7450 Ethernet Service Switch
7750 Service Router
7950 Extensible Routing System
Virtualized Service Router

Release 24.3.R1

Unicast Routing Protocols Guide

3HE 20118 AAAA TQZZA 01
Edition: 01
March 2024

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

Table of contents

1 Getting started 14
1.1 About this guide 14
1.2 Router configuration process 14
1.3 Conventions 15
1.3.1 Precautionary and information messages 15
1.3.2 Options or substeps in procedures and sequential workflows 16

2 RIP 17
2.1 RIP overview 17
2.1.1 RIP features 17
2.1.1.1 RIP version types 18
2.1.1.2 RIPv2 authentication 18
2.1.1.3 RIP packet format 18
2.1.1.4 BFD monitoring of RIP neighbor liveliness 20
2.2 RIPng 20
2.2.1 RIPng protocol 20
2.3 Common attributes 21
2.3.1 Metrics 21
2.3.2 Timers 21
2.3.3 Import and export policies 22
2.3.4 Hierarchical levels 22
2.4 RIP configuration process overview 22
2.5 Configuration notes 23
2.5.1 General 23
2.6 Configuring RIP with CLI 23
2.6.1 RIP and RIPng configuration overview 23
2.6.1.1 Preconfiguration requirements 23
2.6.1.2 RIP hierarchy 24
2.6.2 Basic RIP configuration 24
2.6.3 Common configuration tasks 25
2.6.3.1 Configuring interfaces 25
2.6.3.2 Configuring a route policy 26
2.6.3.3 Configuring RIP command options 28

2.6.3.4 Configuring global-level command options....28
2.6.3.5 Configuring group-level command options....28
2.6.3.6 Configuring neighbor-level command options....29
2.7 RIP configuration management tasks....30
2.7.1 Modifying RIP command options....30
2.7.2 Deleting a group....30
2.7.3 Deleting a neighbor....30

3 OSPF....32
3.1 Configuring OSPF....32
3.1.1 OSPF areas....32
3.1.1.1 Backbone area....33
3.1.1.2 Stub area....34
3.1.1.3 Not-so-stubby area....34
3.1.2 OSPFv3 authentication....38
3.1.3 OSPF graceful restart helper....38
3.1.3.1 BFD interaction with graceful restart....38
3.1.3.2 OSPFv3 graceful restart helper....39
3.1.4 Virtual links....40
3.1.5 Neighbors and adjacencies....40
3.1.5.1 Broadcast and point-to-point networks....40
3.1.5.2 Non-broadcast multi-access networks....40
3.1.6 Link-state advertisements....41
3.1.7 Metrics....41
3.1.8 Authentication....42
3.1.9 IP subnets....42
3.1.10 Preconfiguration recommendations....42
3.1.11 Multiple OSPF instances....43
3.1.11.1 Route export policies for OSPF....43
3.1.11.2 Preventing route redistribution loops....43
3.1.12 Multi-address support for OSPFv3....44
3.1.13 IP Fast-Reroute for OSPF and IS-IS prefixes....44
3.1.13.1 IP FRR configuration....45
3.1.13.2 ECMP considerations....46
3.1.13.3 IP FRR and RSVP shortcut....46
3.1.13.4 IP FRR and BGP next-hop resolution....46

3.1.13.5 OSPF and IS-IS support for LFA calculation....46
3.1.13.6 Multihomed prefix LFA extensions in OSPF....50
3.2 Loop-free alternate shortest path first policies....55
3.2.1 Configuring a route next hop policy template....55
3.2.1.1 Configuring affinity or admin group constraints....56
3.2.1.2 Configuring SRLG group constraints....58
3.2.1.3 Interaction of IP and MPLS admin group and SRLG....59
3.2.1.4 Configuring protection type and next hop type preferences....59
3.2.2 Application of route next hop policy template to an interface....60
3.2.3 Excluding interfaces and prefixes from LFA SPF....60
3.2.4 Modification to LFA next hop selection algorithm....63
3.3 SPF LSA filtering....64
3.4 FIB prioritization....64
3.5 Extended LSA support in OSPFv3....65
3.6 Support of multiple instances of router information LSA in OSPFv2 and OSPFv3....65
3.7 OSPF configuration process overview....66
3.8 Configuration notes....66
3.8.1 General....67
3.8.1.1 OSPF defaults....67
3.9 Configuring OSPF with CLI....67
3.9.1 OSPF configuration guidelines....67
3.9.2 Basic OSPF configurations....68
3.9.2.1 Configuring the router ID....69
3.9.3 Configuring OSPF components....71
3.9.3.1 Configuring OSPF....71
3.9.3.2 Configuring OSPF3....71
3.9.3.3 Configuring an OSPF or OSPF3 area....72
3.9.3.4 Configuring a stub area....73
3.9.3.5 Configuring a not-so-stubby area....74
3.9.3.6 Configuring a virtual link....75
3.9.3.7 Configuring an interface....77
3.9.3.8 Configuring authentication....79
3.9.3.9 Assigning a designated router....82
3.9.3.10 Configuring route summaries....83
3.9.3.11 Configuring route preferences....85
3.10 OSPF configuration management tasks....88

3.10.1 Modifying a router ID ... 88
3.10.2 Deleting a router ID ... 89
3.10.3 Modifying OSPF configuration ... 90

4 IS-IS ... 92
4.1 Configuring IS-IS ... 92
4.1.1 Routing ... 93
4.1.2 IS-IS frequently used terms ... 93
4.1.3 ISO network addressing ... 95
4.1.3.1 IS-IS PDU configuration ... 96
4.1.3.2 IS-IS operations ... 96
4.1.4 IS-IS route summarization ... 97
4.1.4.1 Partial SPF calculation ... 99
4.1.5 IS-IS multitopology support ... 99
4.1.5.1 Native IPv6 support ... 99
4.1.6 IS-IS administrative tags ... 99
4.1.6.1 Setting route tags ... 100
4.1.6.2 Using route tags ... 100
4.1.6.3 Unnumbered interface support ... 100
4.1.7 Multihomed prefix LFA extensions in IS-IS ... 101
4.1.7.1 Feature configuration ... 101
4.1.7.2 Feature applicability ... 102
4.1.7.3 RFC 8518 MHP LFA for IS-IS ... 103
4.2 FIB prioritization ... 103
4.3 IS-IS graceful restart helper ... 103
4.3.1 BFD interaction with graceful restart ... 103
4.4 IS-IS configuration process overview ... 104
4.5 Configuration notes ... 104
4.5.1 General ... 104
4.6 Configuring IS-IS with CLI ... 104
4.6.1 IS-IS configuration overview ... 105
4.6.1.1 Router levels ... 105
4.6.1.2 Configuring area address attributes ... 105
4.6.1.3 Interface level capacity ... 106
4.6.1.4 Route leaking ... 106
4.6.2 Basic IS-IS configuration ... 107

4.6.3 Common configuration tasks ... 112
4.6.4 Configuring IS-IS components ... 112
4.6.4.1 Enabling IS-IS ... 112
4.6.4.2 Modifying router-level parameters ... 112
4.6.4.3 Configuring ISO area addresses ... 113
4.6.4.4 Configuring global IS-IS parameters ... 114
4.6.4.5 Migration to IS-IS multitopology ... 115
4.6.4.6 Configuring IS-IS interfaces ... 119
4.6.4.7 Configuring IS-IS link groups ... 126
4.7 IS-IS configuration management tasks ... 126
4.7.1 Disabling IS-IS ... 126
4.7.2 Removing IS-IS ... 126
4.7.3 Modifying global IS-IS configuration ... 127
4.7.4 Modifying IS-IS interface configuration ... 127
4.7.5 Configuring authentication using keychains ... 127
4.7.6 Guidelines for configuring route leaking from level 2 to level 1 areas ... 129
4.7.7 Configuring route leaking from level 2 to level 1 areas ... 129
4.7.8 Redistributing external IS-IS routers ... 133
4.7.9 Specifying MAC addresses for all IS-IS routers ... 134

5 BGP ... 136
5.1 BGP overview ... 136
5.2 BGP sessions ... 136
5.2.1 BGP session states ... 138
5.2.2 Detecting BGP session failures ... 139
5.2.2.1 Peer tracking ... 139
5.2.2.2 Bidirectional Forwarding Detection ... 140
5.2.2.3 Fast external failover ... 140
5.2.3 High availability BGP sessions ... 141
5.2.3.1 BGP graceful restart ... 141
5.2.3.2 BGP long-lived graceful restart ... 142
5.2.4 BGP session security ... 144
5.2.4.1 TCP MD5 authentication ... 144
5.2.4.2 TTL security mechanism ... 144
5.2.5 BGP address family support for different session types ... 144
5.2.6 BGP groups ... 145

5.3 BGP design concepts....146
5.3.1 Route reflection....146
5.3.2 BGP confederations....149
5.4 BGP messages....149
5.4.1 Open message....150
5.4.1.1 Changing the autonomous system number....151
5.4.1.2 Changing a confederation number....152
5.4.1.3 BGP advertisement....152
5.4.2 Update message....152
5.4.3 Keepalive message....153
5.4.4 Notification message....153
5.4.4.1 Update message error handling....153
5.4.5 Route refresh message....154
5.5 BGP path attributes....154
5.5.1 Origin....155
5.5.2 AS path....156
5.5.2.1 AS override....157
5.5.2.2 Using local AS for ASN migration....158
5.5.2.3 4-octet autonomous system numbers....158
5.5.3 Next-hop....159
5.5.3.1 Unlabeled IPv4 unicast routes....159
5.5.3.2 Unlabeled IPv6 unicast routes....160
5.5.3.3 VPN-IPv4 routes....161
5.5.3.4 VPN-IPv6 routes....163
5.5.3.5 Label-IPv4 routes....165
5.5.3.6 Label-IPv6 routes....166
5.5.3.7 Next-hop resolution....167
5.5.3.8 Next-hop tracking....173
5.5.3.9 Next-hop indirection....173
5.5.3.10 Entropy label for RFC 8277 BGP labeled routes....173
5.5.4 MED....173
5.5.4.1 Deterministic MED....174
5.5.5 Local preference....174
5.5.6 Route aggregation path attributes....174
5.5.7 Community attributes....175
5.5.7.1 Standard communities....176

5.5.7.2 Extended communities....177
5.5.7.3 Large communities....178
5.5.8 Route reflection attributes....179
5.5.9 Multiprotocol BGP attributes....179
5.5.10 4-octet AS attributes....180
5.5.11 AIGP metric....180
5.6 BGP routing information base....181
5.6.1 RIB-IN features....181
5.6.1.1 BGP import policies....182
5.6.2 LOC-RIB features....182
5.6.2.1 BGP decision process....182
5.6.2.2 BGP route installation in the route table....184
5.6.2.3 BGP support for sticky ECMP....185
5.6.2.4 Weighted ECMP for BGP routes....194
5.6.2.5 BGP route installation in the tunnel table....195
5.6.2.6 Selective download of labeled unicast routes on next-hop-self routers....196
5.6.2.7 BGP fast reroute....198
5.6.2.8 QoS policy propagation through BGP....200
5.6.2.9 BGP policy accounting and policing....200
5.6.2.10 Route flap damping....202
5.6.3 RIB-OUT features....202
5.6.3.1 BGP export policies....203
5.6.3.2 Outbound route filtering....204
5.6.3.3 RT constrained route distribution....205
5.6.3.4 Min route advertisement interval....206
5.6.3.5 Advertise-inactive....207
5.6.3.6 Best-external....207
5.6.3.7 Add-paths....208
5.6.3.8 Split-horizon....210
5.7 BGP monitoring protocol....210
5.8 BGP applications....211
5.8.1 BGP FlowSpec....211
5.8.1.1 Validating received FlowSpec routes....215
5.8.1.2 Using flow routes to create dynamic filter entries....216
5.8.2 Configuration of TTL propagation for BGP labeled routes....217
5.8.2.1 TTL propagation for RFC 8277 labeled route at ingress LER....217

5.8.2.2 TTL propagation for RFC 8277 labeled routes at LSR....218
5.8.3 BGP prefix origin validation....219
5.8.4 BGP route leaking....221
5.8.5 BGP optimal route reflection....222
5.8.6 LSP tagging for BGP next-hops or prefixes and BGP-LU....223
5.8.7 BGP-LS....223
5.8.7.1 Supported BGP-LS components....224
5.8.8 BGP-LU traffic statistics....227
5.8.9 BGP Egress Peer Engineering using BGP Link State....227
5.8.9.1 Configuring BGP EPE....228
5.8.10 BGP Egress Peer Engineering using Labeled Unicast....230
5.9 BGP configuration process overview....231
5.10 Configuration notes....231
5.10.1 General....231
5.10.1.1 BGP defaults....231
5.10.1.2 BGP MIB notes....232
5.11 Configuring BGP with CLI....233
5.11.1 Configuration overview....233
5.11.1.1 Preconfiguration requirements....233
5.11.1.2 BGP hierarchy....233
5.11.1.3 Internal and external BGP configuration....233
5.11.1.4 Default external BGP route propagation behavior without policies....234
5.11.2 Basic BGP configuration....235
5.11.3 Common configuration tasks....236
5.11.3.1 Creating an autonomous system....237
5.11.3.2 Configuring a router ID....238
5.11.3.3 BGP confederations....239
5.11.3.4 BGP router reflectors....241
5.11.3.5 BGP components....244
5.12 BGP configuration management tasks....249
5.12.1 Modifying an AS number....249
5.12.2 Modifying a confederation number....250
5.12.3 Modifying the BGP router ID....250
5.12.4 Modifying the router-level router ID....251
5.12.5 Deleting a neighbor....252
5.12.6 Deleting groups....253

6 Route policies 255
6.1 Configuring route policies 255
6.1.1 Policy statements 255
6.1.1.1 Policy statement chaining and logical expressions 256
6.1.1.2 Routing policy subroutines 257
6.1.1.3 Policy evaluation command 257
6.1.1.4 Exclusive editing for policy configuration 257
6.1.1.5 Default action behavior 258
6.1.1.6 Denied IP unicast prefixes 258
6.1.1.7 Controlling route flapping 258
6.1.2 Regular expressions 259
6.1.3 BGP and OSPF route policy support 264
6.1.3.1 BGP route policies 265
6.1.3.2 Re-advertised route policies 265
6.1.3.3 Triggered policies 266
6.1.3.4 Set MED to IGP cost using route policies 266
6.1.3.5 BGP policy subroutines 267
6.1.3.6 Route policies for BGP next-hop resolution and peer tracking 267
6.1.4 Routing policy parameterization 267
6.1.5 When to use route policies 277
6.2 Route policy configuration process overview 277
6.3 Configuration notes 278
6.3.1 General 278
6.3.2 Policy reference checks 278
6.3.2.1 Known limitations 280
6.4 Configuring route policies with CLI 280
6.4.1 Route policy configuration overview 280
6.4.1.1 When to create routing policies 280
6.4.1.2 Default route policy actions 281
6.4.1.3 Policy evaluation 281
6.4.1.4 Damping 284
6.4.2 Basic configurations 285
6.4.3 Configuring route policy components 286
6.4.3.1 Beginning the policy statement 286
6.4.3.2 Creating a route policy 287

6.4.3.3 Configuring a default action....287
6.4.3.4 Configuring an entry....288
6.4.3.5 Configuring a community list....289
6.4.3.6 Configuring damping....289
6.4.3.7 Configuring a prefix list....290
6.5 Route policy configuration management tasks....290
6.5.1 Editing policy statements and parameters....290
6.5.2 Deleting an entry....291
6.5.3 Deleting a policy statement....292

7 Standards and protocol support....293
7.1 Access Node Control Protocol (ANCP)....293
7.2 Bidirectional Forwarding Detection (BFD)....293
7.3 Border Gateway Protocol (BGP)....293
7.4 Bridging and management....295
7.5 Broadband Network Gateway (BNG) Control and User Plane Separation (CUPS)....296
7.6 Certificate management....296
7.7 Circuit emulation....297
7.8 Ethernet....297
7.9 Ethernet VPN (EVPN)....297
7.10 gRPC Remote Procedure Calls (gRPC)....298
7.11 Intermediate System to Intermediate System (IS-IS)....298
7.12 Internet Protocol (IP) Fast Reroute (FRR)....299
7.13 Internet Protocol (IP) general....299
7.14 Internet Protocol (IP) multicast....301
7.15 Internet Protocol (IP) version 4....302
7.16 Internet Protocol (IP) version 6....303
7.17 Internet Protocol Security (IPsec)....304
7.18 Label Distribution Protocol (LDP)....305
7.19 Layer Two Tunneling Protocol (L2TP) Network Server (LNS)....306
7.20 Multiprotocol Label Switching (MPLS)....306
7.21 Multiprotocol Label Switching - Transport Profile (MPLS-TP)....307
7.22 Network Address Translation (NAT)....307
7.23 Network Configuration Protocol (NETCONF)....308
7.24 Open Shortest Path First (OSPF)....308
7.25 OpenFlow....309

7.26 Path Computation Element Protocol (PCEP) 309
7.27 Point-to-Point Protocol (PPP) 309
7.28 Policy management and credit control 310
7.29 Pseudowire (PW) 310
7.30 Quality of Service (QoS) 311
7.31 Remote Authentication Dial In User Service (RADIUS) 311
7.32 Resource Reservation Protocol - Traffic Engineering (RSVP-TE) 311
7.33 Routing Information Protocol (RIP) 312
7.34 Segment Routing (SR) 312
7.35 Simple Network Management Protocol (SNMP) 313
7.36 Timing 316
7.37 Two-Way Active Measurement Protocol (TWAMP) 316
7.38 Virtual Private LAN Service (VPLS) 317
7.39 Voice and video 317
7.40 Yet Another Next Generation (YANG) 317
7.41 Yet Another Next Generation (YANG) OpenConfig Models 317

1 Getting started

1.1 About this guide

This guide describes routing protocols including multicast, RIP, OSPF, IS-IS, BGP, and route policies provided by the router and presents configuration and implementation examples.

This document is organized into functional chapters and provides concepts and descriptions of the implementation flow, as well as Command Line Interface (CLI) syntax and command usage.

Note: Unless otherwise indicated, this guide uses classic CLI command syntax and configuration examples.

The topics and commands described in this document apply to the:

- 7450 ESS
- 7750 SR
- 7950 XRS
- Virtualized Service Router (VSR)

For a list of unsupported features by platform and chassis, see the *SR OS R24.x.Rx Software Release Notes*, part number 3HE 20152 000*x* TQZZA.

Command outputs shown in this guide are examples only; actual displays may differ depending on supported functionality and user configuration.

Note: The SR OS CLI trees and command descriptions can be found in the following guides:

- *7450 ESS, 7750 SR, 7950 XRS, and VSR Classic CLI Command Reference Guide*
- *7450 ESS, 7750 SR, 7950 XRS, and VSR Clear, Monitor, Show, and Tools CLI Command Reference Guide* (for both MD-CLI and Classic CLI)
- *7450 ESS, 7750 SR, 7950 XRS, and VSR MD-CLI Command Reference Guide*

Note: This guide generically covers Release 24.*x*.R*x* content and may contain some content that will be released in later maintenance loads. See the *SR OS R24.x.Rx Software Release Notes*, part number 3HE 20152 000*x* TQZZA, for information about features supported in each load of the Release 24.*x*.R*x* software.

1.2 Router configuration process

Table 1: Configuration process lists the tasks necessary to configure RIP, OSPF, IS-IS, and BGP protocols, and route policies on the 7450 ESS, 7750 SR, and 7950 XRS. This guide is presented in an overall logical configuration flow. Each section describes a software area and provides CLI syntax and command usage to configure parameters for a functional area.

Table 1: Configuration process

Area	Task	Section
RIP configuration	Configure RIP	Configuring RIP with CLI
	RIP configuration management	RIP configuration management tasks
OSPF configuration	Configure an LFA SPF policy	Loop-free alternate shortest path first policies
	Configure OSPF	Configuring OSPF with CLI
	OSPF configuration management	OSPF configuration management tasks
IS-IS configuration	Configure IS-IS	Configuring IS-IS with CLI
	Configure IS-IS components	Configuring IS-IS components
	IS-IS configuration management	IS-IS configuration management tasks
BGP configuration	Configure BGP	Configuring BGP with CLI
	BGP configuration management	BGP configuration management tasks
Route Policies	Configure route policies	Configuring route policies with CLI
	Configure route policy components	Configuring route policy components
	Route policy configuration management	Route policy configuration management tasks

1.3 Conventions

This section describes the general conventions used in this guide.

1.3.1 Precautionary and information messages

The following information symbols are used in the documentation.

DANGER: Danger warns that the described activity or situation may result in serious personal injury or death. An electric shock hazard could exist. Before you begin work on this equipment, be aware of hazards involving electrical circuitry, be familiar with networking environments, and implement accident prevention procedures.

WARNING: Warning indicates that the described activity or situation may, or will, cause equipment damage, serious performance problems, or loss of data.

Caution: Caution indicates that the described activity or situation may reduce your component or system performance.

Note: Note provides additional operational information.

Tip: Tip provides suggestions for use or best practices.

1.3.2 Options or substeps in procedures and sequential workflows

Options in a procedure or a sequential workflow are indicated by a bulleted list. In the following example, at step 1, the user must perform the described action. At step 2, the user must perform one of the listed options to complete the step.

Example: Options in a procedure

1. User must perform this step.
2. This step offers three options. User must perform one option to complete this step.
   - This is one option.
   - This is another option.
   - This is yet another option.

Substeps in a procedure or a sequential workflow are indicated by letters. In the following example, at step 1, the user must perform the described action. At step 2, the user must perform two substeps (a. and b.) to complete the step.

Example: Substeps in a procedure

1. User must perform this step.
2. User must perform all substeps to complete this action.
   a. This is one substep.
   b. This is another substep.

2 RIP

This chapter provides information about configuring the Routing Information Protocol (RIP).

2.1 RIP overview

RIP is an interior gateway protocol (IGP) that uses a distance-vector (Bellman-Ford) algorithm to determine the best route to a destination. The algorithm advertises network reachability by advertising prefix/mask and the metric (also known as hop count or cost). RIP selects the route with the lowest metric as the best route. In order for the protocol to provide complete information about routing, every router in the domain must participate in the protocol.

RIP is a routing protocol based on a distance vector (Bellman-Ford) algorithm, which advertises network reachability by advertising prefix/mask and the metric (also known as hop count or cost). RIP selects the route with the lowest metric as the best route. RIP differs from link-state database protocols, such as OSPF and IS-IS, in that RIP advertises reachability information directly and link-state-database-based protocols advertise topology information. Each node is responsible for calculating the reachability information from the topology.

The router software supports RIPv1 and RIPv2. RIPv1, specified in RFC 1058, was written and implemented before the introduction of classless interdomain routing (CIDR). It assumes the netmask information for non-local routes, based on the class the route belongs to:

- **class A**

  8 bit mask
- **class B**

  16 bit mask
- **class C**

  24 bit mask

RIPv2 was written after CIDR was developed and transmits netmask information with every route. Because of the support for CIDR routes and other enhancements in RIPv2 such as triggered updates, multicast advertisements, and authentication, most production networks use RIPv2. However, some older systems (hosts and routers) only support RIPv1, especially when RIP is used simply to advertise default routing information.

RIP is supported on all IP interfaces, including both network and access interfaces.

2.1.1 RIP features

RIP, a UDP-based protocol, updates its neighbors, and the neighbors update their neighbors, and so on. Each RIP host has a routing process that sends and receives datagrams on UDP port number 520.

Each RIP router advertises all RIP routes periodically via RIP updates. Each update can contain a maximum of 25 route advertisements. This limit is imposed by RIP specifications. RIP can sometimes be configured to send as many as 255 routes per update. The formats of the RIPv1 and RIPv2 updates are

slightly different and are shown below. Additionally, RIPv1 updates are sent to a broadcast address, RIPv2 updates can be either sent to a broadcast or multicast address (224.0.0.9). RIPv2 supports subnet masks, a feature that was not available in RIPv1.

A network address of 0.0.0.0 is considered a default route. A default route is used when it is not convenient to list every possible network in the RIP updates, and when one or more closely-connected gateways in the system are prepared to handle traffic to the networks that are not listed explicitly. These gateways create RIP entries for the address 0.0.0.0, as if it were a network to which they are connected.

2.1.1.1 RIP version types

SR OS allows the user to specify the RIP version that is sent to RIP neighbors and RIP updates that are accepted and processed. The following combinations are allowed:

- Send only RIPv1 or send only RIPv2 to either the broadcast or multicast address or send no messages.

  The default sends RIPv2 formatted messages to the broadcast address.
- Receive only RIPv1, receive only RIPv2, or receive both RIPv1 and RIPv2, or receive none.

  The default receives both.

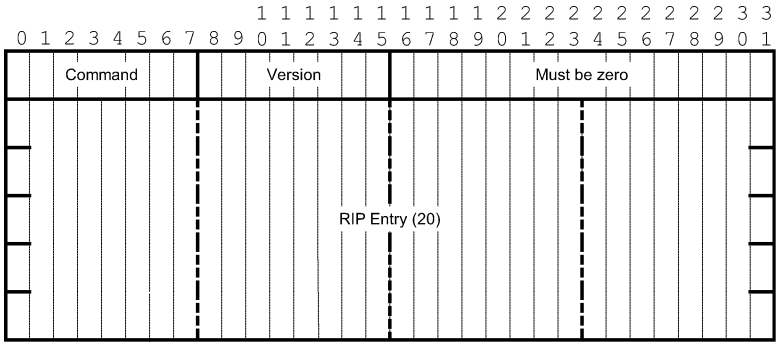
2.1.1.2 RIPv2 authentication

RIPv2 messages carry more information, which allows the use of a simple authentication mechanism to secure table updates. The router implementation enables the use of a simple password (plain text) or message digest (MD5) authentication.

2.1.1.3 RIP packet format

The RIP packet format is displayed in Figure 1: RIP packet format.

Figure 1: RIP packet format



A RIP packet consists of the following fields:

- **Command**

  This field indicates whether the packet is a request or a response message. The request asks the responding system to send all or part of its routing table. The response may be sent in response to a request, or it may be an unsolicited routing update generated by the sender.

