-next-hop
        resolution disabled
      exit
    no shutdown
  exit
exit
    
```

The configuration of the nodes in AS 64497 is similar to the nodes in AS 64496; see [Figure 140: 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: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                               Label
-----
i     192.0.2.1/32                          n/a       None
      192.168.48.1                          None      20
      64496                                  524284
i     192.0.2.2/32                          n/a       None
      192.168.48.1                          None      10
      64496                                  524283
i     192.0.2.3/32                          n/a       10
      192.168.48.1                          None      n/a
      64496                                  524280
    
```

```
-----
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: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 00h07m29s 170
      192.168.48.1                    0
192.0.2.2/32                      Remote BGP_LABEL 00h07m29s 170
      192.168.48.1                    0
192.0.2.3/32                      Remote BGP_LABEL 00h02m21s 170
      192.168.48.1                    0
192.0.2.5/32                      Remote ISIS    00h09m03s 18
      192.168.68.1                    20
192.0.2.6/32                      Remote ISIS    00h09m03s 18
      192.168.68.1                    10
192.0.2.7/32                      Remote ISIS    00h09m03s 18
      192.168.78.1                    10
192.0.2.8/32                      Local  Local    00h09m04s  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: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  65537    9    192.168.24.1  20
192.0.2.2/32     ldp     MPLS  65538    9    192.168.24.1  10
192.0.2.5/32     bgp     MPLS  262146   12   192.168.48.2  1000
192.0.2.6/32     bgp     MPLS  262145   12   192.168.48.2  1000
192.0.2.7/32     bgp     MPLS  262147   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: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         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: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              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: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
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
```

```

Connector      : None
Community      : No Community Members
Cluster        : No Cluster Members
Originator Id  : None                Peer Router Id : 192.0.2.8
IPv4 Label     : 524284              Label Type     : SWAP
Lbl Allocation : NEXT-HOP
Origin         : IGP
AS-Path        : 64496
Route Tag      : 0
Neighbor-AS    : 64496
DB Orig Val    : NotFound            Final Orig Val : N/A
Source Class   : 0                  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: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                Interface Name : NotAvailable
Aggregator AS : None              Aggregator    : None
Atomic Aggr.  : Not Atomic        MED           : 10
AIGP Metric   : None              IGP Cost      : n/a
Connector     : None
Community     : No Community Members
Cluster       : No Cluster Members
Originator Id : None                Peer Router Id : 192.0.2.8
IPv4 Label    : 524280              Label Type    : POP
Lbl Allocation : NEXT-HOP
Origin        : IGP
AS-Path       : 64496
Route Tag     : 0
Neighbor-AS   : 64496
DB Orig Val   : N/A                Final Orig Val : N/A
Source Class  : 0                  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: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    13   0 00h03m15s 2/0/4 (VpnIPv4)
                   14   0
192.0.2.2
Def. Inst          64496    12   0 00h03m15s 1/0/5 (VpnIPv4)
                   15   0
192.0.2.7
Def. Inst          64497     9   0 00h01m26s 3/0/3 (VpnIPv4)
                   9   0
-----

```

Communication between VPRN-1 on PE-1 and on PE-5 is verified with a ping:

```

*A:PE-1# ping router 1 10.0.1.5
PING 10.0.1.5 56 data bytes
64 bytes from 10.0.1.5: icmp_seq=1 ttl=64 time=5.82ms.
64 bytes from 10.0.1.5: icmp_seq=2 ttl=64 time=6.49ms.
64 bytes from 10.0.1.5: icmp_seq=3 ttl=64 time=6.69ms.
64 bytes from 10.0.1.5: icmp_seq=4 ttl=64 time=6.43ms.
64 bytes from 10.0.1.5: icmp_seq=5 ttl=64 time=6.70ms.

---- 10.0.1.5 PING Statistics ----
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min = 5.82ms, avg = 6.42ms, max = 6.70ms, stddev = 0.319ms

```

Shutting down 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 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 **disable-route-table-install** 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 **disable-route-table-install** 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 **disable-route-table-install** policy action can be re-advertised, even if **advertise-inactive** is not configured toward the RIB-OUT peer and even if **next-hop-self** is configured toward the RIB-OUT peer. Because of the latter, incorrect use of this feature can blackhole traffic.



Note:

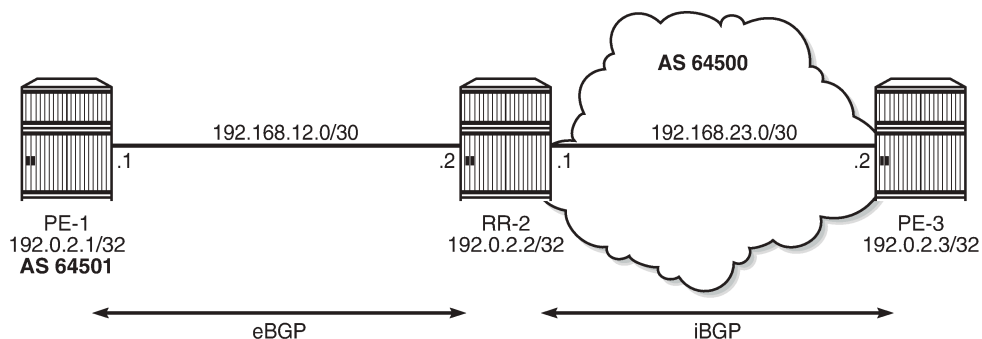
The **disable-route-table-install** command at the BGP instance level does not allow a route to be advertised under next-hop-self conditions.