- **Version**

  This field indicates the RIP version used. This field can signal different potentially incompatible versions.
- **Must be zero**

  Not used in RIPv1. This field provides backward compatibility with pre-standard varieties of RIP. The default value is zero.
- **Address family identifier (AFI)**

  This field indicates the type of address. RIP can carry routing information for several different protocols. Each entry in this field has an AFI to indicate the type of address being specified. The IP AFI is 2.
- **Address**

  This field indicates the IP address for the packet.
- **Metric**

  This field specifies the number of hops to the destination.
- **Mask**

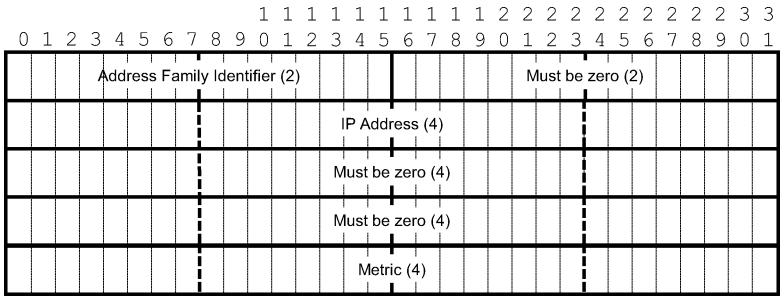
  This field specifies the IP address mask.
- **Next hop**

  This field specifies the IP address of the next router along the path to the destination.

2.1.1.3.1 RIPv1 packet format

There can be between 1 and 25 (inclusive) RIP entries. Figure 2: RIPv1 packet format displays RIPv1 packet format.

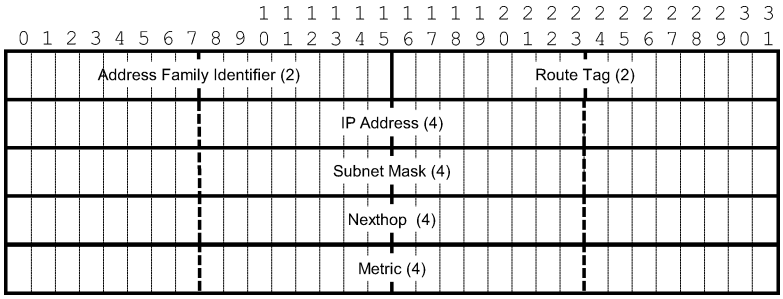
Figure 2: RIPv1 packet format



2.1.1.3.2 RIPv2 packet format

The RIP packet format is displayed in Figure 3: RIPv2 packet format.

Figure 3: RIPv2 packet format



The RIPv2 packets include the following fields:

- **Subnet Mask**

  The subnet mask for the entry. If this field is zero, no subnet mask has been specified for the entry.

- **Next hop**

  The IP address of the next hop to forward packets.

2.1.1.4 BFD monitoring of RIP neighbor liveliness

BFD can be used to monitor the liveliness of the RIP neighbors. If a BFD session, associated with a RIP neighbor fails, that RIP neighbor is declared down and all routes learned from that RIP neighbor are withdrawn from the associated route tables.

BFD is enabled for RIP by configuring the commands in the following context:

- **MD-CLI**

```
configure router rip group neighbor bfd-liveness true
```

- **classic CLI**

```
configure router rip group neighbor enable-bfd
```

BFD must also be enabled on the interface associated with the RIP neighbor. The **bfd** command sets the necessary transmit and receive intervals, as well as sets the optional multiplier.

2.2 RIPng

RIPng is the IPv6 form of the interior gateway protocol (IGP) Routing Information Protocol (RIP), originally implemented for IPv4 routing. This protocol is a distance vector routing protocol that periodically advertises IPv6 routing information to neighbors, typically through the use of UDP based multicast updates carrying a list of one or more entries, each containing an IPv6 prefix, prefix length, route metric and a possible route tag.

RIPng is supported in the base routing context and also as a PE-CE routing protocol within a VPRN context.

2.2.1 RIPng protocol

RIPng packets are sent using the UDP protocol and the protocol port number 521. Unsolicited updates messages are sent with 521 as both the source and destination port.

- **Source IP address**

  The Link-Local IPv6 address of the interface sending the RIPng packet is used as the source IP address of any RIPng update sent.

- **Destination IP address**

  The destination IP for any periodic or triggered update should be sent to the multicast group FF02::9, (all-rip-routers multicast group). When sending responses to an RIPng request, the RIPng response is sent to the unicast IP address of the requester.

Each route entry in an update message contains the following:

- IPv6 prefix
- prefix length
- route metric
- route tag (optional)

2.3 Common attributes

The following sections provide information about common RIP attributes.

2.3.1 Metrics

By default, RIP advertises all RIP routes to each peer every 30 seconds. RIP uses a hop count metric to determine the distance between the packet's source and destination. The metric/cost values for a valid route is 1 through 15. A metric value of 16 (infinity) indicates that the route is no longer valid and should be removed from the router's routing table.

Each router along the path increments the hop count value by 1. When a router receives a routing update with new or different destination information, the metric increments by one.

The maximum number of hops in a path is 15. If a router receives a routing update with a metric of 15 and contains a new or modified entry, increasing the metric value by one will cause the metric increment to 16 (infinity). Then, the destination is considered unreachable.

The router implementation of RIP uses *split horizon* with *poison reverse* to protect from such problems as "counting to infinity". Split horizon with poison reverse means that routes learned from a neighbor through a specified interface are advertised in updates out of the same interface but with a metric of 16 (infinity).

2.3.2 Timers

RIP uses the following timers to determine the frequency of RIP updates and the duration that routes are maintained.

- **update**

  Times the interval between periodic routing updates.

- **timeout**

  This timer is initialized when a route is established and any time an update message is received for the route. When this timer expires, the route is no longer valid. It is retained in the table for a short time, so that neighbors can be notified that the route has been dropped.

- **flush**

  When the flush timer expires, the route is removed from the tables.

2.3.3 Import and export policies

Routing policies can control the content of the routing tables, advertised routes, and the best route to reach a destination. Import route policies determine which routes are accepted from RIP neighbors. Export route policies determine which routes are exported from the route table to RIP. By default, RIP does not export learned routes to its neighbors.

There are no default routing policies. A policy must be created explicitly and applied to a RIP import or export command.

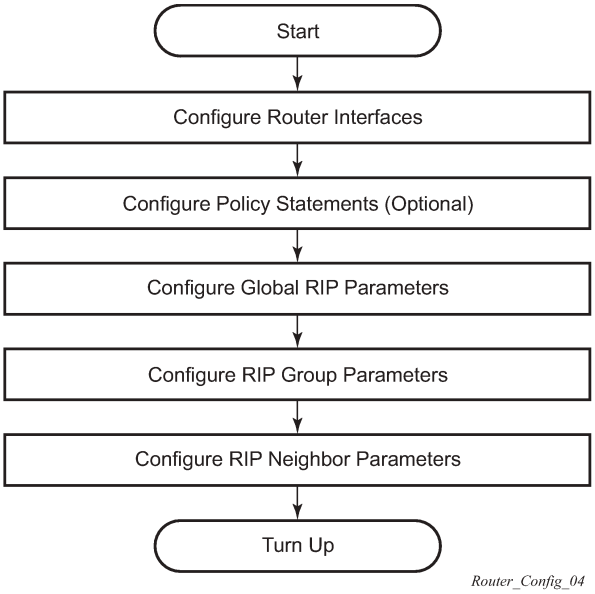
2.3.4 Hierarchical levels

The minimum RIP configuration must define one group and one neighbor. For more information about RIP hierarchy levels, see Basic RIP configuration.

2.4 RIP configuration process overview

Figure 4: RIP configuration and implementation flow displays the process to configure RIP command options.

Figure 4: RIP configuration and implementation flow



2.5 Configuration notes

This section describes RIP configuration restrictions.

2.5.1 General

Before RIP neighbor command options can be configured, router interfaces must be configured.

RIP must be explicitly created for each router interface. There are no default RIP instances on a router.

2.6 Configuring RIP with CLI

This section provides information to configure Routing Information Protocol (RIP) using the command line interface.

2.6.1 RIP and RIPng configuration overview

2.6.1.1 Preconfiguration requirements

Configure the following entities before beginning the RIP configuration.

Optionally, use the commands in the following context to define the policy statements:

- **MD-CLI**

```
configure policy-options
```

- **classic CLI**

```
configure router policy-options
```

2.6.1.2 RIP hierarchy

RIP is configured in the **configure router rip** context. RIP is not enabled by default.

Three hierarchical levels are included in RIP configurations in the classic CLI:

- global
- group
- neighbor

Commands and command options configured at the global level are inherited by the group and neighbor levels. However, command options configured at the group and neighbor levels take precedence over global configurations.

2.6.2 Basic RIP configuration

This section provides information to configure RIP and examples of common configuration tasks. For a router to accept RIP updates, in the **configure router rip** context, you must define at least one group and one neighbor. A router ignores updates received from routers on interfaces not configured for RIP. Configuring other RIP commands and parameters is optional.

By default, the local router imports all routes from this neighbor and does not advertise routes. The router receives both RIPv1 and RIPv2 update messages with 25 to 255 route entries per message.

The RIP configuration commands have three primary configuration levels:

- **rip** for RIP global configurations
- **group** for RIP group configurations
- **neighbor** for RIP neighbor configurations

Within these levels, the RIP configuration commands are identical. For repeated commands, the value most specific to the neighboring router is used. Therefore, a RIP group-specific command takes precedence over a global RIP command. A neighbor-specific configuration statement takes precedence over a global RIP and group-specific command. For example, if the user modifies a RIP neighbor-level command default, the new value takes precedence over group- and global-level settings.

At a minimum, the group- and neighbor-level RIP parameters must be configured in the **configure router rip** context.

The following example displays a basic RIP configuration.

Output example: MD-CLI

```
[ex:/configure router "Base" rip]
A:admin@node-2# info
    group "RIP-ALA-A" {
```

```
        neighbor "to-ALA-4"
    }
```

Output example: classic CLI

```
A:node-2>config>router>rip# info
----------------------------------------------
            group "RIP-ALA-A"
                neighbor "to-ALA-4"
                    no shutdown
----------------------------------------------
```

2.6.3 Common configuration tasks

About this task

This section provides an overview of RIP configuration tasks and the CLI commands.
Configure RIP hierarchically using the global level (applies to all peers), the group level (applies to all peers in peer-group), or the neighbor level (only applies to the specified interface). By default, group members inherit the group's configuration parameters; however, a parameter can be modified on a per-member basis without affecting the group-level command options. For more information about the hierarchy of RIP configuration levels, see RIP hierarchy and Basic RIP configuration.

The user must explicitly create all RIP instances on each device. After the instances are created, RIP is administratively enabled.

To configure RIP, perform the following steps:

Procedure

Step 1. Configure the interfaces.

Step 2. Optionally configure the policy statements.

Step 3. Enable the RIP.

Step 4. Configure the group command options.

Step 5. Configure the neighbor command options.

2.6.3.1 Configuring interfaces

The following command sequences create a logical IP interface. The logical interface can associate attributes like an IP address, port, Link Aggregation Group (LAG), or the system. For more information about configuring interfaces, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Interface Configuration Guide*.

Use the commands in the following context to configure a network interface.

```
configure router interface
```

Note: The **link-local-modifier** command can only be configured in the classic CLI.

The following example displays the interface information.

Output example: MD-CLI

```
[ex:/configure router "Base" interface "itf1"]
A:admin@node-2# info
    port 1/1/1
    ipv4 {
        primary {
            address 10.10.10.1
            prefix-length 24
        }
    }
    ipv6 {
        address 2000:1:: {
            prefix-length 64
        }
        neighbor-discovery {
            secure-nd {
                admin-state enable
            }
        }
    }
```

Output example: classic CLI

```
A:node-2>config>router# info
#------------------------------------------------
echo "IP Configuration"
#------------------------------------------------
      interface "itf1"
            address 10.10.10.1/24
            port 1/1/1
            ipv6
                secure-nd
                    link-local-modifier 0xbe571f90d13a73ebde8ee34b0f90e5ad
                    no shutdown
                exit
                address 2000:1::/64 modifier 0x2ec57d275ba420d094deaeb7f0545827
            exit
            no shutdown
```

2.6.3.2 Configuring a route policy

Use the import route policy to filter routes imported by the local router from its neighbors. If no match is found, the local router does not import any routes.

Use the export route policy to determine which routes are exported from the route table to RIP. By default, RIP does not export learned routes to its neighbors. If no export policy is specified, non-RIP routes are not exported from the routing table manager to RIP.

If multiple policy names are specified, the policies are evaluated in the order they are specified. The first policy that matches is applied. If multiple export commands are issued, the last command entered overrides the previous command. A maximum of five policy names can be specified.

This section only provides brief instructions to configure route policies. For more details, see the Route policy configuration overview chapter.

Use the following command to enter the mode to create or edit route policies:

- **MD-CLI**

```
configure policy-options
```

- **classic CLI**
  In the classic CLI, use the **begin** command in the following context to start creating or editing route policies.

```
configure router policy-options
```

Other editing commands include:

- the **commit** command saves and enables changes made to route policies during a session
- the **abort** command discards changes that have been made to route policies during a session

Use the commands in the following context to configure a policy to use for the RIP global, group, and neighbor commands:

- **MD-CLI**

```
configure policy-options
```

- **classic CLI**

```
configure router policy-options
```

The following example display the policy option information.

Example: MD-CLI

```
[ex:/configure policy-options]
A:admin@node-2# info
    policy-statement "RIP-policy" {
        entry 1 {
            action {
                action-type accept
            }
        }
        default-action {
            action-type reject
        }
    }
```

Example: classic CLI

```
A:node-2>config>router>policy-options# info
----------------------------------------------
            policy-statement "RIP-policy"
                description "this is a test RIP policy"
                entry 1
                    action accept
                    exit
                exit
                default-action drop
                exit
----------------------------------------------
```

Use the **begin** command in the **configure router policy-options** context to enter edit mode and the **commit** command to save the changes.

2.6.3.3 Configuring RIP command options

Use the commands in the following context to configure RIP command options at the global, group, and neighbor level.

```
configure router rip group neighbor
```

2.6.3.4 Configuring global-level command options

After the RIP protocol instance is created, the no shutdown command is not required because RIP is administratively enabled upon creation. To enable RIP on a router, at least one group and one neighbor must be configured. There are no default groups or neighbors. Each group and neighbor must be explicitly configured.

Note: Careful planning is essential to implement commands that can affect the behavior of global, group, and neighbor levels. Because the RIP commands are hierarchical, analyze the values that can disable features on a specific level.

Use the commands in the following context to configure global-level RIP command options.

```
configure router rip
```

The following example displays the RIP group configuration.

Output example: MD-CLI

```
[ex:/configure router "Base" rip]
A:admin@node-2# info
    authentication-key "TRCHaEdwwfZ8PxeZSkzmH/nOiAQxBJXzPGXj hash2"
    authentication-type password
    timers {
        update 300
        timeout 600
        flush 600
    }
```

Output example: classic CLI

```
A:node-2>config>router>rip$ info
----------------------------------------------
            authentication-key "TRCHaEdwwfZ8PxeZSkzmH/nOiAQxBJXzPGXj" hash2
            authentication-type password
            timers 300 600 600
            no shutdown
----------------------------------------------
```

2.6.3.5 Configuring group-level command options

A group is a collection of related RIP peers. The group name should be a descriptive name for the group. Follow your group, name, and ID naming conventions for consistency and to help when troubleshooting faults.

All command options configured for a group are applied to the group and are inherited by each peer (neighbor), but a group command option can be overridden on a specific neighbor-level basis.

Use the commands in the following context to configure a group.

```
configure router rip group
```

The following example displays the RIP group configuration.

Output example: MD-CLI

```
[ex:/configure router "Base" rip]
A:admin@node-2# info
    authentication-key "TRCHaEdwwfZ8PxeZSkzmH/nOiAQxBJXzPGXj hash2"
    authentication-type password
    timers {
        update 300
        timeout 600
        flush 600
    }
    group "headquarters" {
        description "Mt. View"
    }
```

Output example: classic CLI

```
A:node-2>config>router>rip$ info
----------------------------------------------
            authentication-key "TRCHaEdwwfZ8PxeZSkzmH/nOiAQxBJXzPGXj" hash2
            authentication-type password
            timers 300 600 600
            group "headquarters"
                description "Mt. View"
                no shutdown
            exit
            no shutdown
----------------------------------------------
```

2.6.3.6 Configuring neighbor-level command options

After you create a group name and assign options, add neighbor interfaces within the same group. All command options configured for the peer group level are applied to each neighbor, but a group command option can be overridden on a specific neighbor basis.

Use the commands in the following context to add a neighbor to a group and define options that override the same group-level command value.

```
configure router rip group neighbor
```

The following example displays the neighbor configured in group "headquarters".

Output example: MD-CLI

```
[ex:/configure router "Base" rip group "headquarters" neighbor "ferguson-274"]
A:admin@node-2# info
    message-size 255
    preference 255
    split-horizon true
```

Output example: classic CLI

```
A:node-2>config>router>rip>group>neighbor$ info
----------------------------------------------
                    message-size 255
                    preference 255
                    split-horizon enable
                    no shutdown
----------------------------------------------
```

2.7 RIP configuration management tasks

This section provides information about RIP configuration management tasks.

2.7.1 Modifying RIP command options

Modify, add, or remove RIP command options in the CLI. The changes are applied immediately. For the complete list of CLI commands, see the Configuring RIP command options.

2.7.2 Deleting a group

In the classic CLI, you must administratively disable a group before deleting it using the following command:

```
configure router rip group shutdown
```

Deleting the group without first shutting it down displays the following message.

```
INFO: RIP #1204 group should be administratively down - virtual router index 1,group
RIP-ALA-4
```

2.7.3 Deleting a neighbor

In the classic CLI, you must administratively disable a neighbor before deleting it using the following command:

```
configure router rip group neighbor shutdown
```

Deleting the neighbor without first shutting it down causes the following message to appear.

```
INFO: RIP #1101 neighbor should be administratively down - virtual router index
```

3 OSPF

3.1 Configuring OSPF

Open Shortest Path First (OSPF) is a hierarchical link state protocol. OSPF is an interior gateway protocol (IGP) used within large autonomous systems (ASs). OSPF routers exchange state, cost, and other relevant interface information with neighbors. The information exchange enables all participating routers to establish a network topology map. Each router applies the Dijkstra algorithm to calculate the shortest path to each destination in the network. The resulting OSPF forwarding table is submitted to the routing table manager to calculate the routing table.

When a router is started with OSPF configured, OSPF, along with the routing-protocol data structures, is initialized and waits for indications from lower-layer protocols that its interfaces are functional. Nokia's implementation of OSPF conforms to OSPF Version 2 specifications presented in RFC 2328, OSPF Version 2 and OSPF Version 3 specifications presented in RFC 2740, OSPF for IPv6. Routers running OSPF can be enabled with minimal configuration. All default and command parameters can be modified.

Changes between OSPF for IPv4 and OSPF3 for IPv6 include the following:

- Addressing semantics have been removed from OSPF packets and the basic link-state advertisements (LSAs). New LSAs have been created to carry IPv6 addresses and prefixes.
- OSPF3 runs on a per-link basis, instead of on a per-IP-subnet basis.
- Flooding scope for LSAs has been generalized.
- Unlike OSPFv2, OSPFv3 authentication relies on IPV6's authentication header and encapsulating security payload.
- Most packets in OSPF for IPv6 are almost as compact as those in OSPF for IPv4, even with the larger IPv6 addresses.
- Most field and packet-size limitations present in OSPF for IPv4 have been relaxed.
- Option handling has been made more flexible.

Key OSPF features are:

- backbone areas
- stub areas
- not-so-stubby areas (NSSAs)
- virtual links
- authentication
- route redistribution
- routing interface parameters
- OSPF-TE extensions (Nokia's implementation allows MPLS fast reroute)

3.1.1 OSPF areas

The hierarchical design of OSPF allows a collection of networks to be grouped into a logical area. An area's topology is concealed from the rest of the AS which significantly reduces OSPF protocol traffic. With the correct network design and area route aggregation, the size of the route-table can be drastically reduced which results in decreased OSPF route calculation time and topological database size.

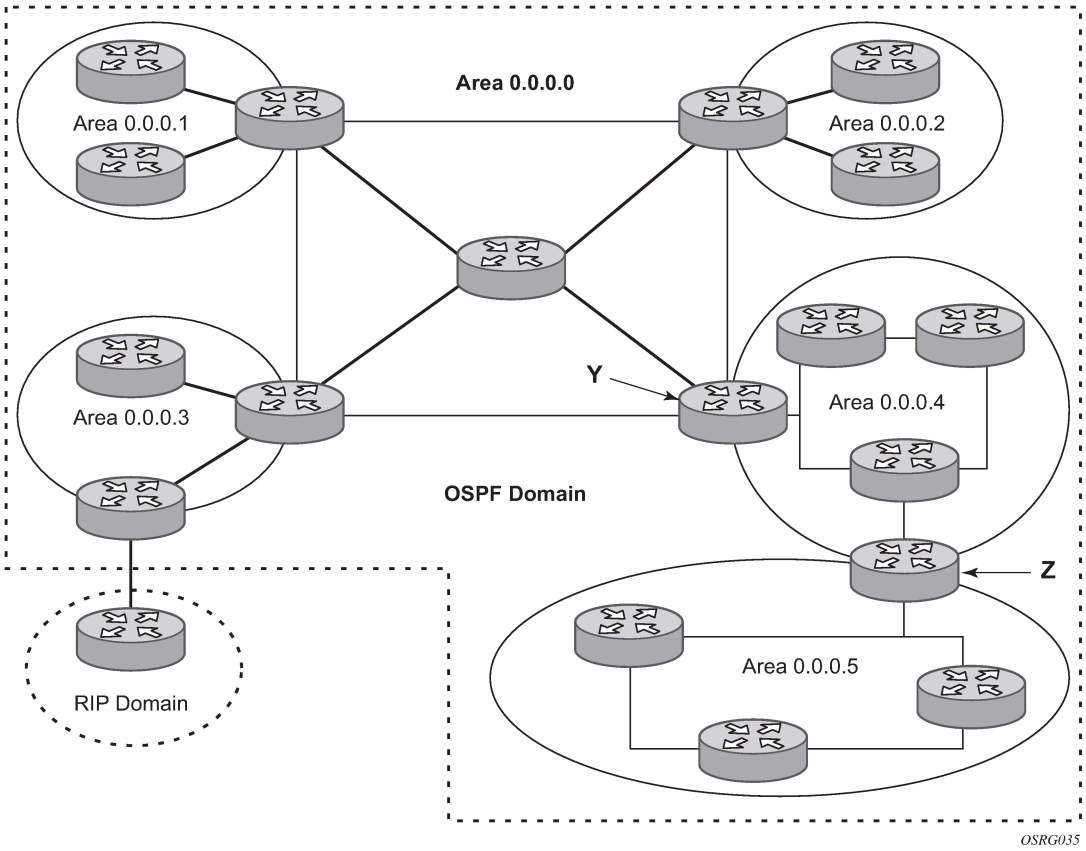
Routing in the AS takes place on two levels, depending on whether the source and destination of a packet reside in the same area (intra-area routing) or different areas (inter-area routing). In intra-area routing, the packet is routed solely on information obtained within the area; no routing information obtained from outside the area is used.

Routers that belong to more than one area are called area border routers (ABRs). An ABR maintains a separate topological database for each area it is connected to. Every router that belongs to the same area has an identical topological database for that area.

3.1.1.1 Backbone area

The OSPF backbone area, area 0.0.0.0, must be contiguous and all other areas must be connected to the backbone area. The backbone distributes routing information between areas. If it is not practical to connect an area to the backbone (see area 0.0.0.5 in Figure 5: Backbone area) then the ABRs (such as routers Y and Z) must be connected via a virtual link. The two ABRs form a point-to-point-like adjacency across the transit area (see area 0.0.0.4).

Figure 5: Backbone area



3.1.1.2 Stub area

A stub area is a designated area that does not allow external route advertisements. Routers in a stub area do not maintain external routes. A single default route to an ABR replaces all external routes. This OSPF implementation supports the optional summary route (type-3) advertisement suppression from other areas into a stub area. This feature further reduces topological database sizes and OSPF protocol traffic, memory usage, and CPU route calculation time.

In Figure 5: Backbone area, areas 0.0.0.1, 0.0.0.2 and 0.0.0.5 could be configured as stub areas. A stub area cannot be designated as the transit area of a virtual link and a stub area cannot contain an AS boundary router. An AS boundary router exchanges routing information with routers in other ASs.

3.1.1.3 Not-so-stubby area

Another OSPF area type is called a Not-So-Stubby area (NSSA). NSSAs are similar to stub areas in that no external routes are imported into the area from other OSPF areas. External routes learned by OSPF routers in the NSSA area are advertised as type-7 LSAs within the NSSA area and are translated by ABRs into type-5 external route advertisements for distribution into other areas of the OSPF domain. An NSSA area cannot be designated as the transit area of a virtual link.

In Figure 5: Backbone area, area 0.0.0.3 could be configured as a NSSA area.

3.1.1.3.1 OSPF super backbone

The 77x0 PE routers have implemented a version of the BGP/OSPF interaction procedures as defined in RFC 4577, OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs). Features included in this RFC includes:

- loop prevention
- handling LSAs received from the CE
- sham links
- managing VPN-IPv4 routes received by BGP

VPRN routes can be distributed among the PE routers by BGP. If the PE uses OSPF to distribute routes to the CE router, the standard procedures governing BGP/OSPF interactions causes routes from one site to be delivered to another in type 5 LSAs, as AS-external routes.

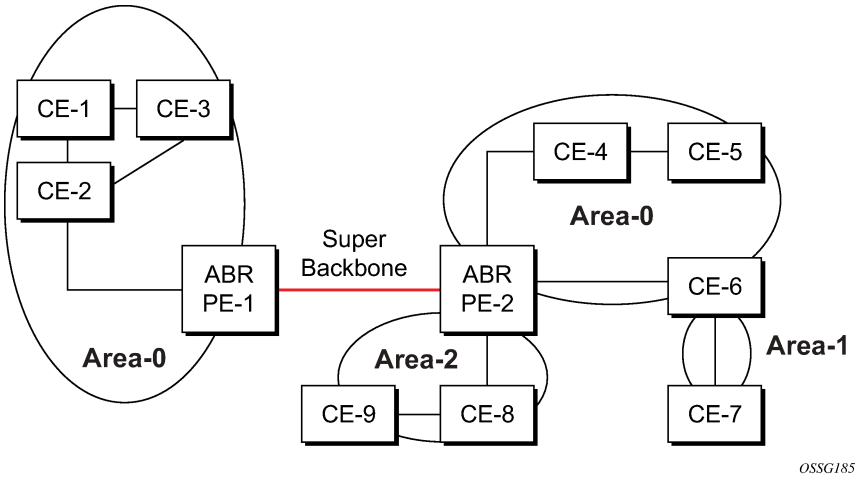
The MPLS VPN super backbone behaves like an additional layer of hierarchy in OSPF. The PE-routers that connect the respective OSPF areas to the super backbone function as OSPF Area Border Routers (ABR) in the OSPF areas to which they are attached. To achieve full compatibility, they can also behave as AS Boundary Routers (ASBR) in non-stub areas.

The PE-routers insert inter-area routes from other areas into the area where the CE-router is present. The CE-routers are not involved at any level, nor are they aware of the super backbone or of other OSPF areas present beyond the MPLS VPN super backbone.

The CE always assumes the PE is an ABR:

- If the CE is in the backbone, then the CE router assumes that the PE is an ABR linking one or more areas to the backbone.
- If the CE in not in the backbone, then the CE believes that the backbone is on the other side of the PE.
- As such, the super backbone looks like another area to the CE.

Figure 6: PEs connected to an MPLS-VPN super backbone



In Figure 6: PEs connected to an MPLS-VPN super backbone, the PEs are connected to the MPLS-VPN super backbone. To be able to distinguish if two OSPF instances are in fact the same and require Type

3 LSAs to be generated, or are two separate routing instances where type 5 external LSAs need to be generated, the concept of a domain-id is introduced.

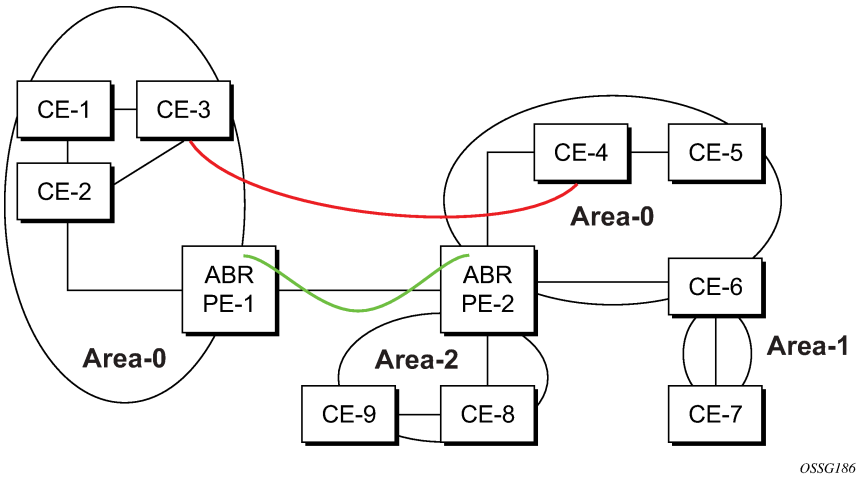
The domain ID is carried with the MP-BGP update and indicates the source OSPF Domain. When the routes are being redistributed into the same OSPF Domain, the concepts of super backbone described above apply and Type 3 LSAs are generated. If the OSPF domain does not match, then the route type is external.

Configuring the super backbone (not the sham links) makes all destinations learned by PEs with matching domain IDs inter-area routes.

When configuring sham links, these links become intra-area routes if they are present in the same area.

3.1.1.3.2 Sham links

Figure 7: Sham links



In Figure 7: Sham links, the red link between CE-3 and CE-4 could be a low speed OC-3/STM-1 link, but because it establishes an intra-area route connection between the CE-3 and CE-4, the potentially high-speed PE-1 to PE-2 connection is not used. Even with a super backbone configuration, it is regarded as an inter-area connection.

The establishment of the (green) sham-link is also constructed as an intra-area link between PE routers, a normal OSPF adjacency is formed and the link-state database is exchanged across the MPLS-VPRN. As a result, the needed intra-area connectivity is created, at this time the cost of the green and red links can be managed such that the red link becomes a standby link only in case the VPN fails.

Because the sham-link forms an adjacency over the MPLS-VPRN backbone network, when protocol protection is enabled, you must explicitly allow the OSPF packets to be received over the backbone network using the following command option.

```
configure system security cpu-protection protocol-protection allow-sham-links
```

3.1.1.3.3 Implementing the OSPF super backbone

With the OSPF super backbone architecture, the continuity of OSPF routing is preserved.

- The OSPF intra-area LSAs (type-1 and type-2) advertised bye the CE are inserted into the MPLS-VPRN super backbone by redistributing the OSPF route into MP-BGP by the PE adjacent to the CE.
- The MP-BGP route is propagated to other PE-routers and inserted as an OSPF route into other OSPF areas. Considering the PEs across the super backbone always act as ABRs they generate inter area route OSPF summary LSAs, Type 3.
- The inter-area route can now be propagated into other OSPF areas by other customer owned ABRs within the customer site.
- Customer Area 0 (backbone) routes when carried across the MPLS-VPRN using MPBGP appear as Type 3 LSAs even if the customer area remains area 0 (backbone).

A BGP extended community (OSPF domain ID) provides the source domain of the route. This domain ID is not carried by OSPF but carried by MP-BGP as an extended community attribute.

If the configured extended community value matches the receiving OSPF domain, then the OSPF super backbone is implemented.

From a BGP perspective, the cost is copied into the MED attribute.

3.1.1.3.4 Loop avoidance

If a route sent from a PE router to a CE router could then be received by another PE router from one of its own CE routers then it is possible for routing loops to occur. RFC 4577 specifies several methods of loop avoidance.

3.1.1.3.5 DN-BIT

When a Type 3 LSA is sent from a PE router to a CE router, the DN bit in the LSA options field is set. This is used to ensure that if any CE router sends this Type 3 LSA to a PE router, the PE router does not redistribute it further.

When a PE router needs to distribute to a CE router a route that comes from a site outside the latter's OSPF domain, the PE router presents itself as an ASBR (Autonomous System Border Router), and distributes the route in a type 5 LSA. The DN bit must be set in these LSAs to ensure that they are ignored by any other PE routers that receive them.

DN-BIT loop avoidance is also supported.

3.1.1.3.6 Route tag

If a particular VRF in a PE is associated with an instance of OSPF, then by default it is configured with a special OSPF route tag value called the VPN route tag. This route tag is included in the Type 5 LSAs that the PE originates and sends to any of the attached CEs. The configuration and inclusion of the VPN Route Tag is required for backward compatibility with deployed implementations that do not set the DN bit in Type 5 LSAs.

3.1.1.3.7 Sham links

A sham link is only required if a backdoor link (shown as the red link in Figure 7: Sham links) is present, otherwise configuring an OSPF super backbone will probably suffice.

3.1.2 OSPFv3 authentication

OSPFv3 authentication requires IPv6 IPsec and supports the following:

- IPsec transport mode
- AH and ESP
- manual keyed IPsec Security Association (SA)
- authentication Algorithms MD5 and SHA1

To pass OSPFv3 authentication, OSPFv3 peers must have matching inbound and outbound SAs configured using the same SA parameters (SPI, keys, and so on). The implementation must allow the use of one SA for both inbound and outbound directions.

This feature is supported on IES and VPRN interfaces as well as on virtual links.

The re-keying procedure defined in RFC 4552,*Authentication/Confidentiality for OSPFv3*, supports the following.

- For every router on the link, create an additional inbound SA for the interface being re-keyed using a new SPI and the new key.
- For every router on the link, replace the original outbound SA with one using the new SPI and key values. The SA replacement operation should be atomic with respect to sending OSPFv3 packet on the link so that no OSPFv3 packets are sent without authentication or encryption.
- For every router on the link, remove the original inbound SA.

The key rollover procedure automatically starts when the operator changes the configuration of the inbound static-sa or bidirectional static-sa under an interface or virtual link. Within the KeyRolloverInterval time period, OSPF3 accepts packets with both the previous inbound static-sa and the new inbound static-sa, and the previous outbound static-sa should continue to be used. When the timer expires, OSPF3 only accepts packets with the new inbound static-sa and for outgoing OSPF3 packets, the new outbound static-sa is used instead.

3.1.3 OSPF graceful restart helper

Both OSPFv2 and OSPFv3 support the graceful restart helper function which provides an OSPF neighbor a grace period during a control plane restart to minimize service disruption. When the control plane of a GR-capable router fails or restarts, the neighboring routers supporting the graceful restart helper mode (GR helpers) temporarily preserve OSPF forwarding information. Traffic continues to be forwarded to the restarting router using the last known forwarding tables. If the control plane of the restarting router comes back up within the grace period, the restarting router resumes normal OSPF operation. If the grace period expires, then the restarting router is presumed to be inactive and the OSPF topology is recalculated to route traffic around the failure.

3.1.3.1 BFD interaction with graceful restart

If the SR OS router is providing a grace period to an adjacent neighbor and the BFD session associated with that neighbor fails, the behavior is determined by the C-bit values sent by each neighbor as follows.

- If both BFD endpoints have set their C-bit value, then the graceful restart helper mode is canceled and any routes from that neighbor that are marked as stale are removed from the forwarding table.

- If either of the BFD endpoints has not set their C-bit value, then the graceful restart helper mode continues.

3.1.3.2 OSPFv3 graceful restart helper

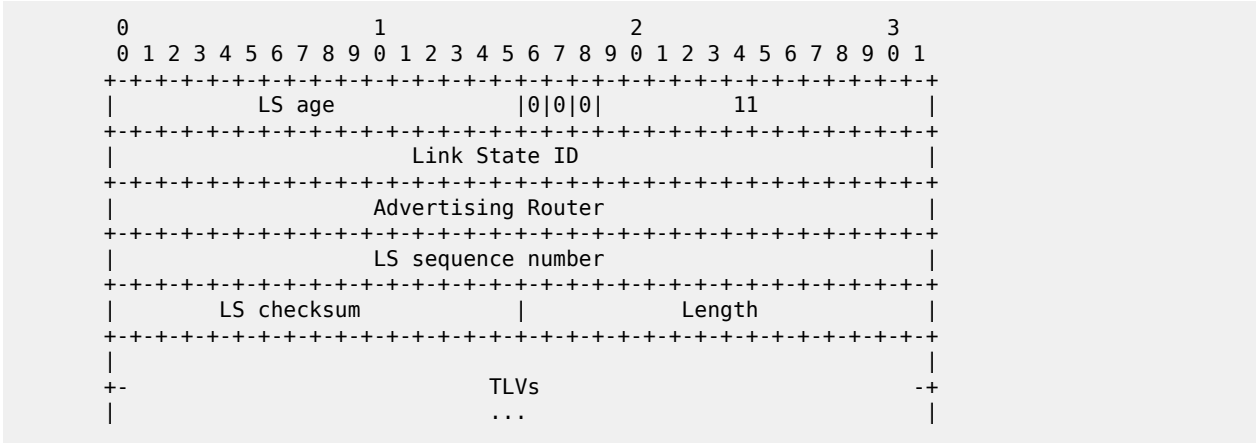
This feature extends the graceful restart helper function supported under other protocols to OSPFv3.

The primary difference between graceful restart helper for OSPFv2 and OSPFv3 is in OSPFv3 a different grace-LSA format is used.

As SR OS platforms can support a fully non-stop routing model for control plane high availability, SR OSs have no need for graceful restart as defined by the IETF in various RFCs for each routing protocol. However, because the router does need to coexist in multivendor networks and other routers do not always support a true non-stop routing model with stateful failover between routing control planes, there is a need to support a graceful restart helper function.

Graceful restart helper mode allows SR OS-based systems to provide a grace period to other routers which have requested it, during which the SR OS systems continue to use routes authored by or transiting the router requesting the grace period. This is typically used when another router is rebooting the control plane but the forwarding plane is expected to continue to forward traffic based on the previously available FIB.

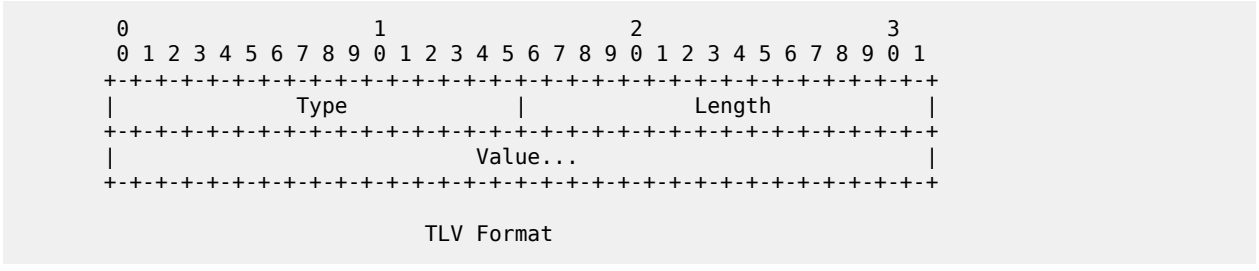
The format of the graceful OSPF restart (GRACE) LSA format is:



For more information, see section 2.2 of RFC 5187, *OSPFv3 Graceful Restart*.

The Link State ID of a grace-LSA in OSPFv3 is the Interface ID of the interface originating the LSA.

The format of each TLV is:



TLV Format

Grace-LSA TLVs are formatted according to section 2.3.2 of RFC 3630, *Traffic Engineering (TE) Extensions to OSPF Version 2*. The Grace-LSA TLVs are used to carry the Grace period (type 1) and the reason the router initiated the graceful restart process (type 2).

Other information in RFC 5187 is directed to routers that require the full graceful restart mechanism as they do not support a stateful transition from primary or backup control plane module (CPM).

3.1.4 Virtual links

The backbone area in an OSPF AS must be contiguous and all other areas must be connected to the backbone area. Sometimes, this is not possible. You can use virtual links to connect to the backbone through a non-backbone area.

Figure 5: Backbone area depicts routers Y and Z as the start and end points of the virtual link while area 0.0.0.4 is the transit area. The router must be an ABR to configure virtual links. Virtual links are identified by the router ID of the other endpoint, another ABR. These two endpoint routers must be attached to a common area, called the transit area. The area through which you configure the virtual link must have full routing information.

Transit areas pass traffic from an area adjacent to the backbone or to another area. The traffic does not originate in, nor is it destined for, the transit area. The transit area cannot be a stub area or a NSSA area.

Virtual links are part of the backbone, and behave as if they were unnumbered point-to-point networks between the two routers. A virtual link uses the intra-area routing of its transit area to forward packets. Virtual links are brought up and down through the building of the shortest-path trees for the transit area.

3.1.5 Neighbors and adjacencies

3.1.5.1 Broadcast and point-to-point networks

A router uses the OSPF Hello protocol to discover neighbors. The router sends hello packets to a multicast address and receives hello packets in return.

In broadcast networks, a designated router and a backup designated router are elected. The designated router is responsible for sending link-state advertisements (LSAs) describing the network, which reduces the amount of network traffic.

The routers attempt to form adjacencies. An adjacency is a relationship formed between a router and the designated or backup designated router. For point-to-point networks, no designated or backup designated router is elected. An adjacency must be formed with the neighbor.

To significantly improve adjacency forming and network convergence, a network should be configured as point-to-point if only two routers are connected, even if the network is a broadcast medium such as Ethernet.

When the link-state databases of two neighbors are synchronized, the routers are considered to be fully adjacent. When adjacencies are established, pairs of adjacent routers synchronize their topological databases. Not every neighboring router forms an adjacency. Routing protocol updates are only sent to and received from adjacencies. Routers that do not become fully adjacent remain in the two-way neighbor state.

3.1.5.2 Non-broadcast multi-access networks

In addition to point-to-point and broadcast networks, OSPF can operate in non-broadcast multi-access (NBMA) mode.

An NBMA segment emulates the function of a broadcast network. Every router on the segment must be configured with the IP addresses of each of its neighbors, and may need to be configured with the MAC address of its neighbor if the network does not support Layer 2 broadcast. OSPF Hello packets are transmitted individually as unicast packets to each adjacent neighbor. Because an NBMA network has no broadcast or multicast capabilities, the routing device cannot discover its neighbors dynamically, so all neighbors must be configured statically.

As in a broadcast network, a designated router and a backup designated router are elected when OSPF is operating in NBMA mode. The designated router is similarly responsible for sending link-state advertisements (LSAs) for the network.

OSPF does not support NBMA interfaces that are part of a multi-area adjacency. An interface can either be in multiple areas or in NBMA mode.

Note:

- OSPFv3 sends Hello traffic to IPv6 link-local addresses and neighbors must be statically configured using their IPv6 link-local address.
- OSPF NBMA is supported on Ethernet ports only.

3.1.6 Link-state advertisements

Link-state advertisements (LSAs) describe the state of a router or network, including router interfaces and adjacency states. Each LSA is flooded throughout an area. The collection of LSAs from all routers and networks form the protocol's topological database.

The distribution of topology database updates take place along adjacencies. A router sends LSAs to advertise its state according to the configured interval and when the router's state changes. These packets include information about the router's adjacencies, which allows detection of non-operational routers.

When a router discovers a routing table change or detects a change in the network, link state information is advertised to other routers to maintain identical routing tables. Router adjacencies are reflected in the contents of its link state advertisements. The relationship between adjacencies and the link states allow the protocol to detect non-operating routers. Link state advertisements flood the area. The flooding mechanism ensures that all routers in an area have the same topological database. The database consists of the collection of LSAs received from each router belonging to the area.

OSPF sends only the part that has changed and only when a change has taken place. From the topological database, each router constructs a tree of shortest paths with itself as root. OSPF distributes routing information between routers belonging to a single AS.

3.1.7 Metrics

In OSPF, all interfaces have a cost value or routing metric used in the OSPF link-state calculation. A metric value is configured based on hop count, bandwidth, or other parameters, to compare different paths through an AS. OSPF uses cost values to determine the best path to a particular destination: the lower the cost value, the more likely the interface will be used to forward data traffic.

Costs are also associated with externally derived routing data, such as those routes learned from the Exterior Gateway Protocol (EGP), like BGP, and is passed transparently throughout the AS. This data is kept separate from the OSPF protocol's link state data. Each external route can be tagged by the advertising router, enabling the passing of detailed information between routers on the boundaries of the AS.

3.1.8 Authentication

All OSPF protocol exchanges can be authenticated. This means that only trusted routers can participate in autonomous system routing. Nokia's implementation of OSPF supports plain text and Message Digest 5 (MD5) authentication (also called simple password).

MD5 allows an authentication key to be configured per network. Routers in the same routing domain must be configured with the same key. When the MD5 hashing algorithm is used for authentication, MD5 is used to verify data integrity by creating a 128-bit message digest from the data input. It is unique to that data. Nokia's implementation of MD5 allows the migration of an MD5 key by using a key ID for each unique key.

By default, authentication is not enabled on an interface.

3.1.9 IP subnets

OSPF enables the flexible configuration of IP subnets. Each distributed OSPF route has a destination and mask. A network mask is a 32-bit number that indicates the range of IP addresses residing on a single IP network/subnet. This specification displays network masks as hexadecimal numbers; for example, the network mask for a class C IP network is displayed as 0xffffff00. Such a mask is often displayed as 255.255.255.0.

Two different subnets with same IP network number have different masks, called variable length subnets. A packet is routed to the longest or most specific match. Host routes are considered to be subnets whose masks are all ones (0xffffffff).

3.1.10 Preconfiguration recommendations

The router ID must be available before configuring OSPF. The router ID is a 32-bit number assigned to each router running OSPF. This number uniquely identifies the router within an AS. OSPF routers use the router IDs of the neighbor routers to establish adjacencies. Neighbor IDs are learned when Hello packets are received from the neighbor.

Before configuring OSPF command options, ensure that the router ID is configured using one of the following methods. If you do not specify a router ID, then the last four bytes of the MAC address are used.

- Define the value using the following command.

  ```
  configure router router-id
  ```

- If the router ID is not specified in the **configure router router-id** context, define the system interface and specify it for the router interface.

  ```
  configure router interface
  ```

A system interface must have an IP address with a 32-bit subnet mask. The system interface is used as the router identifier by higher-level protocols such as OSPF and IS-IS. The system interface is assigned during the primary router configuration process when the interface is created in the logical IP interface context.

Note: For the BGP protocol, you can use the following command to configure a BGP router ID for use within BGP.

```
configure router bgp router-id
```

3.1.11 Multiple OSPF instances

The main route table manager (RTM) can create multiple instances of OSPF by extending the current creation of an instance. A specific interface can only be a member of a single OSPF instance. When an interface is configured in a specified domain and needs to be moved to another domain the interface must first be removed from the old instance and re-created in the new instance.

3.1.11.1 Route export policies for OSPF

When configuring route policies, you can specify the source OSPF name or instance ID using the command options in the following context:

- **MD-CLI**

```
configure policy-options policy-statement entry from protocol
```

- **classic CLI**

```
configure router policy-options policy-statement entry from protocol
```

The following apply for the instance ID:

- If an instance ID is specified, only routes installed by that instance are picked up for announcement.
- If no instance ID is specified, only routes installed by the base instance are announced.
- If the **all** command option is specified, this announces routes installed by all instances of OSPF.

When announcing internal (intra/inter-area) OSPF routes from another process, the default type should be type-1, and metric set to the route metric in RTM. For AS-external routes, by default the route type (type-1/2) should be preserved in the originated LSA, and metric set to the route metric in RTM. By default, the tag value should be preserved when an external OSPF route is announced by another process. All these can be changed with explicit action statements.

Export policy should allow match criteria based on the OSPF route hierarchy (for example, only intra-area, only inter-area, only external, only internal (intra/inter-area)). There must also be a possibility to filter based on existing tag values.

3.1.11.2 Preventing route redistribution loops

The legacy method for this was to assign a tag value to each OSPF process and mark each external route originated within that domain with that value. However, because the tag value must be preserved

throughout different OSPF domains, this only catches loops that go back to the originating domain and not where looping occurs in a remote set of domains. To prevent this type of loop, the route propagation information in the LSA must be accumulative. The following method has been implemented.

- The OSPF tag field in the AS-external LSAs is treated as a bit mask, instead of a scalar value. In other words, each bit in the tag value can be independently checked, set or reset as part of the routing policy.
- When a set of OSPF domains are provisioned in a network, each domain is assigned a specific bit value in the 32-bit tag mask. When an external route is originated by an ASBR using an internal OSPF route in a specified domain, a corresponding bit is set in the AS-external LSA. As the route gets redistributed from one domain to another, more bits are set in the tag mask, each corresponding to the OSPF domain the route visited. Route redistribution looping is prevented by checking the corresponding bit as part of the export policy. If the bit corresponding to the announcing OSPF process is already set, the route is not exported there.

  From the CLI configuration perspective, this involves adding a set of **from tag** and **action tag** commands that allow for bit operations.

3.1.12 Multi-address support for OSPFv3

While OSPFv3 was originally designed to carry only IPv6 routing information, the protocol has been extended to add support for other address families through work within the IETF (RFC 5838). These extensions within SR OS allow separate OSPFv3 instances to be used for IPv6 or IPv4 routing information.

To configure an OSPFv3 instance to distribute IPv4 routing information, a specific OSPFv3 instance must be configured using an instance ID within the range specified by the RFC. For unicast IPv4, the range is 64 to 95.

The following example shows a basic configuration for the OSPFv3 (ospf3) instance to carry IPv4 routing information. When the instance is created, the OSPFv3 instance can be configured as needed for the associated network areas, interfaces, and other protocol attributes as you would for OSPFv2.

Example: MD-CLI

```
[ex:/configure router "Base"]
A:admin@node-2# info
    ospf3 64 {
        admin-state enable
        router-id 10.20.1.3
    }
```

Example: classic CLI

```
A:node-2>config>router# info
        ospf3 64 10.20.1.3
            no shutdown
        exit
```

3.1.13 IP Fast-Reroute for OSPF and IS-IS prefixes

The IP Fast-Reroute (IP FRR) feature supports the use of the Loop-Free Alternate (LFA) backup next-hop to forward packets of IP prefixes when the primary next hop is not available. This allows a node to resume forwarding IP packets to a destination prefix without waiting for the routing convergence.

When any of the following events occurs, IGP instructs in the fast path the IOM or the forwarding engine to enable the LFA backup next hop.

- OSPF/IS-IS interface goes operationally down: physical or local admin shutdown.
- Timeout of a BFD session to a next hop when BFD is enabled on the OSPF/IS-IS interface.

IP FRR is supported on IPv4 and IPv6 OSPF/IS-IS prefixes forwarded in the base router instance to a network IP interface or to an IES SAP interface or spoke interface. It is also supported for VPRN VPN-IPv4 OSPF prefixes and VPN-IPv6 OSPF prefixes forwarded to a VPRN SAP interface or spoke interface.

IP FRR also provides a LFA backup next hop for the destination prefix of a GRE tunnel used in an SDP or in VPRN auto-bind.

The LFA next hop precomputation by IGP is described in RFC 5286 *Basic Specification for IP Fast Reroute: Loop-Free Alternates*.

3.1.13.1 IP FRR configuration

First, the user enables LFA computation by SPF under the OSPF (or IS-IS) routing protocol instance level using the following commands:

- **MD-CLI**

```
configure router ospf loopfree-alternate
```

- **classic CLI**

```
configure router ospf loopfree-alternates
```

The preceding commands instruct the IGP SPF to attempt to precompute both a primary next hop and an LFA next hop for every learned prefix. When found, the LFA next hop is populated into the RTM along with the primary next hop for the prefix.

After enabling LFA computation, the user enables IP FRR using the following commands to cause RTM to download to the IOM or the forwarding engine an LFA next hop, when found by SPF, in addition to the primary next hop for each prefix in the FIB.

- **MD-CLI**

```
configure routing-options ip-fast-reroute
```

- **classic CLI**

```
configure router ip-fast-reroute
```

3.1.13.1.1 Reducing the scope of the LFA SPF computation

To reduce the LFA SPF calculation where it is not needed, use the following command to instruct IGP to not include all interfaces participating in a specific OSPF area (or IS-IS level) in the SPF LFA computation.

```
configure router ospf area loopfree-alternate-exclude
```

You can exclude a specific IP interface from the LFA SPF computation by OSPF (or IS-IS) using the following command.

- **MD-CLI**

```
configure router ospf area interface loopfree-alternate exclude
```

- **classic CLI**

```
configure router ospf area interface loopfree-alternate-exclude
```

When an interface is excluded from the LFA SPF in IS-IS, it is excluded in both level 1 and level 2. When an interface is excluded from the LFA SPF in OSPF, it is excluded in all areas. However, the preceding OSPF command can only be executed under the area in which the specified interface is primary; when enabled, the interface is excluded in that area and in all other areas where the interface is secondary. If the user attempts to apply it to an area where the interface is secondary, the command fails.