The **disable-route-table-install** policy action overrides the effect of the **selective-label-ipv4-install** 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 **disable-route-table-install** policy action suppresses the installation.

Configuration

Figure 141: Example topology shows the example topology for this feature.

Figure 141: 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)

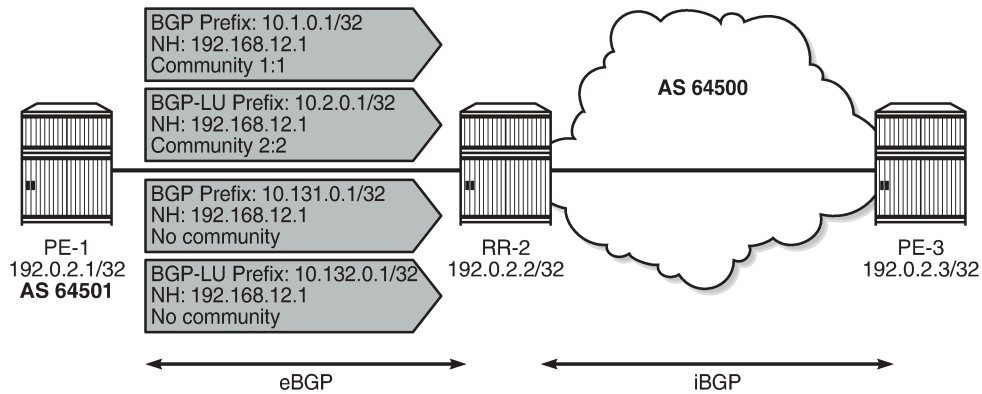
An eBGP session is established between PE-1 in AS 64501 and RR-2 in AS 64500, and an iBGP session between RR-2 and PE-3 in AS 64500 with next-hop-self. The BGP configuration on RR-2 is as follows:

```
# on RR-2:
configure
  router Base
    bgp
      split-horizon
      next-hop-resolution
      labeled-routes
        transport-tunnel
          family label-ipv4
            resolution-filter
            no ldp
            sr-isis
          exit
        resolution filter
      exit
    exit
  exit
  group "eBGP"
    local-as 64500
    peer-as 64501
    neighbor 192.168.12.1
      family ipv4 label-ipv4
      next-hop-self
    exit
  exit
  group "iBGP-IPv4"
    family ipv4 label-ipv4
    cluster 192.0.2.2
    peer-as 64500
    neighbor 192.0.2.3
      next-hop-self
    exit
  exit
  no shutdown
exit
```

Figure 142: 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 142: 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
router Base
  policy-options
  begin
  prefix-list "10.1.0.0/16"
    prefix 10.1.0.0/16 longer
  exit
  prefix-list "10.2.0.0/16"
    prefix 10.2.0.0/16 longer
  exit
  prefix-list "10.131.0.0/16"
    prefix 10.131.0.0/16 longer
  exit
  prefix-list "10.132.0.0/16"
    prefix 10.132.0.0/16 longer
  exit
  community "1:1"
    members "1:1"
  exit
  community "2:2"
    members "2:2"
  exit
  policy-statement "export-10.1"
    entry 10
      from
        prefix-list "10.1.0.0/16"
      exit
      to
        protocol bgp
      exit
      action accept
        community add "1:1"
      exit
    exit
  exit
  policy-statement "export-10.2"
    entry 10
      from
        prefix-list "10.2.0.0/16"
      exit
```

```

        to
        protocol bgp-label
    exit
    action accept
        community add "2:2"
    exit
    exit
exit
policy-statement "export-10.131"
    entry 10
        from
            prefix-list "10.131.0.0/16"
        exit
        to
            protocol bgp
        exit
        action accept
        exit
    exit
exit
policy-statement "export-10.132"
    entry 10
        from
            prefix-list "10.132.0.0/16"
        exit
        to
            protocol bgp-label
        exit
        action accept
        exit
    exit
exit
commit
exit
bgp
    split-horizon
    group "eBGP"
        local-as 64501
        peer-as 64500
        neighbor 192.168.12.2
            family ipv4 label-ipv4
            next-hop-self
            export "export-10.1" "export-10.2" "export-10.131"
                "export-10.132"
        exit
    exit
    no shutdown
exit

```

Initially, RR-2 has no import policy matching any of these four routes, so all these BGP and BGP-LU routes will be active:

```

*A: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                                   -
u*>i 10.131.0.1/32                           None      None
      192.168.12.1                          None      0
      64501                                   -
-----
Routes : 2
=====

```

```

*A: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
-----
u*>i 10.2.0.1/32                            None      None
      192.168.12.1                          None      0
      64501                                   524287
u*>i 10.132.0.1/32                           None      None
      192.168.12.1                          None      0
      64501                                   524287
-----
Routes : 2
=====

```

These routes are installed in the Routing Table Manager (RTM):

```

*A: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    00h43m49s  170
      192.168.12.1                          0
10.131.0.1/32                       Remote BGP    00h43m49s  170
      192.168.12.1                          0
-----
No. of Routes: 2

```

```

*A: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 00h43m49s 170
      192.168.12.1                                0
10.132.0.1/32                                     Remote BGP_LABEL 00h43m49s 170
      192.168.12.1                                0
-----
No. of Routes: 2
    
```

Also, the BGP-LU routes will each have an entry in the tunnel table, as follows:

```

*A: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
10.132.0.1/32   bgp        MPLS  262146   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
=====
    
```

All the BGP routes exported by PE-1 are installed in the FIB of RR-2, as follows:

```

*A:RR-2# show router fib 1 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)
10.131.0.1/32                                     BGP
      192.168.12.1 (int-RR-2-PE-1)
10.132.0.1/32                                     BGP_LABEL
      192.168.12.1 (int-RR-2-PE-1)
-----
Total Entries : 4
=====
    
```

Disable-route-table-install policy action

On RR-2, an import policy is configured that only accepts BGP routes with community "1:1" or "2:2"; all other routes match the policy **default-action disable-route-table-install**. This implies that the 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
router Base
  static-route-entry 10.128.0.0/9
    next-hop 192.168.12.1
    no shutdown
  exit
exit
policy-options
  begin
  community "1:1"
    members "1:1"
  exit
  community "2:2"
    members "2:2"
  exit
  policy-statement "bgp-install-1:1-2:2"
    entry 10
      from
        community "1:1"
      exit
      action accept
    exit
    entry 20
      from
        community "2:2"
      exit
      action accept
    exit
    default-action accept
    disable-route-table-install
  exit
exit
commit
info
exit
bgp
  group "eBGP"
    local-as 64500
    peer-as 64501
    neighbor 192.168.12.1
      family ipv4 label-ipv4
      next-hop-self
      import "bgp-install-1:1-2:2"
    exit
  exit
```

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: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
-----
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: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
-----
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: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: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 now only has one BGP route and one BGP-LU route, as follows:

```
*A: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: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:RR-2# show router fib 1 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: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: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: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
```

Disable-route-table-install command

The **disable-route-table-install** 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 **disable-route-table-install** command in the general **bgp** context, in combination with the **disable-route-table-install** parameter, which is part of the policy framework (**action** or **default-action**).

The **disable-route-table-install** command in the general **bgp** context is configured as follows:

```
# on RR-2:
configure
  router Base
    bgp
      disable-route-table-install
    exit
```

The rest of the BGP configuration (including import policy) remains unchanged.

This **disable-route-table-install** 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: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
=====
```

```
*A: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
=====
```

When a BGP route is suppressed because of a **disable-route-table-install** general BGP command match, no flag is added. The "Disable-RTM-Install" flag is only present for the route when the **disable-route-table-install** policy action is matched. The following output shows that the first route did not get an additional flag:

```
*A: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: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 **disable-route-table-install** command is configured and **next-hop-self** 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:RR-2# show router bgp routes hunt | match "RIB Out Entries" pre-lines 2 post-lines 50
-----
RIB Out Entries
-----
Routes : 2
=====
```

```
*A: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 **disable-route-table-install** 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** 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 CLI in the current edition corresponds to SR OS Release 20.7.R2.

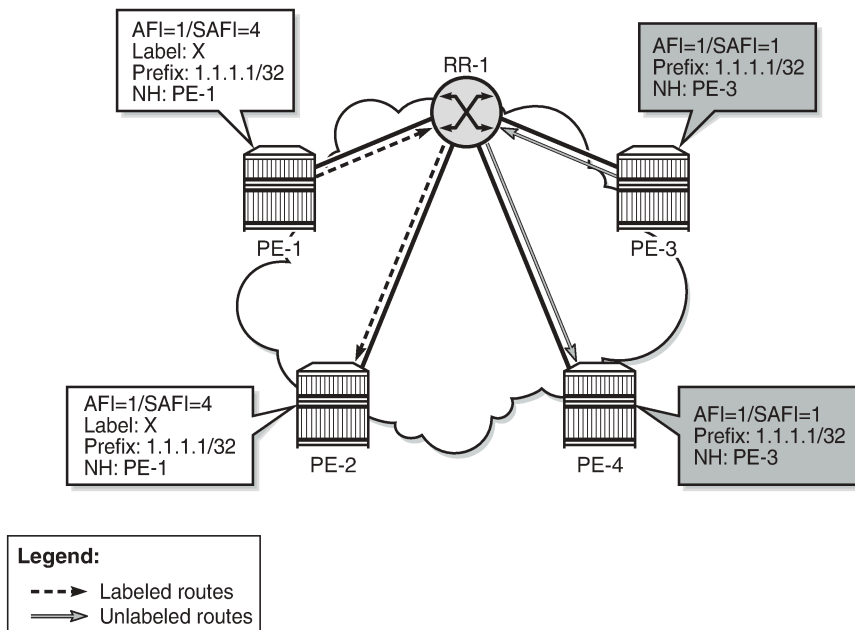
Release 14.0.R4 introduced separate BGP RIBs for labeled-unicast routes.