You can also use the following commands to configure the loopfree-alternate exclude option for an OSPF instance within a VPRN service:

- **MD-CLI**

```
configure service vprn ospf area loopfree-alternate-exclude
configure service vprn ospf area interface loopfree-alternate exclude
```

- **classic CLI**

```
configure service vprn ospf area loopfree-alternate-exclude
configure service vprn ospf area interface loopfree-alternate-exclude
```

3.1.13.2 ECMP considerations

When the SPF calculates more than one primary next hop for a prefix, it does not program an LFA next hop. IP prefixes resolve to the multiple primary next hops, providing the required protection.

3.1.13.3 IP FRR and RSVP shortcut

When both IP FRR and RSVP shortcut (IGP shortcut) and LFA are enabled in IS-IS or OSPF, and IP FRR is also enabled, the following additional IP FRR are supported.

- A prefix that is resolved to a direct primary next hop can be backed up by a tunneled LFA next hop.
- A prefix that is resolved to a tunneled primary next hop does not have an LFA next hop. It relies on RSVP FRR for protection.

The LFA SPF is extended to use IGP shortcuts as LFA next hops as described in OSPF and IS-IS support for LFA calculation.

3.1.13.4 IP FRR and BGP next-hop resolution

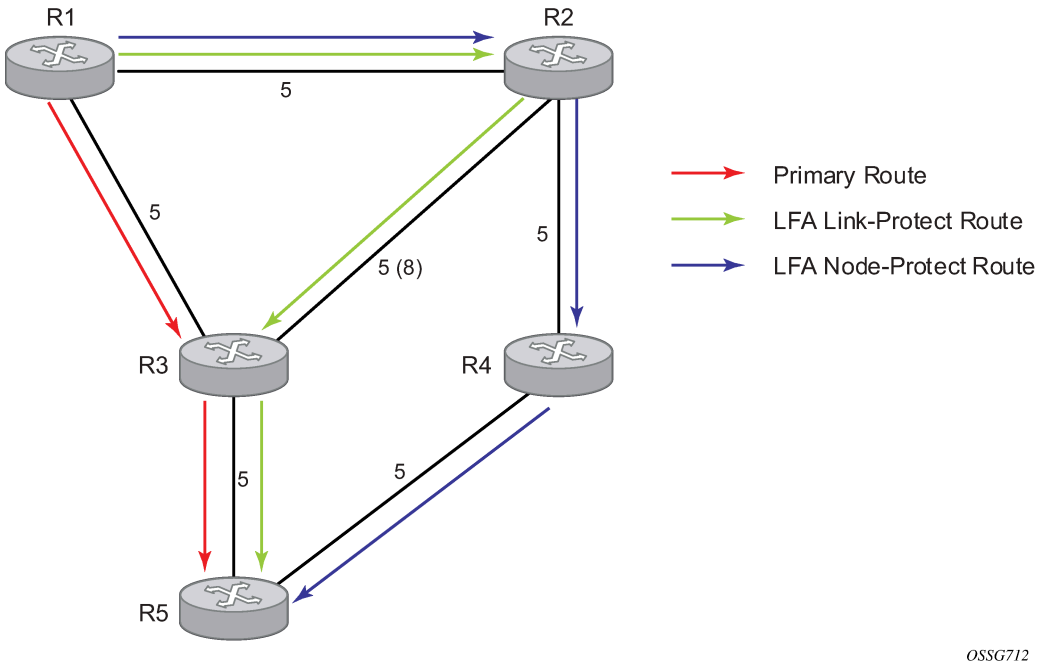
The LFA backup next-hop can protect the primary next-hop to reach a prefix advertised by a BGP neighbor. The BGP next-hop remains up when the FIB switches from the primary IGP next-hop to the LFA IGP next-hop.

3.1.13.5 OSPF and IS-IS support for LFA calculation

SPF computation in IS-IS and OSPF is enhanced to compute LFA alternate routes for each learned prefix and populate it in RTM.

The following figure shows a simple network topology with point-to-point (P2P) interfaces and highlights three routes to reach router R5 from router R1.

Figure 8: Example topology with primary and LFA routes

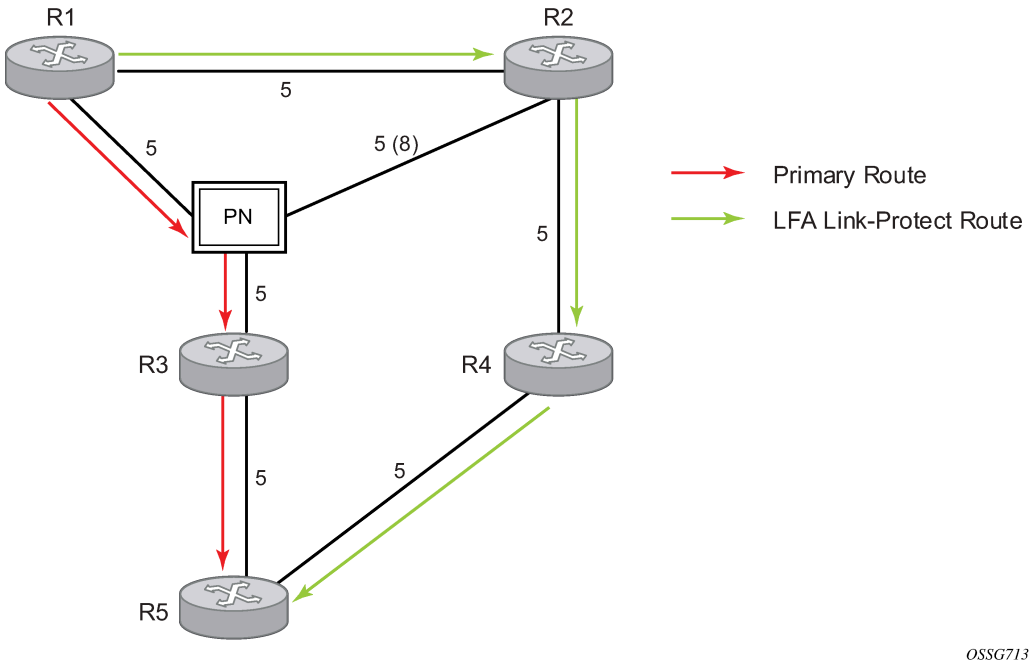


The primary route is via R3. The LFA route via R2 has two equal cost paths to reach R5. The path by way of R3 protects against failure of link R1-R3. R1 computes this route by checking that the cost for R2 to reach R5 by way of R3 is lower than the cost by way of routes R1 and R3. This condition is referred to as the Loop-free Criterion (LFC).

The path by way of R2 and R4 can be used to protect against the failure of router R3. However, with the link R2-R3 metric set to 5, R2 sees the same cost to forward a packet to R5 by way of R3 and R4. Therefore, R1 cannot guarantee that enabling the LFA next-hop R2 protects against R3 node failure. This means that the LFA next-hop R2 provides link protection only for prefix R5. If the metric of link R2-R3 is changed to 8, the LFA next-hop R2 provides node protection since a packet to R5 always goes over R4. That is, it is required that R2 become loop-free with respect to both the source node R1 and the protected node R3.

The following figure shows an example where the primary next hop uses a broadcast interface.

Figure 9: Example topology with broadcast interfaces



In order for next-hop R2 to be a link-protect LFA for route R5 from R1, it must be loop-free with respect to the R1-R3 link Pseudo-Node (PN). However, because R2 also has a link to that PN, its cost to reach R5 by way of the PN or router R4 is the same. Therefore, R1 cannot guarantee that enabling the LFA next-hop R2 protects against a failure impacting link R1-PN, because this may cause the entire subnet represented by the PN to go down. If the metric of link R2-PN is changed to 8, R2 next-hop is an LFA providing link protection.

The following are the detailed equations for this criterion as described in RFC 5286, *Basic Specification for IP Fast Reroute: Loop-Free Alternates*:

- **Rule 1**

  Link-protect LFA backup next-hop (primary next hop R1-R3 is a P2P interface):

  Distance_opt(R2, R5) < Distance_opt(R2, R1) + Distance_opt(R1, R5)

  Distance_opt(R2, R5) >= Distance_opt(R2, R3) + Distance_opt(R3, R5)

- **Rule 2**

  Node-protect LFA backup next-hop (primary next-hop R1-R3 is a P2P interface):

  Distance_opt(R2, R5) < Distance_opt(R2, R1) + Distance_opt(R1, R5)

  Distance_opt(R2, R5) < Distance_opt(R2, R3) + Distance_opt(R3, R5)

- **Rule 3**

  Link-protect LFA backup next-hop (primary next-hop R1-R3 is a broadcast interface):

  Distance_opt(R2, R5) < Distance_opt(R2, R1) + Distance_opt(R1, R5)

  Distance_opt(R2, R5) < Distance_opt(R2, PN) + Distance_opt(PN, R5), where PN stands for the R1-R3 link PN.

For a P2P interface, if SPF finds multiple LFA next-hops for a specified primary next-hop, it uses the following selection algorithm.

1. SPF picks the node-protect type in favor of the link-protect type.
2. If there is more than one LFA next-hop within the selected type, SPF picks one based on the least cost.
3. If more than one LFA next-hop with the same cost results from step 2, SPF selects the first one. This is not a deterministic selection and varies following each SPF calculation.

For a broadcast interface, a node-protect LFA is not necessarily a link-protect LFA if the path to the LFA next-hop goes over the same PN as the primary next-hop. Similarly, a link-protect LFA may not guarantee link protection if it goes over the same PN as the primary next-hop. The selection algorithm when SPF finds multiple LFA next-hops for a specified primary next-hop is modified as follows.

1. The algorithm splits the LFA next-hops into two sets.
   - The first set consists of LFA next-hops that do not go over the PN used by the primary next-hop.
   - The second set consists of LFA next-hops that go over the PN used by the primary next-hop.
2. If there is more than one LFA next-hop in the first set, SPF picks the node-protect type in favor of the link-protect type.
3. If there is more than one LFA next-hop within the selected type, SPF picks one based on the least cost.
4. If more than one LFA next-hop with equal cost results from step 3, SPF selects the first one from the remaining set. This is not a deterministic selection and varies following each SPF calculation.
5. If no LFA next-hop results from step D, SPF reruns steps 2 through 4 using the second set.

This algorithm is more flexible than strictly applying the preceding Rule 3; that is, the link-protect rule in the presence of a PN and specified in RFC 5286. A node-protect LFA that does not avoid the PN, that is, does not guarantee link protection, can still be selected as a last resort. Similarly, a link-protect LFA that does not avoid the PN can still be selected as a last resort.

Both the computed primary next-hop and LFA next-hop for a specified prefix are programmed into RTM.

3.1.13.5.1 Loop-free alternate calculation in the presence of IGP shortcuts

To expand the coverage of the LFA backup protection in a network, RSVP LSP based IGP shortcuts can be placed selectively in parts of the network and be used as an LFA backup next hop.

When IGP shortcut is enabled in IS-IS or OSPF on a specified node, all RSVP LSP originating on this node and with a destination address matching the router-id of any other node in the network are included in the main SPF by default.

To limit the time it takes to compute the LFA SPF, explicitly enable the use of an IGP shortcut as LFA backup next hop using one of the command options for the following command.

```
configure router mpls igp-shortcut
```

The following are LFA options for the IGP shortcut:

- The **lfa-protect** command option allows an LSP to be included in both the main SPF and the LFA SPFs. For a specified prefix, the LSP can be used either as a primary next hop or as an LFA next hop but not both. If the main SPF computation selected a tunneled primary next hop for a prefix, the LFA SPF does not select an LFA next hop for this prefix and the protection of this prefix relies on the RSVP LSP FRR protection. If the main SPF computation selected a direct primary next hop, the LFA SPF selects an LFA next hop for this prefix but prefers a direct LFA next hop over a tunneled LFA next hop.

- The **lfa-only** command option allows an LSP to be included in the LFA SPFs only such that the introduction of IGP shortcuts does not impact the main SPF decision. For a specified prefix, the main SPF always selects a direct primary next hop. The LFA SPF selects a an LFA next hop for this prefix but prefers a direct LFA next hop over a tunneled LFA next hop.

With the selection algorithm when SPF finds multiple LFA next hops for a specified primary next hop is modified as follows.

1. The algorithm splits the LFA next hops into two sets:
   - The first set consists of direct LFA next hops.
   - The second set consists of tunneled LFA next hops. After excluding the LSPs which use the same outgoing interface as the primary next hop.
2. The algorithms continues with first set if not empty, otherwise it continues with second set.
3. If the second set is used, the algorithm selects the tunneled LFA next hop which endpoint corresponds to the node advertising the prefix.
   - If more than one tunneled next hop exists, it selects the one with the lowest LSP metric.
   - If still more than one tunneled next hop exists, it selects the one with the lowest tunnel-id.
   - If none is available, it continues with rest of the tunneled LFAs in second set.
4. Within the selected set, the algorithm splits the LFA next hops into two sets:
   - The first set consists of LFA next hops which do not go over the PN used by primary next hop.
   - The second set consists of LFA next hops which go over the PN used by the primary next hop.
5. If there is more than one LFA next hop in the selected set, it picks the node-protect type in favor of the link-protect type.
6. If there is more than one LFA next hop within the selected type, it picks one based on the least total cost for the prefix. For a tunneled next hop, it means the LSP metric plus the cost of the LSP endpoint to the destination of the prefix.
7. If there is more than one LFA next hop within the selected type (ecmp-case) in the first set, it selects the first direct next hop from the remaining set. This is not a deterministic selection and varies following each SPF calculation.
8. If there is more than one LFA next hop within the selected type (ecmp-case) in the second set, it picks the tunneled next hop with the lowest cost from the endpoint of the LSP to the destination prefix. If there remains more than one, it picks the tunneled next hop with the lowest tunnel-id.

3.1.13.5.2 LFA calculation for inter-area and inter-level prefixes

When SPF resolves OSPF inter-area prefixes or IS-IS inter-level prefixes, it computes an LFA backup next hop to the same exit area or border router as used by the primary next hop.

3.1.13.6 Multihomed prefix LFA extensions in OSPF

3.1.13.6.1 Feature configuration

The multihomed prefix (MHP) LFA feature for IP FRR of OSPF routes and for SR-OSPF FRR is enabled using the commands in the following context:

- **MD CLI**

```
configure router ospf loop-free-alternate multi-homed-prefix
```

- **classic CLI**

```
configure router ospf loop-free-alternates multi-homed-prefix
```

When applied to IP prefixes, IP FRR must also be enabled using the following command, which allows the programming of the backup paths in the FIB:

- **MD CLI**

```
configure routing-options ip-fast-reroute
```

- **classic CLI**

```
configure router ip-fast-reroute
```

This feature makes use of the multihomed prefix model described in RFC 8518 to compute a backup IP next hop via an alternate ABR or ASBR for external prefixes and to an alternate router owner for local anycast prefixes.

The base LFA algorithm is applied to all intra-area and external IP prefixes (IP FRR) and SR-OSPF node SID tunnels (SR-OSPF FRR), as usual. Then the MHP LFA is applied to improve the protection coverage for external prefixes and anycast prefixes. For external /32 prefixes and intra-area local /32 prefixes with multiple owner routers (anycast prefixes), the base LFA backup path, if found, is preferred over the MHP LFA backup in the default behavior with the **preference** command set to **none**. You can force the programming of the MHP LFA backup by setting the **preference** command value to **all**. The algorithm details are described in RFC 8518 MHP LFA for OSPF.

After the IP next-hop based MHP LFA is enabled, the extensions to MHP LFA to compute an SR-TE repair tunnel for an SR-OSPF tunnel are automatically enabled when the following CLI command is configured to enable Topology-Independent Loop-Free Alternate (TI-LFA) or Remote Loop-Free Alternate (RLFA). The algorithm details are described in *7750 SR and 7950 XRS Segment Routing and PCE User Guide,* section "LFA solution across IGP area or instance boundary using SR repair tunnel for SR OSPF". The computation reuses the SID list of the primary path or the TI-LFA or RLFA backup path of the alternate ABR, ASBR, or alternate owner router.

- **MD CLI**

```
configure router ospf loopfree-alternate remote-lfa
configure router ospf loopfree-alternate ti-lfa
```

- **classic CLI**

```
configure router ospf loopfree-alternates remote-lfa
configure router ospf loopfree-alternates ti-lfa
```

TI-LFA, base LFA and RLFA (if enabled) are applied to the SR-OSPF tunnels of all intra-area and external /32 prefixes as usual. For node SID SR-OSPF tunnels of external /32 prefixes or intra-area /32

anycast prefixes, the LFA, TI LFA, or RFLA backup path is preferred over the MHP LFA backup path in the default behavior with the **preference** command set to a value of **none**. The user can force the programming of the MHP LFA backup by setting the **preference** command value to **all**.

The MHP backup path protects SR-OSPF tunnels in both algorithm 0 and flexible-algorithm numbers. It also extends the protection to any SR-TE LSP or SR policy that uses an SR-OSPF SID of those same prefixes in its configured or computed SID list.

3.1.13.6.2 Feature applicability

The **multi-homed-prefix** command enables the feature but its applicability depends on the LFA flavor enabled in the OSPF instance. The following scenarios are possible.

- Enable multihomed prefix and loop-free alternate.

  The IP next-hop based MHP LFA feature enhances base LFA only; it applies to IP FRR (when **ip-fast-reroute** is also enabled) and to SR-OSPF tunnels.

- Enable multihomed prefix and loop-free alternate with remote LFA or TI LFA or both.

  The enabling of RLFA, TI-LFA, or both on top of the MHP LFA automatically enables the SR OS specific extensions to the IP next-hop backup algorithm in RFC 8518. This enhancement improves coverage because it computes SR-TE backup repair tunnel to an alternate ASBR as a means to force the packet to go to the alternate ASBR because the RFC 8518 MHP LFA may not find a loop-free path to this alternate ASBR.

3.1.13.6.3 RFC 8518 MHP LFA for OSPF

This feature uses the multihomed prefix model described in RFC 8518 to compute a backup IP next hop using an alternate ABR or ASBR for external prefixes and to an alternate router owner for local anycast prefixes.

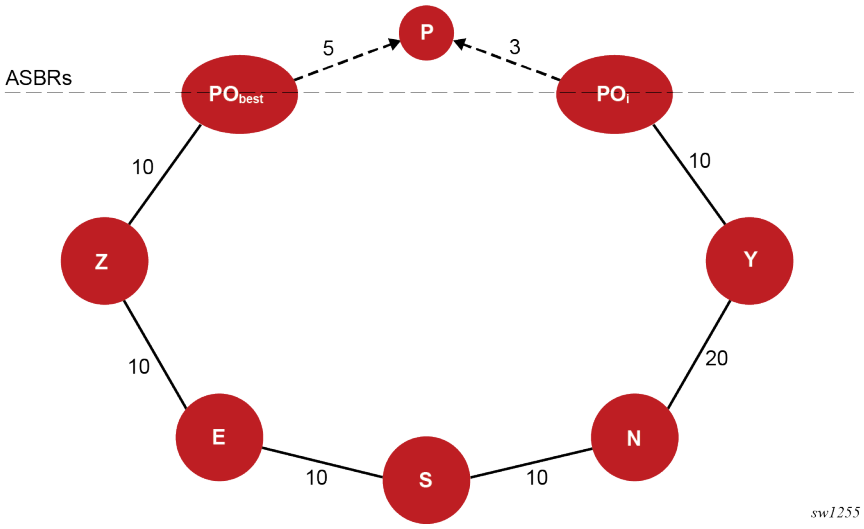
Note that the scope of the algorithm defined in RFC 8518 is limited to computed backup paths that consist of direct IP next hops and tunneled next hops (IGP shortcuts).

The SR OS implementation also extends the algorithm in RFC 8518 with computing the backup path to an alternate inter-area ASBR. The computed backup paths are added to OSPF routes of external /32 prefixes (OSPFv2 route types 3, 4, 5, and 7) and intra-area /32 anycast prefixes in the RTM if the prefixes are not protected by the base LFA or if the user has set the **preference** command value to **all**. The user must enable the **ip-fast-reroute** command to program these backup paths into the FIB in the datapath.

The computed backup path is also programmed for SR-OSPF node SID tunnels of external /32 prefixes and of local /32 anycast prefixes in both algorithm 0 and flexible-algorithm numbers. The backup path, therefore, also extends the protection to any SR-TE LSP or SR policy that uses an SR-OSPF SID of those same prefixes in its configured or computed SID list.

The following figure shows the application of an MHP LFA to IP FRR.

Figure 10: Application of MHP LFA to IP FRR



RFC 8518 creates a specific topology for each external prefix by modeling it as a multihomed node to the Points of Attachment (POi nodes). POi can be an ASBR node for an external prefix or an owner router in the case of an intra-area anycast prefix. The SR OS implementation supports prefixes redistributed by an ABR or ASBR (OSPFv2 route types 3, 5, and 7) and also extends feature support to inter-area ASBR (external routes resolved recursively to OSPFv2 route types 4).

In the topology shown in Figure 10: Application of MHP LFA to IP FRR, prefix P has a dotted link with a metric of 5 to ABR or ASBR node PO_{best} that summarizes the path in the remote OSPF area or instance to the best ABR or ASBR. Node PO_{best} is ABR or ASBR that lies in the shortest path from the computing node S to the destination prefix P.

Prefix P also has a dotted line to ABR or ASBR POi that summarizes the path to an alternate ABR or ASBR with a cost of 3. Node POi propagates prefix P to the local area or instance of computing node S because its shortest path to P is in the remote area or instance, but POi does not lie in the shortest path to P from the point of view of node S.

Node S computes and finds a MHP LFA backup path for an external prefix P using neighbor N and which uses ABR or ASBR POi to exit the local area or instance, or which uses an alternate owner router for an intra-area anycast prefix, if the following rules are satisfied.

- Protection of Link S-E (1)
  $D\_opt(N,POi) + Cost(POi ,P) < D\_opt(N,S) + D\_opt(S,PO_{best}) + Cost(PO_{best} ,P)$
- Protection of Link E (2)
  $D\_opt(N,POi) + Cost(POi ,P) < D\_opt(N,E) + D\_opt(E,PO_{best}) + Cost(PO_{best} ,P)$

Where, D_opt(X,Y) is the distance on the shortest path from node X to node Y and Cost (X,P) is the external cost to reach prefix P from advertising router X.

The MHP LFA calculation applies to the backup next hop of external OSPFv2 /32 prefixes, propagated across an area or instance boundary, and resolved in RTM when IP FRR is enabled in that OSPFv2 instance. The calculation also applies to /32 prefixes in the same area as the computing node S that are advertised by multiple routers (anycast prefixes).

OSPFv2 runs concurrently the base LFA and the MHP LFA computations.

When the alternate ASBR or ABR node POi receives the packet, it always forwards it to the adjacent area but the path to prefix P in that area may use node PO_{best}. When PO_{best} fails, node S has a non-working backup path, which blackholes packets if activated during that same time until IGP converges. That is, unless the neighbor node of PObest in the adjacent area installed a node protect LFA path to reach P.

However, if node Z computed a multihomed backup path for prefix P using an alternate ABR or ASBR POi and that path traverses PO_{best} in the adjacent area, a failure of PO_{best} immediately causes a traffic blackhole. This is because node Z has information that the backup path it activated failed after IGP converged in the adjacent area and POi updated the local area.

3.1.13.6.4 Enhancement to RFC 8518 Algorithm for backup path overlap with path to PO_{best} in the local area

The RFC 8518 inequalities in the preceding section for computing a backup path using an alternate ASBR or ABR node POi can in some topologies result in a path that may still traverse the best ASBR or ABR node PO_{best} in the local area.

The SR OS implementation enhances the node S computation of the backup path by applying the following additional inequality to detect that situation:

$D\_opt(N,POi)+ Cost(POi,P) < D\_opt(N,PO_{best})+ Cost(PO_{best},P)$

Node S prefers a path using a POi node which satisfies this inequality. If there is no such POi node, and in the case of an external prefix or an anycast SID prefix of a SR-OSPF tunnel, node S attempts to compute a SR repair tunnel following the enhancement to this feature described in *7750 SR and 7950 XRS Segment Routing and PCE User Guide,* section "LFA solution across IGP area or instance boundary using SR repair tunnel for SR OSPF".

An example topology in which an SR repair tunnel is preferred over the overlapping IP next-hop based backup path is provided in *7750 SR and 7950 XRS Segment Routing and PCE User Guide,* section "Example application of MH prefix LFA with repair tunnel".

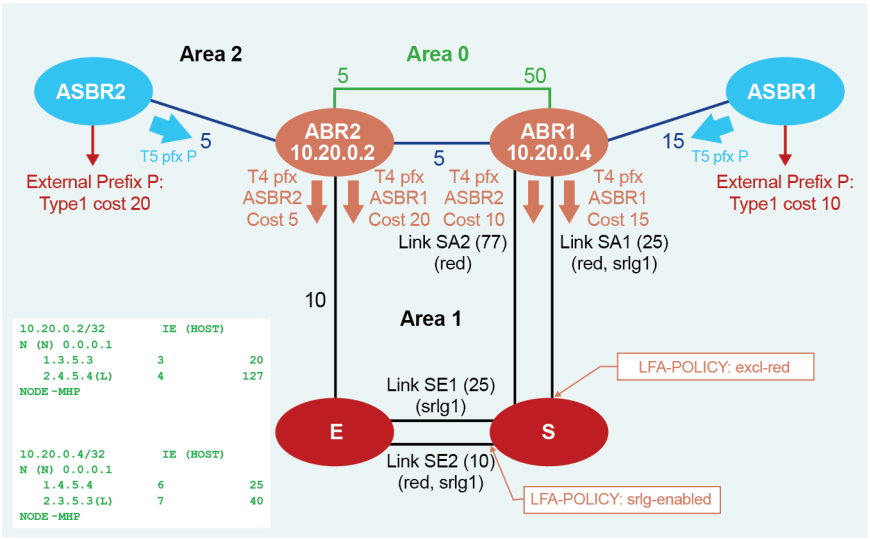
In all cases, the backup path using the POi node which does not satisfy this inequality is programmed as a last resort.

3.1.13.6.5 Interaction with LFA policy

When a LFA policy is enabled on an interface, it applies to the backup path computation of all prefixes, intra-area and external, and to each base LFA, TI-LFA, RLFA and the MHP LFA path computations.

Figure 11: Application of LFA policy to MHP calculation shows the application of LFA policy in the calculation of a MHP LFA backup path. In this example, the topology shows inter-area ASBR nodes that are advertising external prefix P.

Figure 11: Application of LFA policy to MHP calculation



Each ASBR advertises within Area 2 a route type 5 with the cost to reach prefix P, which is propagated by ABR1 and ABR2 into Area 1. This also triggers ABR1 and ABR2 to advertise into Area 1 a route type 4 for each of the prefixes of ASBR1 and ASBR2.

Node S resolves prefix P by recursively with the best path of cost 45 using ABR2 or ASBR2 and link SE2. Base LFA finds a link-protect backup path of cost 60 using ABR2 or ASBR2 and link SE1.

When an LFA policy is applied to link SE2 to exclude its SRLG in the backup path computation, the backup path using link SE2 is excluded. Furthermore, ABR1 and ABR2 do not have a link in local Area 1 and, therefore, no path exists using ABR1 to ABR2 exit router in Area 1. As a result, prefix P remains without protection.

Next, the MHP LFA in node S is enabled. Now S can use the backup path of cost 102 using alternate exit ABR1 to reach ASBR1 and link SA2. This MHP LFA backup path satisfies the SRLG constraint.

3.2 Loop-free alternate shortest path first policies

A Loop-Free Alternate Shortest Path First (LFA SPF) policy allows the user to apply specific criteria, such as admin group and SRLG constraints, to the selection of a LFA backup next hop for a subset of prefixes that resolve to a specific primary next hop. The feature introduces the concept of route next hop template to influence LFA backup next hop selection.

3.2.1 Configuring a route next hop policy template

The LFA SPF policy consists of applying a route next hop policy template to a set of prefixes.

Use the commands in the following context to create a route next hop policy template under the global router context:

- **MD-CLI**

```
configure routing-options route-next-hop-policy template
```

- **classic CLI**

```
configure router route-next-hop-policy template
```

A policy template can be used in both IS-IS and OSPF to apply the specific criteria described in the next sub-sections to prefixes protected by LFA. Each instance of IS-IS or OSPF can apply the same policy template to one or more prefix lists and to one or more interfaces.

- To create a template, enter the name of the new template directly under **route-next-hop-policy** context.
- To delete a template that is not in use, disable the template using the following command:
  - **MD-CLI**

```
route-next-hop-policy delete
```

  - **classic CLI**

```
no route-next-hop-policy
```

For the classic CLI, the commands within the route next hop policy use the **begin-commit-abort** model introduced with BFD templates. Enter the editing mode to configure or modify a policy by executing the **begin** command under **route-next-hop-policy** context. You can edit and change any number of route next hop policy templates. However, the parameter value can still be stored temporarily in the template module until the **commit** is executed under the **route-next-hop-policy** context. Any temporary parameter changes are lost if the user enters the **abort** command before the commit command.

For the classic CLI, you can create or delete a template instantly when in the editing mode without the need to enter the **commit** command. Furthermore, the **abort** command if entered has no effect on the prior deletion or creation of a template.

For the classic CLI, after the **commit** command is issued, OSPF re-evaluates the templates and if there are any net changes, it schedules a new LFA SPF to re-compute the LFA next hop for the prefixes associated with these templates.

3.2.1.1 Configuring affinity or admin group constraints

Administrative groups (admin groups), also known as affinity, are used to tag IP interfaces which share a specific characteristic with the same identifier. For example, an admin group identifier could represent all links which connect to core routers, or all links which have bandwidth higher than 10G, or all links which are dedicated to a specific service.

Use the following command to configure locally the admin group name on each router.

```
configure router interface if-attribute admin-group
```

A maximum of 32 admin groups can be configured per system.

Next use the following commands to configure the admin group membership of the IP interfaces used in LFA. You can apply admin groups to IES, VPRN, or network IP interface. You can configure a maximum of five groups for each of the IES, VPRN or network IP interfaces.

```
configure router interface if-attribute admin-group
configure service ies interface if-attribute admin-group
configure service vprn interface if-attribute admin-group
```

The user can add as many admin groups as configured to a specified IP interface. The same above command can be applied multiple times.

Note: The configured **admin-group** membership is applied in all levels/areas in which the interface is participating. The same interface cannot have different memberships in different levels/areas.

You can delete specific admin groups under the interface or delete all memberships:

- **MD-CLI**
  The **delete** command with the admin-group interface under the interface deletes one or more of the admin-group memberships of the interface. It deletes all memberships if you do not specify a group name.
- **classic CLI**
  The **no** form of the **admin-group** command under the interface deletes one or more of the admin-group memberships of the interface. It deletes all memberships if no group name is specified.

Use the following commands to add the admin group constraints into the route next hop policy template and specify the preference order number for the included group option:

- **MD-CLI**

```
configure routing-options route-next-hop-policy template include-group preference pref-number
configure routing-options route-next-hop-policy template exclude-group preference
```

- **classic CLI**

```
configure router route-next-hop-policy template include-group [pref pref-number]
configure router route-next-hop-policy template exclude-group
```

Each group is entered individually. The **include-group** statement instructs the LFA SPF selection algorithm to pick up a subset of LFA next hops among the links which belong to one or more of the specified admin groups. A link which does not belong to at least one of the admin-groups is excluded. However, a link can still be selected if it belongs to one of the groups in a **include-group** statement but also belongs to other groups which are not part of any **include-group** statement in the route next hop policy.

The option to specify a preference provides a relative preference for the admin group to select. A lower preference value means that LFA SPF first attempts to select a LFA backup next hop which is a member of the corresponding admin group. If none is found, then the admin group with the next higher preference value is evaluated. If no preference is configured for a specified admin group name, then it is supposed to be the least preferred (for example, numerically the highest preference value).

When evaluating multiple **include-group** statements within the same preference, any link which belongs to one or more of the included admin groups can be selected as an LFA next hop. There is no relative preference based on how many of those included admin groups the link is a member of.

The **exclude-group** statement simply prunes all links belonging to the specified admin group before making the LFA backup next hop selection for a prefix.

If the same group name is part of both **include** and **exclude** statements, the **exclude** statement wins. It other words, the **exclude** statement can be viewed as having an implicit preference value of 0.

Note: The admin-group criterion is applied before running the LFA next hop selection algorithm.

3.2.1.2 Configuring SRLG group constraints

Shared Risk Loss Group (SRLG) is used to tag IP interfaces which share a specific fate with the same identifier. For example, an SRLG group identifier could represent all links which use separate fibers but are carried in the same fiber conduit. If the conduit is accidentally cut, all the fiber links are cut which means all IP interfaces using these fiber links fail. The user can enable the SRLG constraint to select a LFA next hop for a prefix which avoids all interfaces that share fate with the primary next.

Use the command options in the following context to configure each SRLG group locally on each router:

- **MD-CLI**

```
configure routing-options if-attribute srlg-group
```

- **classic CLI**

```
configure router if-attribute srlg-group
```

Configure a maximum of 1024 SRLGs per system.

Next use the following commands to configure the admin group membership of the IP interfaces used in LFA. You can apply SRLG groups to IES, VPRN, or network IP interfaces.

```
configure router interface if-attribute srlg-group
configure service ies interface if-attribute srlg-group
configure service vprn interface if-attribute srlg-group
```

Add as many admin groups as configured to a specified IP

Add a maximum of 64 SRLG groups to a specified IP interface. The same above command can be applied multiple times.

Note: The configured SRLG membership is applied in all levels/areas in which the interface is participating. The same interface cannot have different memberships in different levels/areas.

You can delete specific SRLG groups under the interface or delete all memberships:

- **MD-CLI**
  The **delete** command with the SRLG-group interface under the interface deletes one or more of the SRLG-group memberships of the interface. It deletes all memberships if you do not specify a group name.
- **classic CLI**
  The **no** form of the **srlg-group** command under the interface deletes one or more of the admin-group memberships of the interface. It deletes all memberships if no group name is specified.

Finally, use the following command to add the SRLG constraint into the route next hop policy template.

- **MD-CLI**

```
configure routing-options route-next-hop-policy template srlg
```

- **classic CLI**

```
configure router route-next-hop-policy template srlg-enable
```

When this command is applied to a prefix, the LFA SPF selects a LFA next hop, among the computed ones, which uses an outgoing interface that does not participate in any of the SRLGs of the outgoing interface used by the primary next hop.

Note: The SRLG and admin-group criteria are applied before running the LFA next hop selection algorithm.

3.2.1.3 Interaction of IP and MPLS admin group and SRLG

The LFA SPF policy feature generalizes the use of admin-group and SRLG to other types of interfaces. To that end, it is important that the IP admin groups and SRLGs be compatible with the ones already supported in MPLS. The following rules are implemented.

- Perform the binding of an MPLS interface to a group, that is, configure membership of an MPLS interface in a group under the following context.

```
configure router mpls interface
```

- Perform the binding of a local or remote MPLS interface to an SRLG in the SRLG database under the following context.

```
configure router mpls srlg-database
```

- Perform the binding of an IS-IS/OSPF interface to a group in the following router or services contexts, to be used by IS-IS or OSPF in route next hop policies.

```
configure router interface if-attribute
configure service vprn interface if-attribute
configure service ies interface if-attribute
```

- Only the admin groups and SRLGs bound to an MPLS interface context or the SRLG database context are advertised in TE link TLVs and sub-TLVs when the **traffic-engineering** command option is enabled in IS-IS or OSPF. IES and VPRN interfaces do not have their attributes advertised in TE TLVs.

3.2.1.4 Configuring protection type and next hop type preferences

You can select if link protection or node protection is preferred in the selection of a LFA next hop for all IP prefixes and LDP FEC prefixes to which a route next hop policy template is applied. The default in SR OS implementation is node protection. The implementation falls back to the other type if no LFA next hop of the preferred type is found.

You can also select if tunnel backup next hop or IP backup next hop is preferred. The default in SR OS implementation is to prefer IP next hop over tunnel next hop. The implementation falls back to the other type if no LFA next hop of the preferred type is found.

The following options are added into the route next hop policy template.

- **MD-CLI**

```
configure routing-options route-next-hop-policy template nh-type
configure routing-options route-next-hop-policy template protection-type
```

- **classic CLI**

```
configure router route-nh-template template nh-typ
configure router route-nh-template template protection-type
```

When the route next hop policy template is applied to an IP interface, all prefixes using this interface as a primary next hop follows the protection type and next hop type preference specified in the template.

3.2.2 Application of route next hop policy template to an interface

After you configure the route next hop policy template with the required policies, use the following commands to apply it to all OSPF (or IS-IS) prefixes whose primary next hop uses a specific interface name.

- **MD-CLI**

```
configure router ospf area interface loopfree-alternate policy-map route-nh-template
configure router ospf3 area interface loopfree-alternate policy-map route-nh-template
configure service vprn ospf area interface loopfree-alternate policy-map route-nh-template
configure service vprn ospf3 area interface loopfree-alternate policy-map route-nh-template
configure router isis interface loopfree-alternate policy-map route-nh-template
configure service vprn isis interface loopfree-alternate policy-map route-nh-template
```

- **classic CLI**

```
configure router ospf area interface lfa-policy-map route-nh-template
configure router ospf3 area interface lfa-policy-map route-nh-template
configure service vprn ospf area interface lfa-policy-map route-nh-template
configure service vprn ospf3 area interface lfa-policy-map route-nh-template
configure router isis interface lfa-policy-map route-nh-template
configure service vprn isis interface lfa-policy-map route-nh-template
```

When a route next hop policy template is applied to an interface in IS-IS, it is applied in both level 1 and level 2. When a route next hop policy template is applied to an interface in OSPF, it is applied in all areas. However, the above CLI command in an OSPF interface context can only be executed under the area in which the specified interface is primary and then applied in that area and in all other areas where the interface is secondary. If the user attempts to apply it to an area where the interface is secondary, the command fails.

If the user excluded the interface from LFA using the command **loopfree-alternate-exclude**, the LFA policy if applied to the interface has no effect.

Finally, if the user applied a route next hop policy template to a loopback interface or to the system interface, the command is not rejected but it results in no action taken.

3.2.3 Excluding interfaces and prefixes from LFA SPF

You can use the loop-free alternate **exclude** command to exclude an interface in OSPF or in an OSPF area (or in IS-IS or an IS-IS level) from the LFA SPF. You can also exclude prefixes from a prefix policy that matches on prefixes or on IS-IS tags, for the router or VPRN service. You can exclude up to five prefix policies.

-
- **MD-CLI**

```
configure router isis interface loopfree-alternate exclude
configure router ospf area interface loopfree-alternate exclude
configure router ospf loopfree-alternate exclude
configure router ospf loopfree-alternate exclude prefix-policy
configure router ospf3 area interface loopfree-alternate exclude
configure router ospf3 loopfree-alternate exclude
configure router ospf3 loopfree-alternate exclude prefix-policy
configure service vprn ospf area interface loopfree-alternate exclude
configure service vprn ospf loopfree-alternate exclude
configure service vprn ospf loopfree-alternate exclude prefix-policy
configure service vprn ospf3 area interface loopfree-alternate exclude
configure service vprn ospf3 loopfree-alternate exclude
configure service vprn ospf3 loopfree-alternate exclude prefix-policy
configure router isis loopfree-alternate exclude
configure router isis loopfree-alternate exclude prefix-policy
configure service vprn isis interface loopfree-alternate exclude
configure service vprn isis loopfree-alternate exclude
configure service vprn isis loopfree-alternate exclude prefix-policy
```

- **classic CLI**

```
configure router ospf loopfree-alternates exclude prefix-policy
configure router ospf3 loopfree-alternates exclude prefix-policy
configure router ospf loopfree-alternate-exclude
configure router ospf3 loopfree-alternate-exclude
configure router ospf area loopfree-alternate-exclude
configure router ospf3 area loopfree-alternate-exclude
configure router ospf area interface loopfree-alternate-exclude
configure router ospf3 area interface loopfree-alternate-exclude
configure service vprn ospf3 loopfree-alternates exclude prefix-policy
configure service vprn ospf loopfree-alternates exclude prefix-policy
configure service vprn ospf area interface loopfree-alternate-exclude
configure service vprn ospf3 area interface loopfree-alternate-exclude
configure service vprn ospf3 area loopfree-alternate-exclude
configure service vprn ospf area loopfree-alternate-exclude
configure service vprn isis level loopfree-alternate-exclude
configure router isis interface loopfree-alternate-exclude
configure service vprn isis interface loopfree-alternate-exclude
configure router isis level loopfree-alternate-exclude
configure router isis loopfree-alternates exclude prefix-policy
configure service vprn isis loopfree-alternates exclude prefix-policy
```

The prefix policy is configured as in existing SR OS implementation. If you enable IS-IS prefix prioritization based on tag, it also applies to SPF LFA. If a prefix is excluded from LFA, it is not included in LFA calculation regardless of its priority. However, the prefix tag can be used in the main SPF.

Note: Prefix tags are not defined for the OSPF protocol.

If an action is not explicitly specified for the prefix policy, the default action of the loop-free alternate **excluded** command, is "reject". Consequently, regardless of whether you explicitly add a statement "default-action reject" to the prefix policy, a prefix that does not match any entry in the policy is accepted into the LFA SPF.

Example: MD-CLI

```
[ex:/configure policy-options]
A:admin@node-2# info
    prefix-list "prefix-list-1" {
        prefix 62.225.16.0/24 type exact {
        }
    }
    policy-statement "prefix-policy-1" {
        entry 10 {
            from {
                prefix-list ["prefix-list-1"]
            }
            action {
                action-type accept
            }
        }
        default-action {
            action-type reject
        }
    }

[ex:/configure router "Base" ospf 0]
A:admin@node-2# info
    ...
    loopfree-alternate
        exclude {
            prefix-policy "prefix-policy-1"
        }
    }
    ...
```

Example: classic CLI

When configuring policy options, use the **begin** command to start an editing session. To abort or save the session, use the **abort** or **commit** command.

```
A:node-2>config>router>policy-options# info
----------------------------------------------
            prefix-list "prefix-list-1"
                prefix 62.225.16.0/24 exact
            exit
            policy-statement "prefix-policy-1"
                entry 10
                    from
                        prefix-list "prefix-list-1"
                    exit
                    action accept
                    exit
                exit
                default-action reject
                exit
            exit

A:node-2>config>router>ospf>lfa>exclude# info
----------------------------------------------
            ...
```

```
            loopfree-alternates
                exclude
                    prefix-policy "prefix-policy-1"
                exit
             exit
             ...
```

3.2.4 Modification to LFA next hop selection algorithm

This feature modifies the LFA next-hop selection algorithm. The SRLG and admin-group criteria are applied before running the LFA next-hop selection algorithm. In other words, links that do not include one or more of the admin-groups in the include-group statements and links which belong to admin-groups that have been explicitly excluded using an exclude-group statement, and the links which belong to the SRLGs used by the primary next hop of a prefix are first pruned.

This pruning applies only to IP next hops. Tunnel next hops can have the admin-group or SRLG constraint applied to them under MPLS. For example, if a tunnel next hop is using an outgoing interface that belongs to a specific SRLG ID, enable the following command to ensure the RSVP LSP FRR backup LSP does not use an outgoing interface with the same SRLG ID.

```
configure router mpls srlg-frr
```

A prefix that is resolved to a tunnel next hop is protected by the RSVP FRR mechanism and not by the IP FRR mechanism. Similarly, you can include or exclude admin-groups for the RSVP LSP and its FRR bypass backup LSP in MPLS context. The admin-group constraints can, however, be applied to the selection of the outgoing interface of both the LSP primary path and its FRR bypass backup path.

The following is the modified LFA selection algorithm which is applied to prefixes resolving to a primary next hop which uses a specific route next hop policy template.

- Split the LFA next hops into two sets:
  - IP or direct next hops
  - tunnel next hops after excluding the LSPs that use the same outgoing interface as the primary next hop
- Prune the IP LFA next hops that use the following links:
  - that do not include one or more of the admin-groups in an include-group statement in the route next hop policy template
  - that belong to admin-groups that have been explicitly excluded using an exclude-group statement in the route next hop policy template
  - that belong to the SRLGs used by the primary next hop of a prefix
- Continue with the set indicated in the **nh-type** value in the route next hop policy template if not empty; otherwise continue with the other set.
- Within the IP next hop set the following:
  - Prefer LFA next hops that do not go over the Pseudo-Node (PN) used by the primary next hop.
  - Within the selected subset prefer the node-protect type or the link-protect type according to the value of the **protection-type** option in the route next hop policy template.
  - Within the selected subset, select the best admin-groups according to the preference specified in the value of the include-group option in the route next hop policy template.

  - Within the selected subset, select the lowest total cost of a prefix.
  - If the total cost is the same, select the lowest router ID.
  - If the router ID is the same, select the lowest interface index.
- Within tunnel next hop set the following:
  - Select tunnel next hops with endpoints corresponding to the node owning or advertising the prefix.
    - Within selected subset, select the one with the lowest cost (lowest LSP metric).
    - If the lowest cost is the same, select the tunnel with the lowest tunnel index.
  - If none are available, continue with rest of the tunnel LFA next hop set.
  - Prefer LFA next hops that do not go over the Pseudo-Node (PN) used by the primary next hop.
  - Within selected subset prefer the node-protect type or the link-protect type according to the value of the **protection-type** in the route next hop policy template.
  - Within selected subset, select the lowest total coast of a prefix. For a tunnel next hop, it means the LSP metric plus the cost of the LSP endpoint to the destination of the prefix.
  - If the total cost is the same, select the lowest endpoint-to-destination cost.
  - If the endpoint-to-destination is the same, select the lowest router ID.
  - If the router ID is the same, select the lowest tunnel index.

3.3 SPF LSA filtering

The SR OS OSPF implementation supports a configuration option to filter outgoing OSPF LSAs on selected OSPFv2 or OSPFv3 interfaces. This feature should be used with some caution because it goes against the principle that all OSPF routers in an area should have a synchronized Link State Database (LSDB), but it can be a useful resource saving in specific hub and spoke topologies where learning routes through OSPF is only needed in one direction (for example, from spoke to hub).

Three filtering options are available (configurable per interface).

- Do not flood any LSAs out the interface. This option is suitable if the neighbor is simply-connected and has a statically configured default route with the address of this interface as next hop.
- Flood a minimum set of self-generated LSAs out the interface (for example, router-LSA, intra-area-prefix-LSA, and link-LSA and network-LSA corresponding to the connected interface); suppress all non-self-originated LSAs. This option is suitable if the neighbor is simply connected and has a statically configured default route with a loopback or system interface address as next hop.
- Flood a minimum set of self-generated LSAs (for example, router-LSA, intra-area-prefix-LSA, and link-LSA and network-LSA corresponding to the connected interface) and all self-generated type-3, type-5 and type-7 LSAs advertising a default route (0/0) out the interface; suppress all other flooded LSAs. This option is suitable if the neighbor is simply-connected and does not have a statically configured default route.

3.4 FIB prioritization

The RIB processing of specific routes can be prioritized through the use of the rib-priority command. This command allows specific routes to be prioritized through the protocol processing so that updates are propagated to the FIB as quickly as possible.

Configuring the **rib-priority** command either within the global OSPF or OSPFv3 routing context or under a specific OSPF/OSPFv3 interface context enables this feature. Under the global OSPF context, a prefix list can be specified that identifies which route prefixes should be considered high priority. If the **rib-priority high** command is configured under an OSPF interface context, all routes learned through that interface is considered high priority.

The routes that have been designated as high priority are the first routes processed and then passed to the FIB update process so that the forwarding engine can be updated. All known high priority routes should be processed before the OSPF routing protocol moves on to other standard priority routes. This feature has the most impact when a large number of routes are learned through the OSPF routing protocols.

3.5 Extended LSA support in OSPFv3

The SR OS supports extended LSA format in OSPFv3, as described in draft-ietf-ospf-ospfv3-lsa-extend, *OSPFv3 LSA Extendibility*. Before this, the SR OS used the fixed-format LSA to carry the prefix and link information as described in RFC 5340, *OSPF for IPv6*. However, the fixed-format is not extensible and for this reason it needs to use the TLV format of the extended LSA.

With the extended LSA format, the default mode of operation for OSPFv3 is referred to as sparse mode, meaning that the router advertises the fixed-format for existing LSAs and adds the TLV-based extended LSA only when needed to advertise new sub-TLVs. This mode of operation is very similar to the way OSPFv2 advertises the Segment Routing information. It sends the prefix in the original fixed-format prefix LSA and then follows with the extended prefix TLV, which is sent in an extended prefix opaque LSA containing the prefix SID sub-TLV.

Use the following command option to enable the full extended LSA mode, which causes all existing and new LSAs to use the extended LSA format.

```
configure router ospf3 extended-lsa only
```

An OSPFv3 area inherits the instance level configuration but can also be configured independently to the sparse of full extended LSA mode.

Note: For the classic CLI, shut down the OSPFv3 instance before changing the mode of operation, because the protocol must flush all LSAs and re-establish all adjacencies.

3.6 Support of multiple instances of router information LSA in OSPFv2 and OSPFv3

This feature adds the support of multiple instances of the Router Information LSA as described in RFC 7770, *Extensions to OSPF for Advertising Optional Router Capabilities*.

The original method of advertising router capabilities used options fields in LSAs and hello packets. However, this method is not extensible because of the limited size of the options field. The RFC 4970,

Extensions to OSPF for Advertising Optional Router Capabilities, defined the Router Information LSA which can carry multiple router capability TLVs. It also defined a single TLV called the Router Informational Capabilities TLV to carry all previously defined capabilities in the options field in LSAs and hello packets. The SR OS supports RFC 4970.

RFC 7770 deprecated RFC 4970 by adding the ability to send multiple instances of the Router Information LSA to circumvent the maximum LSA size of 64 kB.

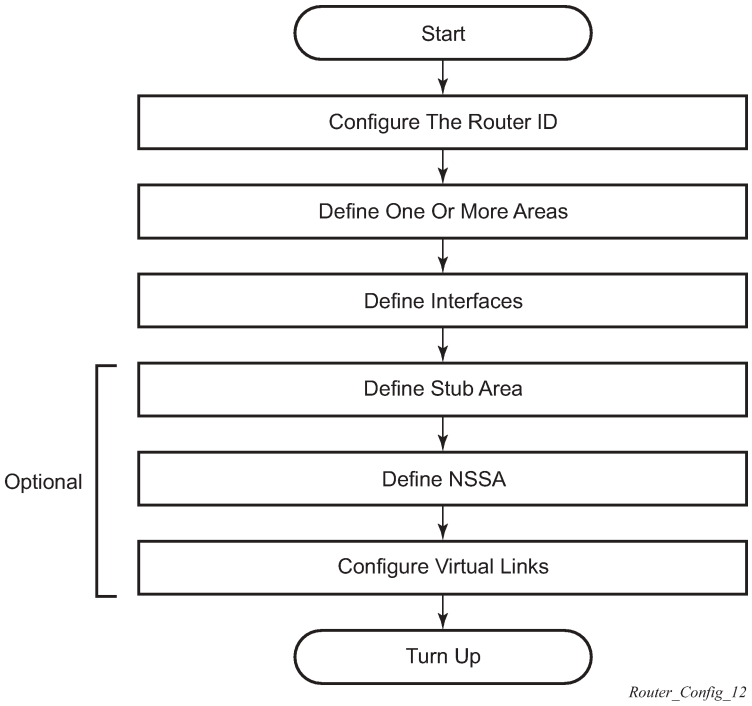
There is no CLI to enable the support of multiple instances of the Router Information LSA. The existing Router Information Capabilities TLVs is carried as the first TLV (Opaque ID 0) of the first instance (instance ID 0) of the Router Information LSA. The existing router information TLVs, such as the OSPFv2 SR-Algorithm TLV and the SID/Label Range TLV, are sent in the first instance of the Router Information LSA.

If a router information TLV is received in multiple instances of the Router Information LSA, the default behavior is to process the one in the lowest instance ID and ignore the other ones.

3.7 OSPF configuration process overview

Figure 12: OSPF configuration and implementation flow displays the process to provision basic OSPF parameters.

Figure 12: OSPF configuration and implementation flow



3.8 Configuration notes

This section describes OSPF configuration restrictions.

3.8.1 General

- Before OSPF can be configured, the router ID must be configured.
- The basic OSPF configuration includes at least one area and an associated interface.
- All default and command parameters can be modified.

3.8.1.1 OSPF defaults

The following list summarizes the OSPF configuration defaults.

- By default, a router has no configured areas.
- An OSPF instance is created in the administratively enabled state.

3.9 Configuring OSPF with CLI

This section provides information to configure Open Shortest Path First (OSPF) using the command line interface.