Overview

BGP separate labeled-IPv4 RIB implementation

[Figure 143: 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 143: RR-1 with separate labeled-IPv4 RIB implementation

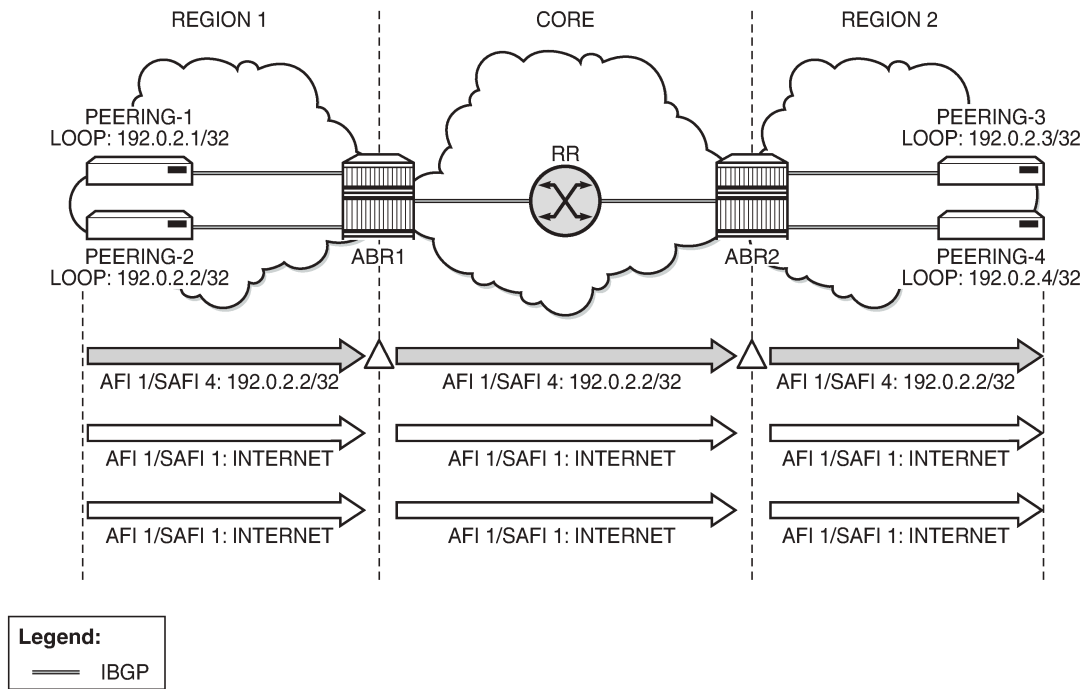


25972

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 144: [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 144: Seamless MPLS - Separate labeled-IPv4 implementation

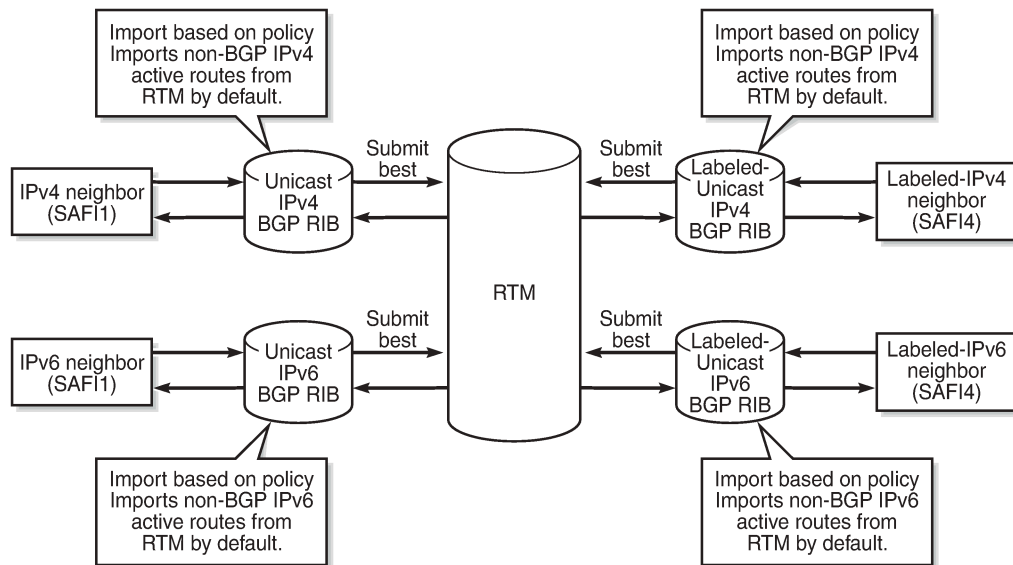


25973

RIB architecture

Figure 145: 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 145: 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.