3.9.1 OSPF configuration guidelines

Configuration planning is essential to organize routers, backbone, non-backbone, stub, NSSA areas, and transit links. OSPF provides essential defaults for basic protocol operability. You can configure or modify commands and parameters. OSPF is not enabled by default.

The minimal OSPF parameters which should be configured to deploy OSPF are described below.

- **Router ID**

  Each router running OSPF must be configured with a unique router ID. The router ID is used by both OSPF and BGP routing protocols in the routing table manager.

  When configuring a new router ID, protocols are not automatically restarted with the new router ID. Shut down and restart the protocol to initialize the new router ID.

- **OSPF instance**

  OSPF instances must be defined when configuring multiple instances and the instance being configured is not the base instance.

- **An area**

  At least one OSPF area must be created. An interface must be assigned to each OSPF area.

- **Interfaces**

  An interface is the connection between a router and one of its attached networks. An interface has state information associated with it, which is obtained from the underlying lower level protocols and the routing protocol itself. An interface to a network has associated with it a single IP address and mask (unless the network is an unnumbered point-to-point network). An interface is sometimes also referred to as a link.

3.9.2 Basic OSPF configurations

This section provides information to configure OSPF and OSPF3 as well as configuration examples of common configuration tasks.

The minimal OSPF configuration includes the following:

- a router ID; if a router ID is not configured in the **configure router** context, the router's system interface IP address is used
- one or more areas
- interfaces (interface "system")

The following example shows a basic OSPF configuration.

Example: OSPF configuration (MD-CLI)

```
[ex:/configure router "Base" ospf 0]
A:admin@node-2# info
    area 0.0.0.0 {
        interface "system" {
        }
    }
    area 0.0.0.20 {
        nssa {
        }
        interface "to-104" {
            priority 10
        }
    }
    area 0.0.1.1 {
    }
```

Example: OSPF configuration (classic CLI)

```
A:node-2>config>router>ospf# info
----------------------------------------------
            area 0.0.0.0
                interface "system"
                exit
            exit
            area 0.0.0.20
                nssa
                exit
                interface "to-104"
                    priority 10
                exit
            exit
            area 0.0.1.1
            exit
----------------------------------------------
```

Example: OSPF3 configuration (MD-CLI)

```
[ex:/configure router "Base" ospf3 0]
A:admin@node-2# info
    export-policy ["ospf-export"]
    overload true
    timers {
        lsa-arrival 50000
```

```
    }
    asbr { }
    area 0.0.0.0 {
        interface "system" {
        }
    }
    area 0.0.0.20 {
        nssa {
        }
        interface "SR1-2" {
        }
    }
    area 0.0.0.25 {
        stub {
            default-metric 5000
        }
    }
```

Example: OSPF3 configuration (classic CLI)

```
A:node-2>config>router>ospf3# info
----------------------------------------------
            asbr
            overload
            timers
                lsa-arrival 50000
            exit
            export "OSPF-Export"
            area 0.0.0.0
                interface "system"
                exit
            exit
            area 0.0.0.20
                nssa
                exit
                interface "SR1-2"
                exit
            exit
            area 0.0.0.25
                stub
                    default-metric 5000
                exit
            exit
```

3.9.2.1 Configuring the router ID

The router ID uniquely identifies the router within an AS. In OSPF, routing information is exchanged between autonomous systems, groups of networks that share routing information. It can be set to be the same as the loopback (system interface) address. Subscriber services also use this address as far-end router identifiers when service distribution paths (SDPs) are created.

You can define the router ID as follows:

- Define the value using the following command:

```
configure router router-id
```

- If the router ID is not specified in the **configure router router-id** context, define the system interface and specify it for the router interface.

  ```
  configure router interface
  ```

  A system interface requires an IP address with a 32-bit subnet mask. The system interface is used as the router identifier by higher-level protocols such as OSPF and IS-IS. The system interface is assigned during the primary router configuration process when the interface is created in the logical IP interface context.
- inheriting the last four bytes of the MAC address
- Define the router ID when creating the OSPF or OSPF3 instance:
  - **MD-CLI**

    ```
    configure router ospf ospf-instance [router-id]
    ```

  - **classic CLI**

    ```
    configure router ospf ospf-instance [router-id]
    ```

- For the BGP protocol, you can use the following command to configure a BGP router ID for use within BGP.

  ```
  configure router bgp router-id
  ```

When configuring a new router ID, protocols are not automatically restarted with the new router ID. The next time a protocol is (re) initialized the new router ID is used. An interim period of time can occur when different protocols use different router IDs. To force the new router ID, issue the shutdown and no shutdown commands for each protocol that uses the router ID or restart the entire router.

It is possible to configure an SR OS to operate with an IPv6 only BOF and no IPv4 system interface address. When configured in this manner, the operator must explicitly define IPv4 router IDs for protocols such as OSPF and BGP as there is no mechanism to derive the router ID from an IPv6 system interface address.

The following example shows a router ID configuration.

Example: MD-CLI

```
[ex:/configure router "Base"]
A:admin@node-2# info
    autonomous-system 100
    router-id 10.10.10.104
    interface "system" {
        ipv4 {
            primary {
                address 10.10.10.104
                prefix-length 32
            }
        }
    }
    interface "to-103" {
        port 1/1/1
        ipv4 {
            primary {
                address 10.0.0.104
                prefix-length 24
```

```
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config>router# info
----------------------------------------
    ...
        interface "system"
            address 10.10.10.104/32
        exit
        interface "to-103"
            address 10.0.0.104/24
            port 1/1/1
        exit
        autonomous-system 100
        router-id 10.10.10.104
    ...
----------------------------------------
```

3.9.3 Configuring OSPF components

Use the CLI syntax displayed in the following subsections to configure OSPF components.

3.9.3.1 Configuring OSPF

The following displays a basic OSPF configuration example:

Example: MD-CLI

```
[ex:/configure router "Base" ospf 0]
A:admin@node-2# info
    export-policy ["OSPF-Export"]
    overload true
    traffic-engineering true
    overload-on-boot {
        timeout 60
    }
    asbr {
    }
```

Example: classic CLI

```
A:node-2>config>router>ospf# info
----------------------------------------------
            asbr
            overload
            overload-on-boot timeout 60
            traffic-engineering
            export "OSPF-Export"
            exit
----------------------------------------------
```

3.9.3.2 Configuring OSPF3

Use commands in the following contexts configure OSPF3 for routers or VPRN services, including export policies, external preferences, overload, router ID, timers, and so on.

```
configure router ospf3
configure service vprn ospf3
```

OSPF supports the creation of multiple OSPFv2 and OSPFv3 instances, to allow separate instances of the OSPF protocols to run independently within SR OS. To create separate instances you can specify a different optional instance ID to the **configure router ospf** and **configure router ospf3** commands. This creates unique OSPF instances for which separate link-state databases are maintained.

The following example shows an OSPF3 configuration.

Example: OSPF3 instance configuration (MD-CLI)

```
ex:/configure router "Base" ospf3 0]
A:admin@node-2# info
    export-policy ["OSPF-Export"]
    overload true
    timers {
        lsa-arrival 50000
    }
    asbr
    }
```

Example: OSPF3 instance configuration (classic CLI)

```
A:node-2>config>router>ospf3# info
----------------------------------------------
            asbr
            overload
            timers
                lsa-arrival 50000
            exit
            export "OSPF-Export"
----------------------------------------------
```

3.9.3.3 Configuring an OSPF or OSPF3 area

An OSPF area consists of routers configured with the same area ID. To include a router in a specific area, the common area ID must be assigned and an interface identified.

If your network consists of multiple areas you must also configure a backbone area (0.0.0.0) on at least one router. The backbone consists of the area border routers and other routers not included in other areas. The backbone distributes routing information between areas. The backbone is considered to be a participating area within the autonomous system. To maintain backbone connectivity, there must be at least one interface in the backbone area or have a virtual link configured to another router in the backbone area.

The minimal configuration must include an area ID and an interface. Modifying other command parameters are optional.

Use commands in the following contexts to configure an OSPF or OSPF3 area, including the area ID, area range with IP prefix and mask, blackhole-aggregate, interface, and so on.

```
configure router ospf area
```

```
configure router ospf3 area
```

The following example displays a basic OSPF area configuration.

Example: OSPF area configuration (MD-CLI)

```
[ex:/configure router "Base" ospf 0]
A:admin@node-2# info
    area 0.0.0.0 {
        interface "system" {
        }
    }
    area 0.0.0.20 {
        interface "to-104" {
            priority 10
        }
    }
    area 0.0.1.1 {
    }
```

Example: OSPF area configuration (classic CLI)

```
A:node-2>config>router>ospf# info
----------------------------------------------
        area 0.0.0.0
            interface "system"
            exit
        exit
        area 0.0.0.20
            interface "to-104"
                priority 10
                no shutdown
            exit
        exit
        area 0.0.1.1
        exit
----------------------------------------------
```

3.9.3.4 Configuring a stub area

Configure stub areas to control external advertisements flooding and to minimize the size of the topological databases on an area's routers. A stub area cannot also be configured as an NSSA.

By default, summary route advertisements are sent into stub areas. The **no** form of the summary command disables sending summary route advertisements and only the default route is advertised by the ABR. This example retains the default so the command is not entered.

If this area is configured as a transit area for a virtual link, then existing virtual links of a non-stub or NSSA area are removed when its designation is changed to NSSA or stub.

Stub areas for OSPF3 are configured the same as OSPF stub areas. Stub areas for VPRN services can also be configured the same as for OSPF stub areas. Use the commands in the following contexts to configure stub areas.

```
configure router ospf area stub
configure router ospf3 area stub
configure service vprn ospf area stub
configure service vprn ospf3 area stub
```

Example: OSPF area stub configuration (MD-CLI)

```
[ex:/configure router "Base" ospf 0]
A:admin@node-2# info
    ...
    }
    area 0.0.0.20 {
        stub {
            default-metric 5000
        }
    }
    ...
-----------------------------------------------
```

Example: OSPF area stub configuration (classic CLI)

```
A:node-2>config>router>ospf# info
-----------------------------------------------
    ...
            area 0.0.0.20
                stub
                    default-metric 5000
                exit
            exit
    ...
-----------------------------------------------
```

3.9.3.5 Configuring a not-so-stubby area

You must explicitly configure an area to be a Not-So-Stubby Area (NSSA) area. NSSAs are similar to stub areas in that no external routes are imported into the area from other OSPF areas. The major difference between a stub area and an NSSA is an NSSA has the capability to flood external routes it learns throughout its area and by an area border router to the entire OSPF domain. An area cannot be both a stub area and an NSSA.

If this area is configured as a transit area for a virtual link, then existing virtual links of a non-stub or NSSA area are removed when its designation is changed to NSSA or stub.

Use the commands in the following contexts to configure the NSSA, including the area range, default originating routes, external redistribution of routes into the NSSA, and sending of summary LSAs to the NSSA on an ABR.

```
configure router ospf area nssa
configure router ospf3 area nssa
configure service vprn ospf area nssa
configure service vprn ospf3 area nssa
```

Example: OSPF NSSA configuration (MD-CLI)

```
[ex:/configure router "Base" ospf 0]
A:admin@node-2# info
    export-policy ["OSPF-Export"]
    overload true
    traffic-engineering true
    overload-on-boot {
        timeout 60
    }
```

```
    asbr {
    }
    area 0.0.0.0 {
    }
    area 0.0.0.20 {
        stub {
        }
    }
    area 0.0.0.25
        nssa {
    }
```

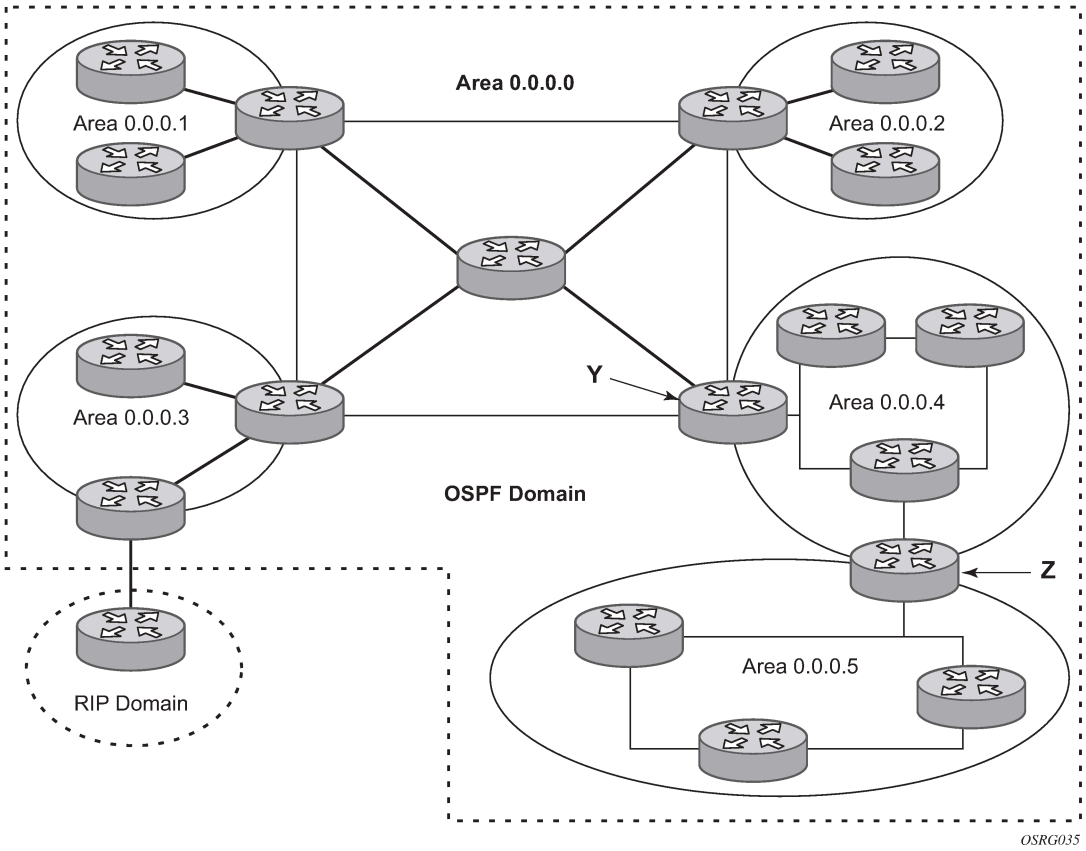
Example: OSPF NSSA configuration (classic CLI)

```
A:node-2>config>router>ospf# info
----------------------------------------------
            asbr
            overload
            overload-on-boot timeout 60
            traffic-engineering
            export "OSPF-Export"
            exit
            area 0.0.0.0
            exit
            area 0.0.0.20
                stub
                exit
            exit
            area 0.0.0.25
                nssa
                exit
            exit
----------------------------------------------
```

3.9.3.6 Configuring a virtual link

The OSPF backbone area, area 0.0.0.0, must be contiguous and all other areas must be connected to the backbone area. The backbone distributes routing information between areas. If it is not practical to connect an area to the backbone (see Area 0.0.0.5 in Figure 13: OSPF areas) then the area border routers (such as routers Y and Z) must be connected via a virtual link. The two area border routers form a point-to-point-like adjacency across the transit area (see Area 0.0.0.4). A virtual link can only be configured while in the area 0.0.0.0 context.

Figure 13: OSPF areas



The router ID specified in the **virtual-link** command must be associated with the virtual neighbor, that is, enter the virtual neighbor's router ID, not the local router ID. The transit area cannot be a stub area or an NSSA.

Use the commands in the following contexts to configure a virtual link.

```
configure router ospf area virtual-link
configure router ospf3 area virtual-link
configure service vprn ospf area virtual-link
configure service vprn ospf3 area virtual-link
```

Example: OSPF virtual-link configuration (MD-CLI)

```
[ex:/configure router "Base" ospf 0]
A:admin@node-2# info
    export-policy ["OSPF-Export"]
    overload true
    traffic-engineering true
    overload-on-boot {
        timeout 60
    }
    timers {
        lsa-arrival 50000
    }
    asbr
```

```
    }
    area 0.0.0.0 {
        virtual-link 1.2.3.4 transit-area 1.2.3.4 {
            hello-interval 9
            dead-interval 40
        }
    }
    area 0.0.0.20 {
        stub {
            default-metric 5000
        }
    }
    area 0.0.0.25 {
        nssa
    }
    area 1.2.3.4
    }
```

Example: OSPF virtual-link configuration (classic CLI)

```
A:node-2>config>router>ospf# info
----------------------------------------------
            asbr
            overload
            timers
                lsa-arrival 50000
            exit
            overload-on-boot timeout 60
            traffic-engineering
            export "OSPF-Export"
            exit
            area 0.0.0.0
                virtual-link 1.2.3.4 transit-area 1.2.3.4
                    hello-interval 9
                    dead-interval 40
                exit
            exit
            area 0.0.0.20
                stub
                   default-metric 5000
                exit
            exit
            area 0.0.0.25
                nssa
                exit
            exit
            area 1.2.3.4
            exit
----------------------------------------------
```

3.9.3.7 Configuring an interface

In OSPF, an interface can be configured to act as a connection between a router and one of its attached networks. An interface includes state information obtained from underlying lower level protocols and from the routing protocol itself. An interface to a network is associated with a single IP address and mask (unless the network is an unnumbered point-to-point network). If the address is merely changed, the OSPF configuration is preserved.

By default, only interfaces that are configured under the OSPF interface context are advertised as OSPF interfaces. The **passive** command allows an interface to be advertised as an OSPF interface without

running the OSPF protocol. When enabled, the interface ignores ingress OSPF protocol packets and does not transmit any OSPF protocol packets.

An interface can be part of more than one area, as specified in RFC 5185. This allows multiple secondary adjacencies, in addition to the primary adjacency, to be established over a single IP interface. To do this, add the keyword **secondary** when creating the interface. The keyword **secondary** can also be applied to the system interface and to loopback interfaces to allow them to participate in multiple areas, although no adjacencies are formed over these types of interfaces.

Use the commands in the following contexts to configure an OSPF interface, including authentication, hello and dead intervals, interface type, metrics, MTU, priority, retransmit intervals, and so on.

```
configure router ospf area interface
configure router ospf3 area interface
configure service vprn ospf area interface
configure service vprn ospf3 area interface
```

The following example shows some interfaces configured for OSPF.

Example: OSPF interface configuration (MD-CLI)

```
[ex:/configure router "Base" ospf 0]
A:admin@Dut-AC# info
    export-policy ["OSPF-Export"]
    overload true
    overload-on-boot {
        timeout 60
    }
    traffic-engineering true
    timers {
        lsa-arrival 50000
    }
    asbr {
    }
    area 0.0.0.0 {
        nssa {
        }
        interface "system" {
        }
        virtual-link 1.2.3.4 transit-area 1.2.3.4 {
            hello-interval 9
            dead-interval 40
        }
    }
    area 0.0.0.20 {
        stub {
            default-metric 5000
        }
        interface "to-103" {
        }
    }
    area 0.0.0.25 {
        nssa
    }
    area 1.2.3.4
    }
    area 4.3.2.1 {
        interface "SR1-3" {
        }
    }
```

Example: OSPF interface configuration (classic CLI)

```
A:node-2>config>router>ospf# info
----------------------------------------------
            asbr
            overload
            overload-on-boot timeout 60
            traffic-engineering
            timers
                lsa-arrival 50000
            exit
            export "OSPF-Export"
            exit
            area 0.0.0.0
                virtual-link 1.2.3.4 transit-area 1.2.3.4
                    hello-interval 9
                    dead-interval 40
                exit
                interface "system"
                exit
            exit
            area 0.0.0.20
                stub
                    default-metric 5000
                exit
                interface "to-103"
                exit
            exit
            area 0.0.0.25
                nssa
                exit
            exit
            area 1.2.3.4
            exit
            area 4.3.2.1
                interface "SR1-3"
                exit
            exit
----------------------------------------------
```

3.9.3.8 Configuring authentication

3.9.3.8.1 Overview

The use of protocol authentication is recommended to protect against malicious attack on the communications between routing protocol neighbors. These attacks could aim to either disrupt communications or to inject incorrect routing information into the systems routing table. The use of authentication keys can help to protect the routing protocols from these types of attacks.

Authentication must be explicitly configured and can be done so through two separate mechanisms. First is configuration of an explicit authentication key and algorithm through the use of the authentication and authentication-type commands. The second method is through the use of the authentication keychain mechanism. Both mechanisms are described in the following sections.

3.9.3.8.2 Configuring authentication keys and algorithms

The following authentication commands can be configured on the interface level or the virtual link level:

- **authentication-key**

  Configures the password used by the OSPF interface or virtual-link to send and receive OSPF protocol packets on the interface when simple password authentication is configured.

- **authentication-type**

  Enables authentication and specifies the type of authentication to be used on the OSPF interface, either password or message digest.

- **message-digest-key**

  Use this command when message-digest keyword is selected in the authentication-type command. The Message Digest 5 (MD5) hashing algorithm is used for authentication. MD5 is used to verify data integrity by creating a 128-bit message digest from the data input. It is unique to that specific data.

An special checksum is included in transmitted packets and are used by the far-end router to verify the packet by using an authentication key (a password). Routers on both ends must use the same MD5 key.

MD5 can be configured on each interface and each virtual link. If MD5 is enabled on an interface, then that interface accepts routing updates only if the MD5 authentication is accepted. Updates that are not authenticated are rejected. A router accepts only OSPF packets sent with the same key-id value defined for the interface.

When the hash parameter is not used, non-encrypted characters can be entered. After configured using the **message-digest-key** command, then all keys specified in the command are stored in encrypted format in the configuration file using the **hash** command option. When using the **hash** command option, the password must be entered in encrypted form. Hashing cannot be reversed. You must first delete the configuration and re-enter it without the **hash** command option to configure an unhashed key.

The following CLI commands are displayed to illustrate the key authentication features. These command parameters can be defined at the same time interfaces and virtual-links are being configured. See Configuring an interface and Configuring a virtual link.

Use the commands in the following contexts to configure authentication.

```
configure router ospf area interface
configure router ospf area virtual-link
configure service vprn ospf area interface
configure service vprn ospf area virtual-link
```

Example: OSPF interface authentication configuration (MD-CLI)

```
[ex:/configure router "Base" ospf 0]
A:admin@node-2# info
    export-policy ["OSPF-Export"]
    overload true
    traffic-engineering true
    overload-on-boot {
        timeout 60
    }
    asbr {
    }
    area 0.0.0.0 {
        virtual-link 1.2.3.4 transit-area 1.2.3.4 {
            hello-interval 9
            dead-interval 40
        }
```

```
        virtual-link 10.0.0.1 transit-area 0.0.0.1 {
            authentication-type message-digest
            message-digest-key 2 md5 "Mi6BQAFi3MI hash"
        }
    }
    area 0.0.0.20 {
        stub {
        interface "to-103" {
        }
    }
    area 0.0.0.25 {
        nssa {
        }
    }
    area 0.0.0.40 {
        interface "test1" {
            authentication-key "fBgGq1uSV4L9CTRfb2dINH8S72U4f8wT hash2"
            authentication-type password
        }
    }
    area 1.2.3.4 {
    }
```

Example: OSPF interface and virtual-link authentication configuration (classic CLI)

```
A:node-2>config>router>ospf# info
----------------------------------------------
            asbr
            overload
            overload-on-boot timeout 60
            traffic-engineering
            export "OSPF-Export"
            exit
            area 0.0.0.0
                virtual-link 10.0.0.1 transit-area 0.0.0.1
                    authentication-type message-digest
                    message-digest-key 2 md5 "Mi6BQAFi3MI" hash
                exit
                virtual-link 1.2.3.4 transit-area 1.2.3.4
                    hello-interval 9
                    dead-interval 40
                exit
            exit
            area 0.0.0.20
                stub
                exit
                interface "to-103"
                exit
            exit
            area 0.0.0.25
                nssa
                exit
            exit
            area 0.0.0.40
                interface "test1"
                    authentication-type password
                    authentication-key "3WErEDozxyQ" hash
                exit
            exit
            area 1.2.3.4
            exit
----------------------------------------------
```

3.9.3.8.3 Configuring authentication using keychains

The use of authentication mechanism is recommended to protect against malicious attack on the communications between routing protocol neighbors. These attacks could aim to either disrupt communications or to inject incorrect routing information into the systems routing table. The use of authentication keys can help to protect the routing protocols from these types of attacks. In addition, the use of authentication keychains provides the ability to configure authentication keys and make changes to them without affecting the state of the routing protocol adjacencies.

To configure the use of an authentication keychain within OSPF, use the following steps.

1. Configure an authentication keychain within the following context. The configured keychain must include at least on valid key entry, using a valid authentication algorithm for the OSPF protocol.
   - **MD-CLI**

     ```
     configure system security keychains
     ```

   - **classic CLI**

     ```
     configure system security keychain
     ```

2. Associate the configure authentication keychain within OSPF. Authentication keychains can be used to specify the authentication key and algorithm on a per interface basis within the configuration for the OSPF protocol.

For a key entry to be valid, it must include a valid key, the current system clock value must be within the begin and end time of the key entry, and the algorithm specified in the key entry must be supported by the OSPF protocol.

The OSPF protocol supports the following algorithms:

- clear text password
- MD5
- HMAC-SHA-1-96
- HMAC-SHA-1
- HMAC-SHA-256

The keychain error handling is described below.

- If a keychain exists but there are no active key entries with an authentication type that is valid for the associated protocol, then inbound protocol packets are not authenticated and discarded and no outbound protocol packets are sent.
- If keychain exists, but the last key entry has expired, a log entry is raised indicating that all keychain entries have expired. The OSPF protocol requires that the protocol continue to authenticate inbound and outbound traffic using the last valid authentication key.

3.9.3.9 Assigning a designated router

A designated router is elected according to the priority number advertised by the routers. When a router starts up, it checks for a current designated router. If a designated router is present, then the router accepts that designated router, regardless of its own priority designation. When a router fails, then new designated and backup routers are elected according their priority numbers.

Note: The **priority** command is only used if the interface is configured as a broadcast type. The designated router is responsible for flooding network link advertisements on a broadcast network to describe the routers attached to the network. A router uses hello packets to advertise its priority. The router with the highest priority interface becomes the designated router. A router with priority 0 is not eligible to be a designated router or a backup designated router. At least one router on each logical IP network or subnet must be eligible to be the designated router. By default, routers have a priority value of 1.

Use the **priority** command in the following contexts to configure the priority for the OSPF or OSPF3 area interface.

```
configure router ospf area interface
configure router ospf3 area interface
configure service vprn ospf area interface
configure service vprn ospf3 area interface
```

Example: Priority configuration for an OSPF area interface (MD-CLI)

```
[ex:/configure router "Base" ospf 0]
A:admin@node-2# info
    ...
    area 0.0.0.25 {
        nssa {
        }
        interface "if2" {
            priority 100
        }
    }
    ...
```

Example: Priority configuration for an OSPF area interface (classic CLI)

```
A:node-2>config>router>ospf# info
----------------------------------------------
            ...
            area 0.0.0.25
                nssa
                exit
                interface "if2"
                    priority 100
                exit
            exit
            ...
----------------------------------------------
```

3.9.3.10 Configuring route summaries

Area border routers send summary (type 3) advertisements into a stub area or NSSA to describe the routes to other areas. This command is particularly useful to reduce the size of the routing and Link State Database (LSDB) tables within the stub or NSSA.

By default, summary route advertisements are sent into the stub area or NSSA. The no form of the summaries command disables sending summary route advertisements and, in stub areas, the default route is advertised by the area border router.

Use the summaries command in the following contexts to configure route summary features. These command options can be defined at the same time when stub areas and NSSAs are configured. See Configuring a stub area and Configuring a not-so-stubby area.

Use the following commands to configure a route summary.

```
configure router ospf area stub summaries
configure router ospf3 area stub summaries
configure service vprn ospf area stub summaries
configure service vprn ospf3 area stub summaries
configure router ospf area nssa summaries
configure router ospf3 area nssa summaries
configure service vprn ospf area nssa summaries
configure service vprn ospf3 area nssa summaries
```

Example: Stub and NSSA route summary configuration for OSPF area (MD-CLI)

```
[ex:/configure router "Base" ospf 0]
A:admin@node-2# info
    export-policy ["OSPF-Export"]
    overload true
    traffic-engineering true
    overload-on-boot {
        timeout 60
    }
    asbr {
    }
    area 0.0.0.0 {
        virtual-link 1.2.3.4 transit-area 1.2.3.4 {
            hello-interval 9
            dead-interval 40
        }
        virtual-link 10.0.0.1 transit-area 0.0.0.1 {
            authentication-type message-digest
            message-digest-key 2 md5 "Mi6BQAFi3MI hash"
        }
    }
    area 0.0.0.20 {
        stub {
        interface "to-103" {
        }
    }
    area 0.0.0.25 {
        interface "if2" {
            priority 100
        }
        nssa {
        }
    }
    area 0.0.0.40 {
        interface "test1" {
            authentication-key "fBgGq1uSV4L9CTRfb2dINH8S72U4f8wT hash"
            authentication-type password
        }
    }
    area 1.2.3.4 {
    }
```

Example: Stub and NSSA route summary configuration for OSPF area (classic CLI)

```
A:node-2>config>router>ospf# info
----------------------------------------------
```

```
            asbr
            overload
            overload-on-boot timeout 60
            traffic-engineering
            export "OSPF-Export"
            exit
            area 0.0.0.0
                virtual-link 10.0.0.1 transit-area 0.0.0.1
                    authentication-type message-digest
                    message-digest-key 2 md5 "Mi6BQAFi3MI" hash
                exit
                virtual-link 1.2.3.4 transit-area 1.2.3.4
                    hello-interval 9
                    dead-interval 40
                exit
                interface "system"
                exit
            exit
            area 0.0.0.20
                stub
                exit
                interface "to-103"
                exit
            exit
            area 0.0.0.25
                nssa
                exit
                interface "if2"
                    priority 100
                exit
            exit
            area 0.0.0.40
                interface "test1"
                    authentication-type password
                    authentication-key "3WErEDozxyQ" hash
                exit
            exit
            area 1.2.3.4
            exit
-------------------------------------------
```

3.9.3.11 Configuring route preferences

A route can be learned by the router from different protocols, in which case, the costs are not comparable. When this occurs, the preference value is used to decide which route is installed in the forwarding table if several protocols calculate routes to the same destination. The route with the lowest preference value is selected.

Different protocols should not be configured with the same preference, if this occurs the tiebreaker is per the default preference table as defined in Table 2: Route preference defaults by route type . If multiple routes are learned with an identical preference using the same protocol, the lowest cost route is used.

Table 2: Route preference defaults by route type

Route type	Preference	Configurable
Direct attached	0	—
Static routes	5	Yes

Route type	Preference	Configurable
OSPF internal	10	Yes[1]
IS-IS level 1 internal	15	Yes
IS-IS level 2 internal	18	Yes
OSPF external	150	Yes
IS-IS level 1 external	160	Yes
IS-IS level 2 external	165	Yes
BGP	170	Yes

If multiple routes are learned with an identical preference using the same protocol and the costs (metrics) are equal, the decision of what route to use is determined by the configuration of the **ecmp** command in the **configure router** or **configure service vprn** context.

Use the commands in the following contexts to configure route-preference features. The command options can be defined at the same time you configure OSPF. See Configuring OSPF components.

Use the following commands to configure route preferences and external preferences for OSPF and OSPF3.

```
configure router ospf external-preference
configure router ospf preference
configure router ospf3 external-preference
configure router ospf3 preference
configure service vprn ospf external-preference
configure service vprn ospf preference
configure service vprn ospf3 external-preference
configure service vprn ospf3 preference
```

The following example displays a route preference and external-preference configuration for OSPF.

Example: Route preference and external-preference configuration for OSPF (MD-CLI)

```
[ex:/configure router "Base" ospf 0]
A:admin@node-2# info
    export-policy ["OSPF-Export"]
    overload true
    preference 9
    external-preference 140
    traffic-engineering true
    overload-on-boot {
        timeout 60
    }
    asbr {
    }
    area 0.0.0.0 {
        virtual-link 1.2.3.4 transit-area 1.2.3.4 {
            hello-interval 9
            dead-interval 40
        }
```

[1] Preference for OSPF internal routes is configured with the preference command.

```
        virtual-link 10.0.0.1 transit-area 0.0.0.1 {
            authentication-type message-digest
            message-digest-key 2 md5 "Mi6BQAFi3MI" hash
        }
    }
    area 0.0.0.20 {
        stub {
        interface "to-103" {
        }
    }
    area 0.0.0.25 {
        interface "if2" {
            priority 100
        }
        nssa {
        }
    }
    area 0.0.0.40 {
        interface "test1" {
            authentication-key "fBgGq1uSV4L9CTRfb2dINH8S72U4f8wT hash2"
            authentication-type password
        }
    }
    area 1.2.3.4 {
    }
```

Example: Route preference and external-preference configuration for OSPF (classic CLI)

```
A:node-2>config>router>ospf# info
----------------------------------------------
            asbr
            overload
            overload-on-boot timeout 60
            timers
                lsa-arrival 50000
            exit
            traffic-engineering
            preference 9
            external-preference 140
            export "OSPF-Export"
            exit
            area 0.0.0.0
                virtual-link 10.0.0.1 transit-area 0.0.0.1
                    authentication-type message-digest
                    message-digest-key 2 md5 "Mi6BQAFi3MI" hash
                exit
                virtual-link 1.2.3.4 transit-area 1.2.3.4
                    hello-interval 9
                    dead-interval 40
                exit
                interface "system"
                exit
            exit
            area 0.0.0.20
                stub
                exit
                interface "to-103"
                exit
            exit
            area 0.0.0.25
                nssa
                exit
                interface "if2"
```

```
                    priority 100
                exit
            exit
            area 0.0.0.40
                interface "test1"
                    authentication-type password
                    authentication-key "3WErEDozxyQ" hash
                exit
            exit
            area 1.2.3.4
            exit
-----------------------------------------------
```

3.10 OSPF configuration management tasks

This section discusses OSPF configuration management tasks.

3.10.1 Modifying a router ID

Because the router ID is defined in the **configure router** context, not in the OSPF configuration context, the protocol instance is not aware of the change. Re-examine the plan detailing the router ID. Changing the router ID on a device could cause configuration inconsistencies if associated values are not also modified.

After you have changed a router ID, manually shut down and restart the protocol using the shutdown and no shutdown commands in order for the changes to be incorporated.

Use the following command to change the router ID.

```
configure router router-id
```

The following example displays the results of a modification to the original router ID.

Example: Router ID configuration change (MD-CLI)

```
[ex:/configure router "Base"]
A:admin@node-2# info
    autonomous-system 100
    router-id 10.10.10.104
    interface "system" {
        ipv4 {
            primary {
                address 10.10.10.104
                prefix-length 32
            }
        }
    }
    interface "to-103" {
        port 1/1/1
        ipv4 {
            primary {
                address 10.0.0.103
                prefix-length 24
            }
        }
    }

[ex:/configure router "Base"]
```

```
A:admin@node-2# info
    router-id 10.10.10.103
    interface "system" {
        ipv4 {
            primary {
                address 10.10.10.10
                prefix-length 32
            }
        }
    }
    interface "to-104" {
        port 1/1/1
        ipv4 {
            primary {
                address 10.0.0.104
                prefix-length 24
            }
        }
```

Example: Router ID configuration change (classic CLI)

```
A:node-2>config>router# info
----------------------------------------
IP Configuration
----------------------------------------
        interface "system"
            address 10.10.10.104/32
        exit
        interface "to-103"
            address 10.0.0.103/24
            port 1/1/1
        exit
        autonomous-system 100
        router-id 10.10.10.104
----------------------------------------

A:node-2>config>router# info
----------------------------------------
IP Configuration
----------------------------------------
        interface "system"
            address 10.10.10.103/32
        exit
        interface "to-104"
            address 10.0.0.104/24
            port 1/1/1
        exit
        autonomous-system 100
        router-id 10.10.10.103
----------------------------------------
```

3.10.2 Deleting a router ID

You can modify a router ID, but you cannot delete the configured router ID. When you delete the router ID, it reverts to the default value (the system interface address, which is also the loopback address). If a system interface address is not configured, the last 32 bits of the chassis MAC address is used as the router ID.

Use the following command to delete the router ID:

- **MD-CLI**

```
configure router router-id delete
```

- **classic CLI**

```
configure router no router-id
```

3.10.3 Modifying OSPF configuration

You can change or remove the existing OSPF configuration. The changes are applied immediately.

The following example displays an OSPF modification in which an interface is removed and another interface added.

Use the following commands to remove an interface configuration:

- **MD-CLI**

```
configure router ospf area interface delete
```

- **classic CLI**

```
configure router ospf area no interface
```

Note:
If you want to disable an interface instead of deleting it, use the following commands:

- **MD-CLI**

```
configure router ospf area interface admin-state disable
```

- **classic CLI**

```
configure router ospf area interface shutdown
```

The following example displays modifications to a configured OSPF interface.

Example: Modification to OSPF interface (MD-CLI)

```
[ex:/configure router "Base"]
A:admin@node-2# info
    ...
    area 0.0.0.20 {
        stub {
        interface "to-103" {
        }
    }
    ...

[ex:/configure router "Base" area 0.0.0.20 interface "to-103"]
A:admin@node-2# delete

*[ex:/configure router "Base" area 0.0.0.20]
A:admin@node-2# interface "to-HQ"
```

```
*[ex:/configure router "Base" area 0.0.0.20  interface "to-HQ"]
A:admin@node-2# priority 50

[ex:/configure router "Base"]
A:admin@node-2# info
    ...
    area 0.0.0.20 {
        stub {
        interface "to-HQ" {
            priority 50
        }
    }
    ...
```

Example: Modification to OSPF interface (classic CLI)

```
A:node-2>config>router>ospf# info
----------------------------------------------
            ...
            area 0.0.0.20
                stub
                exit
                interface "to-103"
                exit
            exit
            ...

A:node-2>config>router>ospf>area# no interface "to-103"
*A:node-2>config>router>ospf>area# interface "to-HQ"
*A:node-2>config>router>ospf>area>interface# priority 50
----------------------------------------------

A:node-2>config>router>ospf># info
----------------------------------------------
            ...
            area 0.0.0.20
                stub
                exit
                interface "to-HQ"
                    priority 50
                exit
            exit
            ...
----------------------------------------------
```

4 IS-IS

4.1 Configuring IS-IS

Intermediate-system-to-intermediate-system (IS-IS) is a link-state interior gateway protocol (IGP) which uses the Shortest Path First (SPF) algorithm to determine routes. Routing decisions are made using the link-state information. IS-IS evaluates topology changes and, if necessary, performs SPF recalculations.

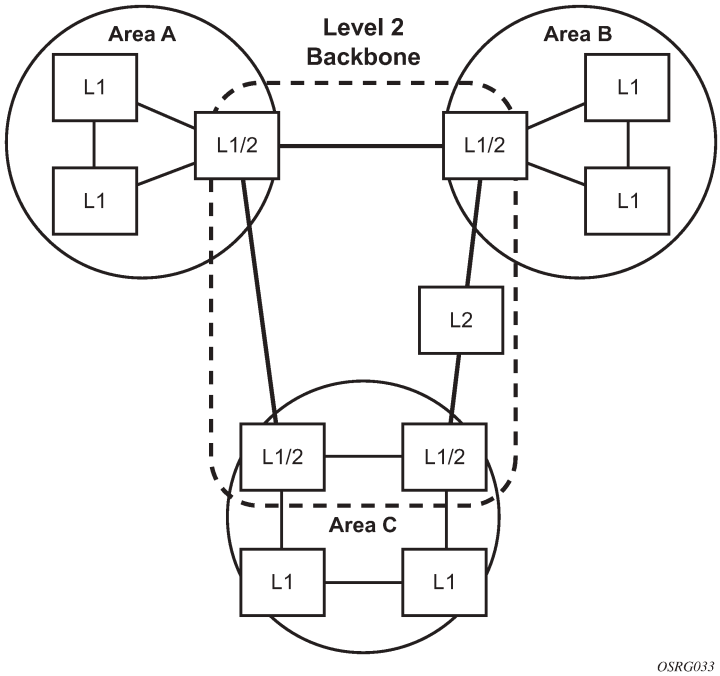
Entities within IS-IS include networks, intermediate systems, and end systems. In IS-IS, a network is an autonomous system (AS), or routing domain, with end systems and intermediate systems. A router is an intermediate system. End systems are network devices which send and receive protocol data units (PDUs), the OSI term for packets. Intermediate systems send, receive, and forward PDUs.

End system and intermediate system protocols allow routers and nodes to identify each other. IS-IS sends out link-state updates periodically throughout the network, so each router can maintain current network topology information.

IS-IS supports large ASs by using a two-level hierarchy. A large AS can be administratively divided into smaller, more manageable areas. A system logically belongs to one area. Level 1 routing is performed within an area. Level 2 routing is performed between areas. The routers can be configured as level 1, level 2, or both level 1/2.

Figure 14: IS-IS routing domain displays an example of an IS-IS routing domain.

Figure 14: IS-IS routing domain



4.1.1 Routing

OSI IS-IS routing uses two-level hierarchical routing. A routing domain can be partitioned into areas. Level 1 routers know the topology in their area, including all routers and end systems in their area but do not know the identity of routers or destinations outside of their area. Level 1 routers forward traffic with destinations outside of their area to a level 2 router in their area.

Level 2 routers know the level 2 topology, and know which addresses are reachable by each level 2 router. Level 2 routers do not need to know the topology within any level 1 area, except to the extent that a level 2 router can also be a level 1 router within a single area. By default, only level 2 routers can exchange PDUs or routing information directly with external routers located outside the routing domain.

The two types of routers in IS-IS are described below.

- **Level 1 intermediate systems**

  Routing is performed based on the area ID portion of the ISO address called the *Network Entity Title* (NET). Level 1 systems route within an area. They recognize, based on the destination address, whether the destination is within the area. If so, they route toward the destination. If not, they route to the nearest level 2 router.

- **Level 2 intermediate systems**

  Routing is performed based on the area address. They route toward other areas, regardless of other area's internal structure. A level 2 intermediate system can also be configured as a level 1 intermediate system in the same area.

The level 1 router's area address portion is manually configured (see ISO network addressing). A level 1 router does not become a neighbor with a node that does not have a common area address. However, if a level 1 router has area addresses A, B, and C, and a neighbor has area addresses B and D, then the level 1 router accepts the other node as a neighbor, as address B is common to both routers. Level 2 adjacencies are formed with other level 2 nodes whose area addresses do not overlap. If the area addresses do not overlap, the link is considered by both routers to be level 2 only and only level 2 LSPDUs flow on the link.

Within an area, level 1 routers exchange LSPs which identify the IP addresses reachable by each router. Specifically, zero or more IP address, subnet mask, and metric combinations can be included in each LSP. Each level 1 router is manually configured with the IP address, subnet mask, and metric combinations, which are reachable on each interface. A level 1 router routes as follows:

- if a specified destination address matches an IP address, subnet mask, or metric reachable within the area; the PDU is routed via level 1 routing
- if a specified destination address does not match any IP address, subnet mask, or metric combinations listed as reachable within the area; the PDU is routed toward the nearest level 2 router

Level 2 routers include in their LSPs, a complete list of IP address, subnet mask, and metrics specifying all the IP addresses which reachable in their area. This information can be obtained from a combination of the level 1 LSPs (by level 1 routers in the same area). Level 2 routers can also report external reachability information, corresponding to addresses reachable by routers in other routing domains or autonomous systems.

4.1.2 IS-IS frequently used terms

- **Area**

An area is a routing sub-domain which maintains detailed routing information about its own internal composition, and also maintains routing information which allows it to reach other routing sub-domains. Areas correspond to the level 1 sub-domain.

- **End system**

  End systems send NPDUs to other systems and receive NPDUs from other systems, but do not relay NPDUs. This International Standard does not specify any additional end system functions beyond those supplied by ISO 8473 and ISO 9542.

- **Neighbor**

  A neighbor is an adjacent system reachable by traversing a single sub-network by a PDU.

- **Adjacency**

  An adjacency is a portion of the local routing information which pertains to the reachability of a single neighboring end or intermediate system over a single circuit. Adjacencies are used as input to the decision process to form paths through the routing domain. A separate adjacency is created for each neighbor on a circuit and for each level of routing (level 1 and level 2) on a broadcast circuit.

- **Circuit**

  The subset of the local routing information base pertinent to a single local Subnetwork Point of Attachments (SNPAs).

- **Link**

  The communication path between two neighbors. A link is up when communication is possible between the two SNPAs.

- **Designated IS**

  The intermediate system on a LAN which is designated to perform additional duties. In particular, the designated IS generates link-state PDUs on behalf of the LAN, treating the LAN as a pseudonode.

- **Pseudonode**

  Where a broadcast sub-network has *n* connected intermediate systems, the broadcast sub-network itself is considered to be a pseudonode. The pseudonode has links to each of the *n* intermediate systems and each of the ISs has a single link to the pseudonode (instead of *n*-1 links to each of the other intermediate systems). Link-state PDUs are generated on behalf of the pseudonode by the designated IS.

- **Broadcast sub-network**

  A multi-access subnetwork that supports the capability of addressing a group of attached systems with a single PDU.

- **General topology sub-network**

  A topology that is modeled as a set of point-to-point links, each of which connects two systems. There are several generic types of general topology subnetworks, multipoint links, permanent point-to-point links, dynamic and static point-to-point links.

- **Routing sub-domain**

  A routing sub-domain consists of a set of intermediate systems and end systems located within the same routing domain.

- **Level 2 sub-domain**

  Level 2 sub-domain is the set of all level 2 intermediate systems in a routing domain.

4.1.3 ISO network addressing

IS-IS uses ISO network addresses. Each address identifies a point of connection to the network, such as a router interface, and is called a Network Service Access Point (NSAP).

An end system can have multiple NSAP addresses, in which case the addresses differ only by the last byte (called the *n selector*). Each NSAP represents a service that is available at that node. In addition to having multiple services, a single node can belong to multiple areas.

Each network entity has a special network address called a Network Entity Title (NET). Structurally, a NET is identical to an NSAP address but has an n-selector of 00. Most end systems have one NET. Intermediate systems can have up to three area IDs (area addresses).

NSAP addresses are divided into three parts. Only the area ID portion is configurable.

- **Area ID**

  A variable length field between 1 and 13 bytes long. This includes the Authority and Format Identifier (AFI) as the most significant byte and the area ID.

- **System ID**

  A six-byte system identification. This value is not configurable. The system ID is derived from the system or router ID.

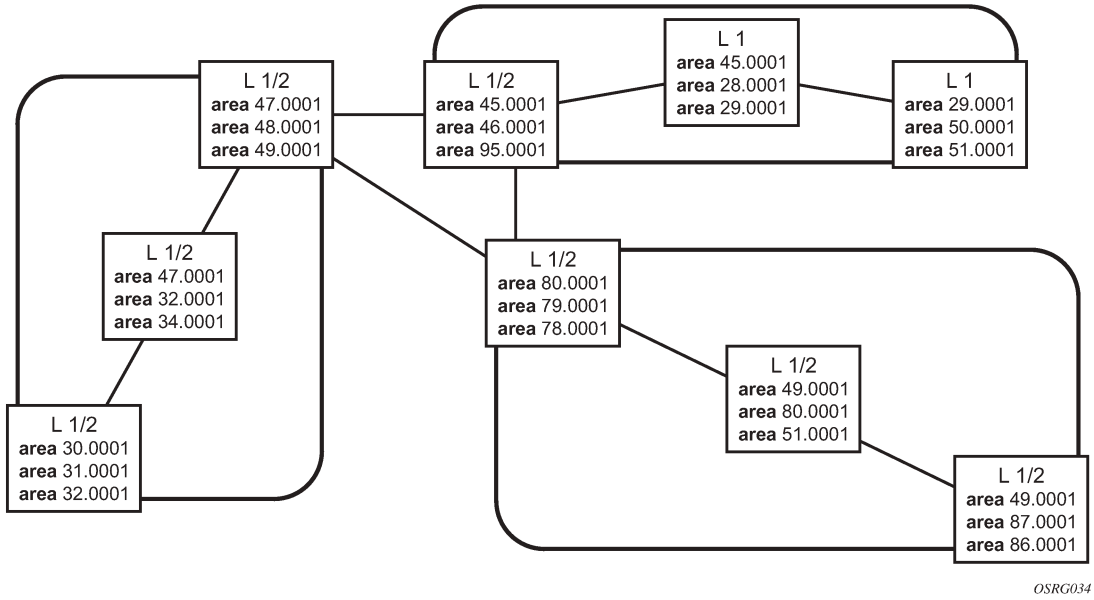
- **Selector ID**

  A one-byte selector identification that must contain zeros when configuring a NET. This value is not configurable. The selector ID is always 00.

Of the total 20 bytes comprising the NET, only the first 13 bytes, the area ID portion, can be manually configured. As few as one byte can be entered or, at most, 13 bytes. If less than 13 bytes are entered, the rest is padded with zeros.

Routers with common area addresses form level 1 adjacencies. Routers with no common NET addresses form level 2 adjacencies, if they are capable (Figure 15: Using area addresses to form adjacencies).

Figure 15: Using area addresses to form adjacencies



4.1.3.1 IS-IS PDU configuration

The following PDUs are used by IS-IS to exchange protocol information:

- **IS-IS hello PDU**

  Routers with IS-IS enabled send hello PDUs to IS-IS-enabled interfaces to discover neighbors and establish adjacencies.

- **Link-state PDUs**

  Contain information about the state of adjacencies to neighboring IS-IS systems. LSPs are flooded periodically throughout an area.

- **Complete sequence number PDUs**

  In order for all routers to maintain the same information, CSNPs inform other routers that some LSPs can be outdated or missing from their database. CSNPs contain a complete list of all LSPs in the current IS-IS database.

- **Partial sequence number PDUs (PSNPs)**

  PSNPs are used to request missing LSPs and acknowledge that an LSP was received.

4.1.3.2 IS-IS operations

The routers perform IS-IS routing as follows.

1. Hello PDUs are sent to the IS-IS-enabled interfaces to discover neighbors and establish adjacencies.
2. IS-IS neighbor relationships are formed if the hello PDUs contain information that meets the criteria for forming an adjacency.

3. The routers can build a link-state PDU based upon their local interfaces that are configured for IS-IS and prefixes learned from other adjacent routers.
4. The routers flood LSPs to the adjacent neighbors except the neighbor from which they received the same LSP. The link-state database is constructed from these LSPs.
5. A Shortest Path Tree (SPT) is calculated by each IS, and from this SPT the routing table is built.

4.1.4 IS-IS route summarization

IS-IS route summarization allows users to create aggregate IPv4 or IPv6 addresses that include multiple groups of IPv4 or IPv6 addresses for a specific IS-IS level. IPv4 and IPv6 routes redistributed from other routing protocols can also be summarized, similar to OSPF configuration using the **area-range** command. IS-IS IPv4 and IPv6 route summarization helps to reduce the size of the Link-State Database (LSDB) and the IPv4 or IPv6 routing table, and reduces the chance of route flapping.

IS-IS route summarization supports:

- level 1, level 1-2, and level 2
- route summarization for the IPv4 or IPv6 routes redistributed from other protocols
- the smallest metric used to advertise summary addresses of all the more specific IPv4 or IPv6 routes
- IS-IS IPv6 route summarization algorithm and SRv6 locator awareness
- full Multitopology Intermediate System to Intermediate System (MT-ISIS) MT0 and MT2 support:
  - The MT0 summary summarizes the MT0 routes and advertises them as an MT0 route.
  - The MT2 summary summarizes the MT2 routes and advertises them as an MT2 route.
- IS-IS Unreachable Prefix Announcement (UPA), as defined in *draft-ietf-lsr-igp-ureach-prefix-announce-01*, to assist BGP FRR for SRv6.

IS-IS UPA

Organizing networks into levels, areas, or IGP domains serves to confine link-state information within specific boundaries. However, the dissemination of state information related to prefix reachability often necessitates propagation across these areas (level 1 and level 2) or domains (through an Autonomous System Boundary Router [ASBR]).

An ASBR, running multiple protocols, acts as a gateway to routers outside the IS-IS domain and those operating with different protocols. The introduction of SRv6 rekindles the significance of summarization and necessitates improved visibility for fast convergence in the reachability of summary member prefixes. However, summarization involves suppressing the individual prefix state, crucial for triggering fast-convergence mechanisms outside the Interior Gateway Routing Protocols (IGPs), such as the Border Gateway Protocol Fast Reroute (FRR) feature.

The UPA technology facilitates the notification of individual prefixes that become unreachable in their area or domain when summarization is employed between areas or domains to advertise reachability. UPA technology allows existing SRv6 deployments that use summarization to react faster after network failures. SRv6 deployments require prompt detection of an unreachable egress router failure. The prompt failure detection can be achieved through UPAs so that BGP SRv6 triggers the SRv6 data plane to switch to the backup path. The MT-ISIS instance on the Area Boundary Router (ABR) monitors and detects events where a summarized member prefix suddenly disappears, leading it to originate a corresponding UPA for a configurable short duration. Simultaneously, the BGP SRv6 component can leverage UPA as a trigger for FRR.