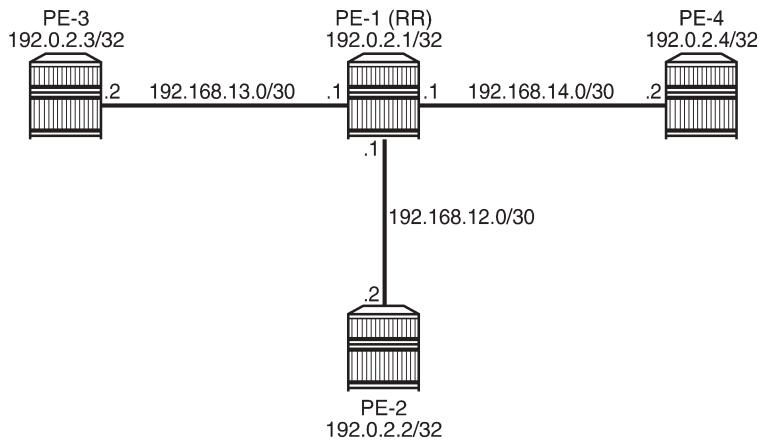
The default RTM preference for labeled IP routes is configurable (**label-preference**) in the **bgp** context of the base router or a VPRN. The default preference is 170.

Configuration

All the examples are based on labeled and unlabeled IPv4 addresses. For IPv6, the configuration is similar.

[Figure 146: Example IPv4 topology](#) shows the example topology using IPv4 addresses.

Figure 146: 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

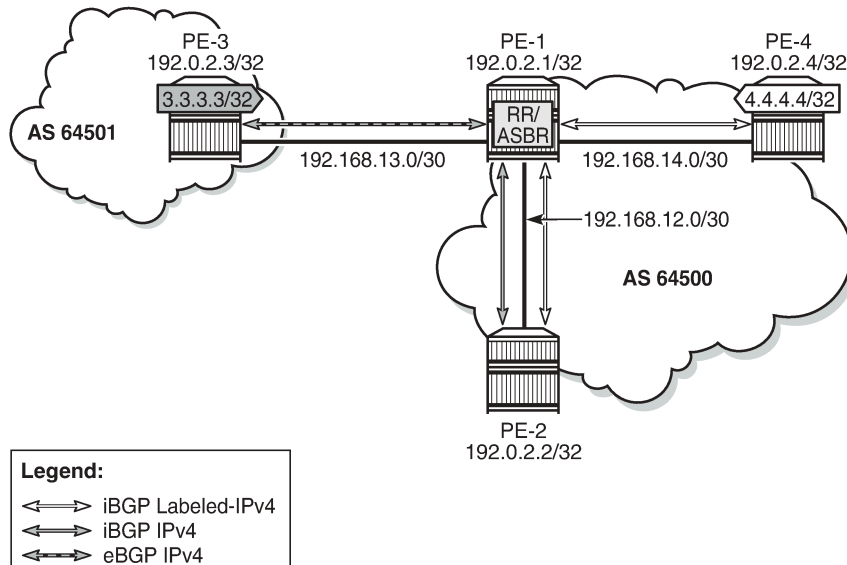
The following will be configured and verified:

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 147: BGP sessions shows the eBGP and iBGP sessions that are established between the nodes and the routes advertised for the loopback addresses.

Figure 147: 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
  autonomous-system 64500
  bgp
    split-horizon
    group "eBGP"
      peer-as 64501
      neighbor 192.168.13.2
      family ipv4
    exit
  exit
  group "iBGP"
    cluster 192.0.2.1
    export "export-bgp"
    peer-as 64500
    neighbor 192.0.2.2
      family ipv4 label-ipv4
    exit
    neighbor 192.0.2.4
      family label-ipv4
    exit
  exit
  no shutdown
exit
```

The BGP configuration on PE-2 is as follows:

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

```

autonomous-system 64500
bgp
  split-horizon
  group "iBGP"
    export "export-bgp"
    peer-as 64500
    neighbor 192.0.2.1
      family ipv4 label-ipv4
    exit
  exit
  no shutdown
exit

```

The BGP configuration on PE-3 in AS 64501 is as follows:

```

# on PE-3:
configure
router
  autonomous-system 64501
  bgp
    split-horizon
    group "eBGP"
      export "export-bgp"
      peer-as 64500
      neighbor 192.168.13.1
        family ipv4
      exit
    exit
    no shutdown
exit

```

The BGP configuration on PE-4 is as follows:

```

configure
router
  autonomous-system 64500
  bgp
    split-horizon
    group "iBGP"
      export "export-bgp"
      peer-as 64500
      neighbor 192.0.2.1
        family label-ipv4
      exit
    exit
    no shutdown
exit

```

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: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: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: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: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: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

```

```

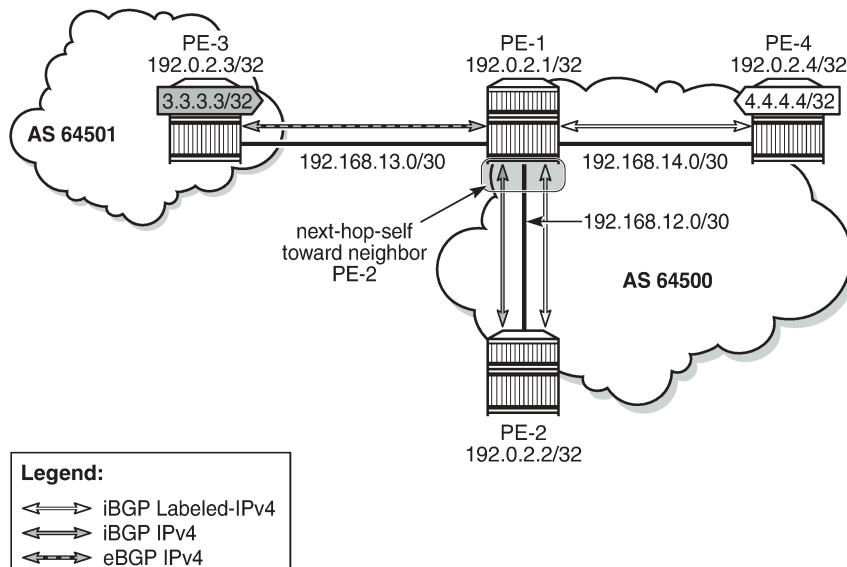
=====
BGP Routes
=====
Flag Network                               LocalPref  MED
  Nexthop (Router)                        Path-Id    IGP Cost
  As-Path                                   Label
-----
u*>i 4.4.4.4/32                             100        None
      192.0.2.4                             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 148: 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 148: 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
  bgp
    group "iBGP"
      neighbor 192.0.2.2
        next-hop-self
    exit
    
```

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: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: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
  bgp
    group "iBGP"
      neighbor 192.0.2.2
        no next-hop-self
    exit
  exit
```

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:

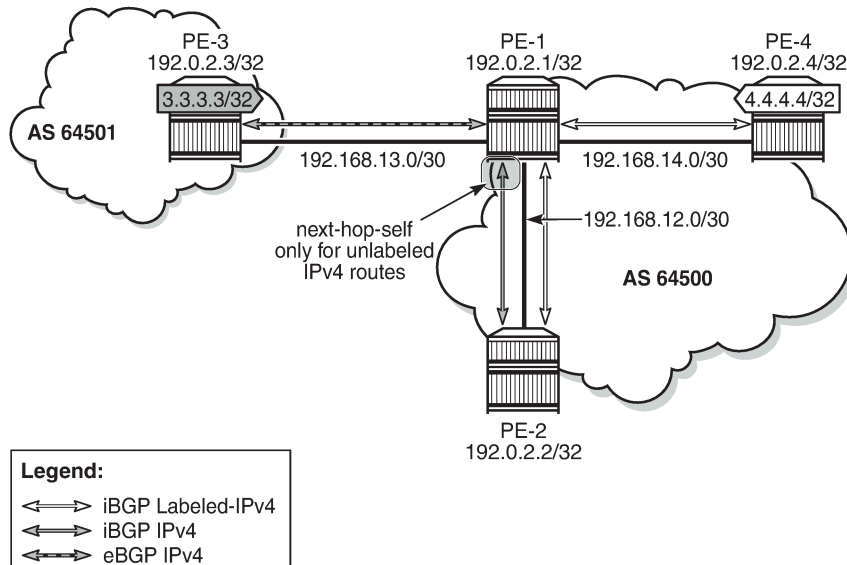
```
# on PE-1:
configure
router
  policy-options
  begin
  policy-statement "export-nhs"
  entry 10
  from
    protocol bgp
  exit
  action accept
  next-hop-self
  exit
  exit
  entry 20
  from
    protocol bgp-label
  exit
  action accept
  exit
  exit
  exit
  commit
```

The export policy "export-nhs" is configured for neighbor PE-2, as follows:

```
# on PE-1:
configure
router
  bgp
  group "iBGP"
  neighbor 192.0.2.2
  export "export-nhs"
  exit
```

Figure 149: 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 149: 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 BGP labeled-unicast (BGP-LU) route 4.4.4.4/32 will have next hop 192.0.2.4, as follows:

```
*A: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: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
=====
```

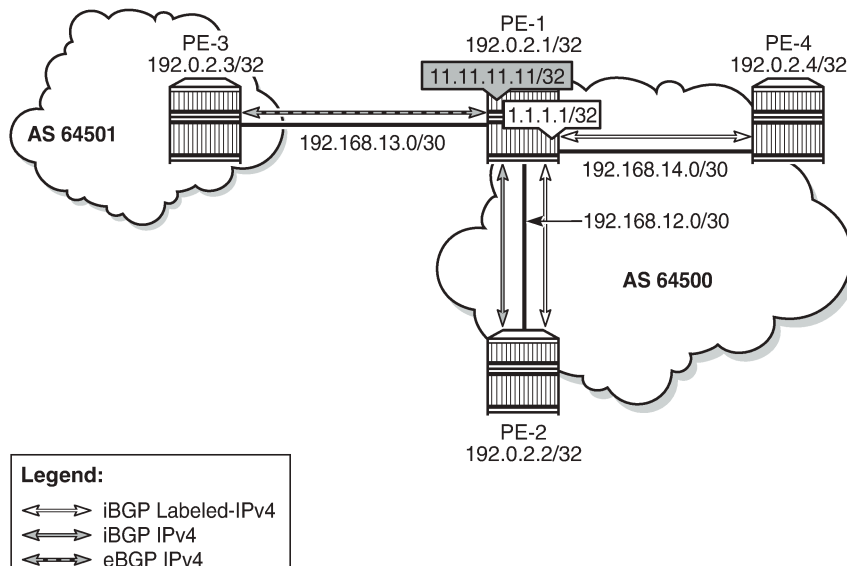
The export policy "export-nhs" toward neighbor PE-2 is removed as follows:

```
# on PE-1:
configure
router
  bgp
    group "iBGP"
      neighbor 192.0.2.2
        no export
    exit
```

Export policy to advertise route as labeled/unlabeled

Figure 150: 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 150: 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: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: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                               Label
-----
*>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
router
  policy-options
  begin
  prefix-list "1.1.1.1/32"
    prefix 1.1.1.1/32 exact
  exit
  prefix-list "11.11.11.11/32"
    prefix 11.11.11.11/32 exact
  exit
  policy-statement "export-bgp1"
  entry 10
    from
      prefix-list "1.1.1.1/32"
    exit
    to
      protocol bgp-label
    exit
    action accept
    exit
  exit
  entry 20
    from
      prefix-list "11.11.11.11/32"
    exit
    to
      protocol bgp
    exit
    action accept
    exit
  exit
  default-action drop
  exit
exit
commit
  
```

This policy is applied on PE-1 as an export policy for neighbor PE-2, as follows:

```

# on PE-1:
configure
router
  bgp
    group "iBGP"
      neighbor 192.0.2.2
        export "export-bgp1"
    exit
  
```

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: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 Nextthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
u*>i	11.11.11.11/32 192.0.2.1 No As-Path	100 None	None 10 -
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:PE-2# show router bgp routes label-ipv4
```

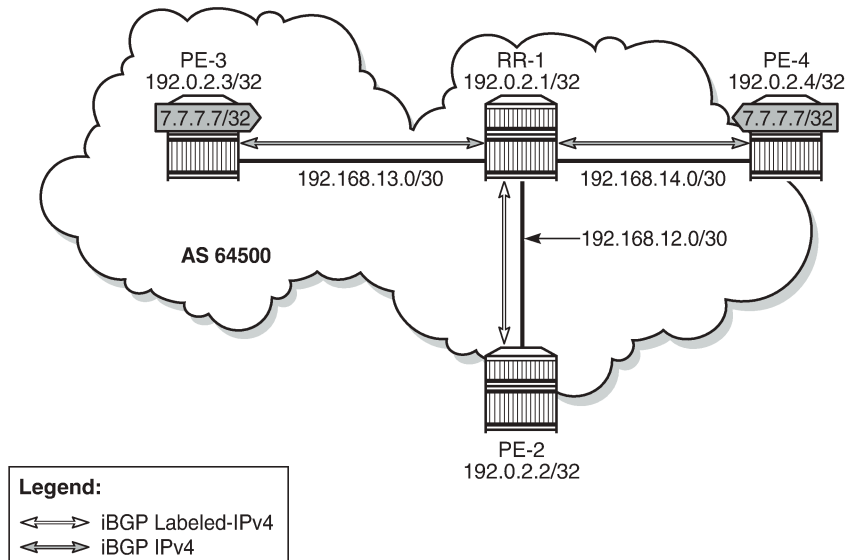
Flag	Network Nextthop (Router) As-Path	LocalPref Path-Id	MED IGP Cost Label
u*>i	1.1.1.1/32 192.0.2.1 No As-Path	100 None	None 10 524285
Routes : 1			

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 151: 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 151: RR with labeled and unlabeled BGP sessions



25980

On RR-1, BGP is configured as follows:

```
# on RR-1:
configure
router
  bgp
    split-horizon
    group "iBGP"
      cluster 192.0.2.1
      export "export-bgp"
      peer-as 64500
      neighbor 192.0.2.2
        family label-ipv4
        add-paths
          label-ipv4 send 2 receive
      exit
    exit
    neighbor 192.0.2.3
      family ipv4
    exit
    neighbor 192.0.2.4
      family ipv4
    exit
  exit
no shutdown
exit
```

RR-1 receives the prefix 7.7.7.7/32 from neighbors PE-3 and PE-4, as follows:

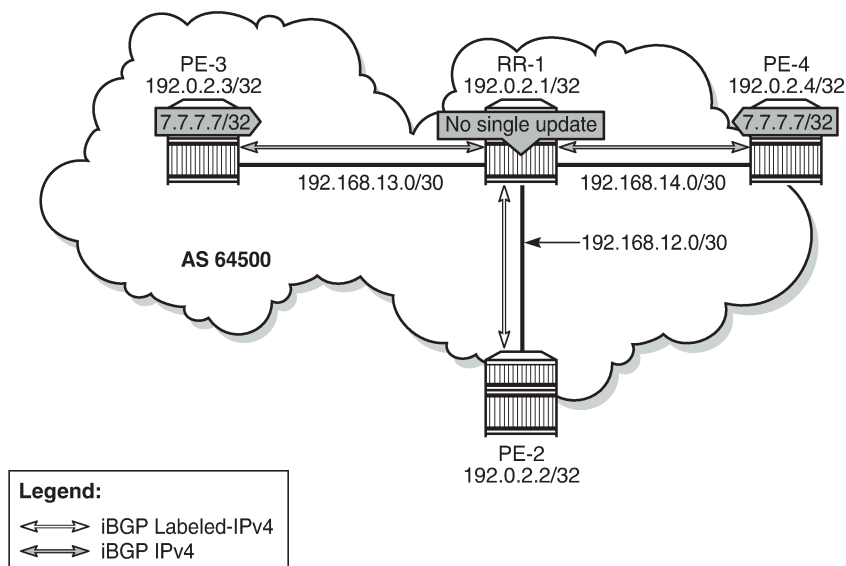
```
*A:RR-1# show router bgp routes 7.7.7.7/32
=====
BGP Router ID:192.0.2.1      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked, x - stale, > - best, b - backup, p - purge
```

```

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 152: Updates from unlabeled sessions not propagated to labeled sessions \(default\)](#).

Figure 152: 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: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: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
-----
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
router
  policy-options
  begin
  policy-statement "import-all"
  entry 10
  from
    protocol bgp
  exit
  action accept
  exit
  exit
  entry 20
  from
    protocol bgp-label
  exit
  action accept
  exit
  exit
  exit
  commit

```

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

```

```

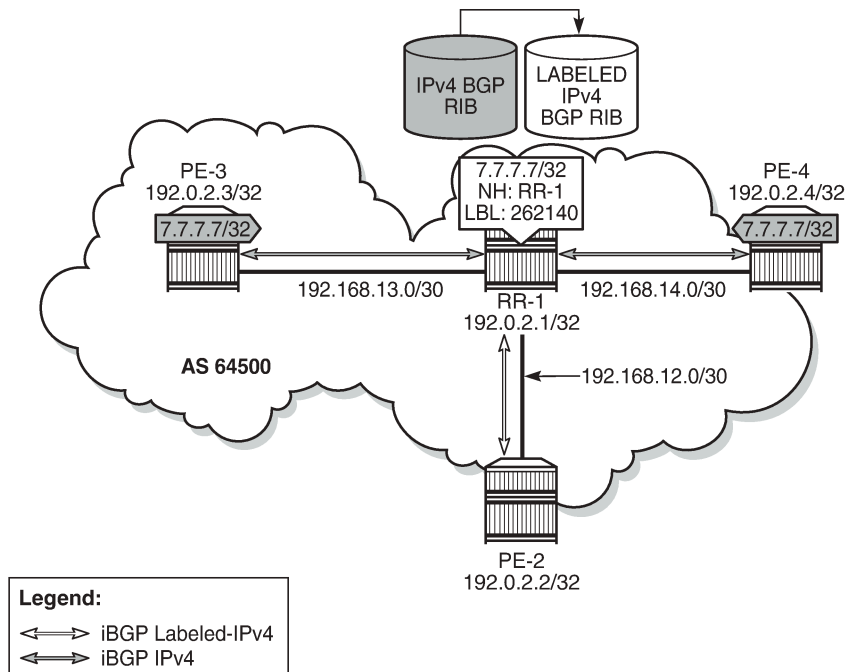
router
  bgp
    rib-management
      ipv4
        route-table-import "import-all"
      exit
    label-ipv4
      route-table-import "import-all"
    exit
  exit

```

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 153: RIB leaking from IPv4 BGP RIB to labeled-IPv4 BGP RIB shows this RIB leaking process.

Figure 153: 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: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                               None     524282
---snip---

```

The BGP update message is as follows:

```

4 2020/10/14 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: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                               None     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.

Customer document and product support



Customer documentation

[Customer documentation welcome page](#)



Technical support

[Product support portal](#)



Documentation feedback

[Customer documentation feedback](#)