UPA is defined by *draft-ietf-lsr-igp-ureach-prefix-announce-01*. At a high level, a UPA is a regular IS-IS prefix advertised with an exceptionally high metric. In MT-ISIS, a prefix can be advertised with a metric higher than 0xFE000000.

Note: Additionally, for further identification purposes, the Unreachable Prefix Flag (U-Flag) and Unreachable Planned Prefix Flags (UP-Flags) in the IPv4/IPv6 Extended Reachability Attribute Flags, as defined in RFC 7794, exist. These flags are not supported by Nokia, but can be displayed in the LSDB and leaked along the UPA prefix between IS-IS levels and MT-ISIS instances.

The following behavior along a UPA applies:

- The IS-IS support for UPA is for algorithm 0 and flexible algorithm *0 to 255* prefixes advertised in an ABR base routing instance IP Reachability TLV and the SRv6 Locator TLV (in either MT-ISIS MT0 or MT-ISIS MT2).
- The IS-IS algorithm 0 can be used to detect sudden unreachability of the BGP NLRI next-hop address. A received UPA cannot be used to trigger FRR when suddenly an SRv6 locator becomes unreachable.
- UPAs originated on an ABR must have originally been member prefixes of a configured summary prefix.
- Originated UPAs can be host route prefixes (/32 for IPv4 or /128 for IPv6) or a shorter prefix.
- Use the following IS-IS configuration commands to control IS-IS UPAs:

```
configure router isis summary-address advertise-unreachable match-route-tag
configure router isis summary-address advertise-unreachable advertise-route-tag
configure router isis summary-address prefix-unreachable maximum-number-upas
configure router isis summary-address prefix-unreachable process-received-upa
configure router isis summary-address prefix-unreachable upa-lifetime
configure router isis summary-address prefix-unreachable upa-metric
```

- Consider the following for the IS-IS configuration commands to control IS-IS UPAs:
  - For backward compatibility, use the **process-received-upa** command to insert received UPAs into the local UPA routing table, where all unreachable routes are stored.
  - A UPA configurable metric using the max-metric range exists using the **upa-metric** command.
  - Support with **match-route-tag** is provided to use route tags on received prefixes to distinguish monitored summary member prefixes, which can trigger a UPA.
  - The origin of an IS-IS UPA is limited to a brief duration, within the range of the **upa-lifetime** command.
  - Use the **advertise-route-tag** command to filter UPAs when there are multiple ABRs to contain UPAs to an area, and to avoid IGP routing instability.
  - ABRs have a configurable maximum limit for the number of UPAs a router can originate within the range of the **maximum-number-upas** command.
  - UPA flags are not supported; however:
    - IS-IS will interpret and display the flags in the LSDB.
    - Additionally, IS-IS will redistribute these flags in an ABR from level 1 to level 2 by default. Through a configured policy these flags can also be leaked from level 2 to level 1 or between base MT-ISIS instances.
- IS-IS can leak UPAs between base MT-ISIS levels and between base MT-ISIS instances.

- ABRs can filter UPAs while leaking by using route tags and existing export policies, and policy filtering commands apply to UPA prefixes.
- While leaking UPAs between MT-ISIS levels or instances, the metric of a UPA is irrelevant and remains unchanged when leaked.
- BGP listens to the UPA routing table to trigger BGP next-hop resolution, but no UPA is exported through an export policy from the MT-ISIS UPA routing table to BGP.
- BGP conditional expression logic applied toward MT-ISIS prefixes for BGP export are not supported. The conditional expression logic only looks at normal RTM prefixes and not UPA RTM prefixes.

4.1.4.1 Partial SPF calculation

IS-IS supports partial SPF calculation, also referred to as partial route calculation. When an event does not change the topology of the network, IS-IS is not perform full SPF but instead performs an IP reach calculation for the impacted routes. Partial SPF is performed at the receipt of IS-IS LSPs with changes to IP reach TLVs and in general, for any IS-IS LSP TLV and sub-TLV change that does not impact the network topology.

4.1.5 IS-IS multitopology support

Multitopology IS-IS (MT-ISIS) support within SR OS allows for the creation of different topologies within IS-IS that contribute routes to specific route tables for IPv4 unicast, IPv6 unicast, IPv4 multicast, and IPv6 multicast. This capability allows for non-congruent topologies between these different routing tables. As a result, networks are able to control which links or nodes are to be used for forwarding different types of traffic.

For example, MT-ISIS could allow all links to carry IPv4 traffic, while only a subset of links can also carry IPv6 traffic.

SR OS supports the following multitopologies:

- IPv4 Unicast – MT-ID 0
- IPv6 Unicast – MT-ID 2
- IPv4 Multicast – MT-ID 3
- IPv6 Multicast – MT-ID 4

4.1.5.1 Native IPv6 support

IS-IS IPv6 TLVs for IPV6 routing is supported in SR OS. This support is considered native IPv6 routing within IS-IS. However, it has limitations in that IPv4 and IPv6 topologies must be congruent, otherwise traffic may be blackholed. Service providers should ensure that the IPv4 topology and IPv6 topologies are the same if native IPv6 routing is used within IS-IS.

4.1.6 IS-IS administrative tags

IS-IS administrative tags enable a network administrator to configure route tags to tag IS-IS route prefixes. These tags can subsequently be used to control IS-IS route redistribution or route leaking.

IS-IS route tagging can be applied to IP addresses of an interface and to administrative policies with a route map. A network administrator can tag a summary route and then use a route policy to match the tag with one or more attributes for the route.

Using these administrative policies, the operator can control how a router handles route exchanges with its IS-IS neighboring routers. Administrative policies are also used to govern the installation of routes in the routing table.

Route tags allow policies to do the following:

- redistribute routes received from other protocols in the routing table to IS-IS
- redistribute routes or SRv6 locators between levels in an IS-IS routing hierarchy
- summarize routes redistributed into IS-IS or within IS-IS by creating aggregate (summary) addresses

4.1.6.1 Setting route tags

IS-IS route tags are configurable in the following ways:

- for an IS-IS interface
- on an IS-IS passive interface
- for a route redistributed from another protocol to IS-IS
- for a route redistributed from one IS-IS level to another IS-IS level
- for an IS-IS default route
- for an IS-IS summary address or SRv6 locator

4.1.6.2 Using route tags

The configured setting of the IS-IS administrative tags on this or on a neighboring IS-IS router only takes effect if policies are configured to instruct how to process the specified tag value.

Policies configured in the following contexts can process tags where IS-IS is the origin, destination, or both origin and destination protocol:

- **MD-CLI**

```
configure policy-options policy-statement entry from
configure policy-options policy-statement entry action tag
configure policy-options policy-statement default-action tag
```

- **classic CLI**

Note: To configure policy-statements, use the **begin** command to enter the edit mode and use the **commit** command to save your changes.

```
configure router policy-options policy-statement entry from
configure router policy-options policy-statement entry action tag
configure router policy-options policy-statement default-action tag
```

4.1.6.3 Unnumbered interface support

IS-IS supports unnumbered point-to-point interface with both Ethernet and PPP encapsulations.

Unnumbered interfaces borrow the address from other interfaces such as system or loopback interfaces and uses it as the source IP address for packets originated from the interface. This feature supports both dynamic and static ARP for unnumbered interfaces to allow interworking with unnumbered interfaces that may not support dynamic ARP.

An unnumbered interface is an IPv4 capability only used in cases where IPv4 is active (IPv4-only and mixed IPv4/IPv6 environments). When configuring an unnumbered interface, the interface specified for the unnumbered interface (system or other) must have an IPv4 address. Also, the interface type for the unnumbered interface automatically is point-to-point. The unnumbered option can be used in IES and VPRN access interfaces, as well as in a network interface with MPLS support.

4.1.7 Multihomed prefix LFA extensions in IS-IS

4.1.7.1 Feature configuration

Use the following command to configure the Multihomed Prefix (MHP) LFA feature for IP FRR for IS-IS routes, SR-ISIS tunnel, and SRv6-ISIS tunnel FRR:

- **MD-CLI**

```
configure router isis loopfree-alternate multi-homed-prefix preference
```

- **classic CLI**

```
configure router isis loopfree-alternates multi-homed-prefix preference
```

When applied to IP prefixes, IP FRR must also be enabled. Use the following command to allow the programming of the backup paths in the FIB:

- **MD-CLI**

```
configure routing-options ip-fast-reroute
```

- **classic CLI**

```
configure router ip-fast-reroute
```

This feature uses the multihomed prefix model described in RFC 8518 to compute a backup IP next hop via an alternate ABR or ASBR for external prefixes and to an alternate router owner for local anycast prefixes. Without this feature, the backup path is computed to the ASBR, ABR, or router owner, which is the best path for the prefix.

This feature further enhances the multihomed prefix backup path calculation beyond RFC 8518 with the addition of repair tunnels that make use of a PQ node or a P-Q set to reach the alternate exit ABR or ASBR of external prefixes or the alternate owner router of intra-area anycast prefixes.

The base LFA algorithm is applied to all intra-area and external prefixes of IP routes (IP FRR), of SR-ISIS node SID tunnels (SR-ISIS FRR), and to SRv6-ISIS remote locator tunnels (SRv6-ISIS FRR), as usual. Then the MHP LFA is applied to improve the protection coverage for external prefixes and anycast prefixes. For external /32 IPv4 prefixes and /128 IPv6 prefixes and for intra-area /32 IPv4 and /128 IPv6

prefixes with multiple owner routers (anycast prefixes), the base LFA backup path, if found, is preferred over the MHP LFA backup path in the default behavior with the **preference** command set to a value of **none**. The user can force the programming of the MHP LFA backup path by setting **preference** command value to **all**.

After the IP next-hop based MHP LFA is enabled, the extensions to MHP LFA to compute an SR-TE repair tunnel for an SR-ISIS or SRv6-ISIS tunnel are automatically enabled when the following CLI command is configured to enable Topology-Independent Loop-Free Alternate (TI-LFA) or Remote Loop-Free Alternate (RLFA). The computation reuses the SID list of the primary path or the TI-LFA or RLFA backup path of the alternate ABR, ASBR, or alternate owner router.

- **MD-CLI**

```
configure router isis loopfree-alternate remote-lfa
configure router isis loopfree-alternate ti-lfa
```

- **classic CLI**

```
configure router isis loopfree-alternates remote-lfa
configure router isis loopfree-alternates ti-lfa
```

TI-LFA, base LFA, and RLFA (if enabled) are applied to the SR-ISIS node SID tunnels of all intra-area and external /32 IPv4 and /128 IPv6 prefixes as usual, and to all SRv6-ISIS locator tunnels of intra-area and external prefixes of any size.

For node SID SR-ISIS tunnels of external /32 IPv4 and /128 IPv6 prefixes or intra-area /32 IPv4 and /128 IPv6 anycast prefixes, the LFA, TI-LFA, or RLFA backup path is preferred over the MHP LFA backup path in the default behavior with the **preference** command set to a value of **none**. The user can force the programming of the MHP LFA backup by setting the **preference** command value to **all**. Finally, the same preference rule also applies to SRv6-ISIS remote locator tunnels of external prefixes and intra-area prefixes with multiple router owners.

The MHP LFA backup path protects SR-ISIS tunnels and SRv6-ISIS locator tunnels in both algorithm 0 and flexible-algorithm numbers. Therefore, it also extends the protection to any SR-TE LSP, SR-MPLS policy, or SRv6 policy that uses an SR-ISIS SID or an SRv6-ISIS SID of those same prefixes in its configured or computed SID list.

4.1.7.2 Feature applicability

The **multi-homed-prefix** command enables the feature, but its applicability depends on the LFA flavor enabled in the IS-IS instance. The following scenarios are possible:

- Scenario 1
  - **MD-CLI**
    The **loopfree-alternate** and **multi-homed-prefix** commands are enabled.
  - **classic CLI**
    The **loopfree-alternates** and **multi-homed-prefix** commands are enabled.

  The IP next-hop based MHP LFA feature enhances base LFA only; it applies to IP FRR (when the **ip-fast-reroute** command is also enabled) and to SR-ISIS tunnels and SRv6-ISIS tunnels.
- Scenario 2
  - **MD-CLI**

The **loopfree-alternate remote-lfa** and **loopfree-alternate ti-lfa** commands are enabled, or both commands and the **multi-homed-prefix** command is enabled.

– **classic CLI**
The **loopfree-alternates remote-lfa** and **loopfree-alternates ti-lfa** commands are enabled, or both commands and the **multi-homed-prefix** command is enabled.

The enabling of RLFA, TI-LFA, or both on top of the MHP LFA automatically enables the SR OS specific enhancements to RFC 8518 that compute a repair tunnel to the alternate exit ABR or ASBR of external prefixes or to the alternate owner router for intra-area anycast prefixes. This enhancement improves coverage because it computes a SR-TE or SRv6 backup repair tunnel to an alternate ASBR. This forces the packet to go to the alternate ASBR because the RFC 8518 MHP LFA may not find a loop-free path to this alternate ASBR.

4.1.7.3 RFC 8518 MHP LFA for IS-IS

The behavior of this feature is the same as in OSPF. See Multihomed prefix LFA extensions in OSPF.

4.2 FIB prioritization

The RIB processing of specific routes can be prioritized through the use of the **rib-priority** command. This command allows specific routes to be prioritized through the protocol processing so that updates are propagated to the FIB as quickly as possible.

The **rib-priority** command is configured within the global IS-IS routing context, and the administrator has the option to either specify a prefix list or an IS-IS tag value. If a prefix list is specified, route prefixes matching any of the prefix list criteria is considered high priority. If instead an IS-IS tag value is specified, any IS-IS route with that tag value is considered high priority.

Routes designated as high priority are the first routes processed and passed to the FIB update process so that the forwarding engine can be updated. All known high priority routes should be processed before the IS-IS routing protocol moves on to other standard priority routes. This feature has the most impact when a large number of routes are learned through the IS-IS routing protocols.

4.3 IS-IS graceful restart helper

IS-IS supports the graceful restart helper function which provides an IS-IS neighbor a grace period during a control plane restart to minimize service disruption. When the control plane of a GR-capable router fails or restarts, the neighboring routers supporting the graceful restart helper mode (GR helpers) temporarily preserve IS-IS forwarding information. Traffic continues to be forwarded to the restarting router using the last known forwarding tables. If the control plane of the restarting router comes back up within the grace period, the restarting router resumes normal IS-IS operation. If the grace period expires, then the restarting router is presumed inactive and the IS-IS topology is recalculated to route traffic around the failure.

4.3.1 BFD interaction with graceful restart

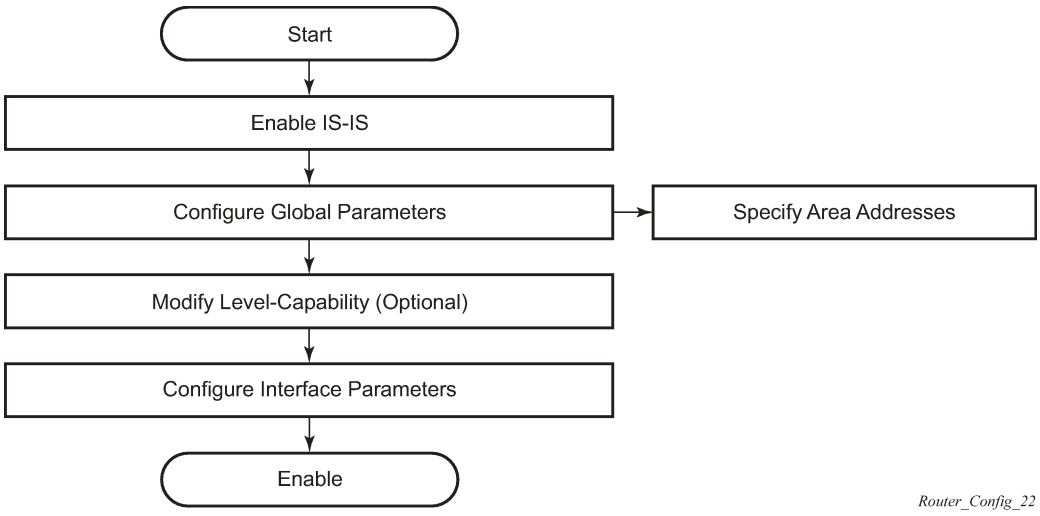
If the SR OS router is providing a grace period to an adjacent neighbor and the BFD session associated with that neighbor fails, the behavior is determined by the C-bit values sent by each neighbor.

- If both BFD end-points have set their C-bit value, then the graceful restart helper mode is canceled and any routes from that neighbor that are marked as stale are removed from the forwarding table.
- If either of the BFD end-points has not set their C-bit value, then the graceful restart helper mode continues.

4.4 IS-IS configuration process overview

Figure 16: IS-IS configuration and implementation flow displays the process to provision basic IS-IS parameters.

Figure 16: IS-IS configuration and implementation flow



4.5 Configuration notes

This section describes IS-IS configuration restrictions.

4.5.1 General

- IS-IS must be enabled on each participating router.
- There are no default network entity titles.
- There are no default interfaces.
- By default, the routers are assigned a level 1/level 2 level capability.

4.6 Configuring IS-IS with CLI

This section provides information to configure IS-IS using the command line interface.

4.6.1 IS-IS configuration overview

4.6.1.1 Router levels

The router's level capability can be configured globally and on a per-interface basis. The interface-level parameters specify the interface's routing level. The neighbor capability and parameters define the adjacencies that are established.

IS-IS is not enabled by default. When IS-IS is enabled, the global default level capability is level 1/2 which enables the router to operate as either a level 1, a level 2 router with the associated databases, or both. The router runs separate shortest path first (SPF) calculations for the level 1 area routing and for the level 2 multi-area routing to create the IS-IS routing table.

The level value can be modified on both or either of the global and interface levels to be only level 1-capable, only level 2-capable or level 1 and level 2-capable.

If the default value is not modified on any routers in the area, then the routers try to form both level 1 and level 2 adjacencies on all IS-IS interfaces. If the default values are modified to level 1 or level 2, then the number of adjacencies formed are limited to that level only.

4.6.1.2 Configuring area address attributes

About this task

The area ID, also called an area address, specifies the area address portion of the NET which is used to define the IS-IS area to which the router belongs. At least one area ID should be configured on each router participating in IS-IS. You can configure a maximum of three area IDs per router.

Use the following command to configure the area ID:

- **MD-CLI**

```
configure router isis area-address
```

- **classic CLI**

```
configure router isis area-id
```

The area address identifies a point of connection to the network, such as a router interface, and is called a Network Service Access Point (NSAP). The routers in an area manage routing tables about destinations within the area. The Network Entity Title (NET) value is used to identify the IS-IS area to which the router belongs.

NSAP addresses are divided into three parts. Only the Area ID portion is configurable.

Procedure

Step 1. Set the area ID.

A variable length field between 1 and 13 bytes long. This includes the Authority and Format Identifier (AFI) as the most significant byte and the area ID.

Step 2. Set the system ID.

A six-byte system identification. This value is not configurable. The system ID is derived from the system or router ID.

Step 3. Set the selector ID.

A one-byte selector identification that must contain zeros when configuring a NET. This value is not configurable. The selector ID is always 00.

Example

The following example displays ISO addresses in IS-IS address format:
MAC address 00:a5:c7:6b:c4:9049.0011.00a5.c76b.c490.00 IP address: 218.112.14.5
49.0011.2181.1201.4005.00

4.6.1.3 Interface level capacity

The level capability value configured on the interface level is compared to the level capability value configured on the global level to determine the type of adjacencies that can be established. The default level capability for routers and interfaces is level 1/2.

Table 3: Potential adjacency displays configuration combinations and the potential adjacencies that can be formed.

Table 3: Potential adjacency

Global level	Interface level	Potential adjacency
L 1/2	L 1/2	Level 1 and/or Level 2
L 1/2	L 1	Level 1 only
L 1/2	L 2	Level 2 only
L 2	L 1/2	Level 2 only
L 2	L 2	Level 2 only
L 2	L 1	—
L 1	L 1/2	Level 1 only
L 1	L 2	—
L 1	L 1	Level 1 only

4.6.1.4 Route leaking

Nokia's implementation of IS-IS route leaking is performed in compliance with RFC 2966, *Domain-wide Prefix Distribution with Two-Level IS-IS*. As previously stated, IS-IS is a routing domain (an autonomous system running IS-IS) which can be divided into level 1 areas with a level 2-connected subset (backbone) of the topology that interconnects all of the level 1 areas. Within each level 1 area, the routers exchange link state information. Level 2 routers also exchange level 2 link state information to compute routes between areas.

Routers in a level 1 area typically only exchange information within the level 1 area. For IP destinations not found in the prefixes in the level 1 database, the level 1 router forwards PDUs to the nearest router that is in both level 1/level 2 with the *attached bit* set in its level 1 link-state PDU.

There are many reasons to implement domain-wide prefix distribution. The goal of domain-wide prefix distribution is to increase the granularity of the routing information within the domain. The routing mechanisms specified in RFC 1195 are appropriate in many situations and account for excellent scalability properties. However, in specific circumstances, the amount of scalability can be adjusted which can distribute more specific information than described by RFC 1195.

Distributing more prefix information can improve the quality of the resulting routes. A well-known property of default routing is that loss of information can occur. This loss of information affects the computation of a route based upon less information which can result in sub-optimal routes.

4.6.2 Basic IS-IS configuration

For IS-IS to operate on the routers, IS-IS must be explicitly enabled, and at least one area address and interface must be configured. If IS-IS is enabled but no area address or interface is defined, the protocol is enabled but no routes are exchanged. When at least one area address and interface are configured, then adjacencies can be formed and routes exchanged.

To configure IS-IS, perform the following steps.

1. Enable IS-IS (specifying the instance ID of multi-instance IS-IS is to be enabled).
2. Modify the level capability, if necessary, on the global level (default is level-1/2).
3. Define area addresses.
4. Configure IS-IS interfaces.

The following example displays IS-IS default values.

Example: MD-CLI

```
[ex:/configure router "Base" isis 0]
A:admin@node-2# info detail
 ## apply-groups
 ## apply-groups-exclude
    admin-state disable
 ## authentication-keychain
 ## authentication-key
 ## authentication-type
    csnp-authentication true
    psnp-authentication true
    advertise-passive-only false
 ## advertise-router-capability
    advertise-tunnel-link false
    all-l1isis 01:80:c2:00:00:14
    all-l2isis 01:80:c2:00:00:15
    authentication-check true
 ## default-route-tag
    ldp-sync true
    hello-authentication true
    ignore-attached-bit false
    ignore-lsp-errors false
    ignore-narrow-metric false
    iid-tlv false
    ipv4-multicast-routing native
    ipv4-routing true
    ipv6-multicast-routing native
 ## export-limit
 ## graceful-restart
    entropy-label {alse
        override-tunnel-elc false
```

```
    }sp-lifetime 1200
## multi-topologymaining-lifetime
    multicast-import {
        ipv4 falseetection false
        ipv6 falset-interlevel false
    }verload-export-external false
## overloadfalse
## overload-on-boots-tlv false
## prefix-limitndwidth
    lsp-refresh {
        interval 600nstance false
        half-lifetime true false
    }uppress-attached-bit false
    rib-priority {.0000.0000
        high {
         ## prefix-list
         ## tag
        }
    }
    flexible-algorithms {
        admin-state disable
        advertise-admin-group prefer-ag
     ## flex-algonitial-wait 1000
    }       spf-second-wait 1000
    traffic-engineering-options {
        advertise-delay false
        ipv6 falsex-wait 5000
     ## application-link-attributes
    }       lsp-second-wait 1000
    segment-routing {
     ## apply-groups
     ## apply-groups-exclude
        admin-state disable
        adj-sid-hold 15
        class-forwarding false
        entropy-label true
     ## export-tunnel-table
     ## srlb
     ## tunnel-mtu
        tunnel-table-pref 11
        adjacency-sid {
            allocate-dual-sids false
        mapping-server {
            admin-state disable
         ## node-sid-map
        }maximum-sid-depth {
    }    ## override-bmi
    segment-routing-v6 {d
     ## apply-groups
     ## apply-groups-exclude
        admin-state disable
        adj-sid-hold 15se
     ## locator-sid false
     ## micro-segment-locator
    }   }
    igp-shortcut {atistics {
     ## apply-groupsfalse
     ## apply-groups-exclude
        admin-state disable
        allow-sr-over-srte false
        tunnel-next-hop {
            family ipv4 {
             ## apply-groups
             ## apply-groups-exclude
```

```
                resolution none
                resolution-filter {
                    rsvp false
                    sr-te false
                }
            }
            family srv6 {
             ## apply-groups
             ## apply-groups-exclude
                resolution none
                resolution-filter {
                    rsvp false
                    sr-te false
                 }
            }
        }
    }
## interface
    level 1 {
     ## apply-groups
     ## apply-groups-exclude
     ## authentication-keychain
     ## authentication-key
     ## authentication-type
        csnp-authentication true
        psnp-authentication true
        advertise-router-capability true
        database-export-exclude false
        default-ipv4-multicast-metric 10
        default-ipv6-multicast-metric 10
        default-ipv6-unicast-metric 10
        default-metric 10
        external-preference 160
        hello-authentication true
     ## hello-padding
        loopfree-alternate-exclude false
        lsp-mtu-size 1492
        preference 15
        wide-metrics-only false
        bier {
            admin-state disable
         ## template
        }
    }
    level 2 {
     ## apply-groups
     ## apply-groups-exclude
     ## authentication-keychain
     ## authentication-key
     ## authentication-type
        csnp-authentication true
        psnp-authentication true
        advertise-router-capability true
        database-export-exclude false
        default-ipv4-multicast-metric 10
        default-ipv6-multicast-metric 10
        default-ipv6-unicast-metric 10
        default-metric 10
        external-preference 165
        hello-authentication true
     ## hello-padding
        loopfree-alternate-exclude false
        lsp-mtu-size 1492
        preference 18
```

```
        wide-metrics-only false
        bier {
            admin-state disable
         ## template
        }
    }
 ## link-group
 ## summary-address
```

Example: classic CLI

```
A:node-2>config>router>isis# info detail
----------------------------------------------
           no system-id
           no router-id
           level-capability level-1/2
           no graceful-restart
           no auth-keychain
           no authentication-key
           no authentication-type
           authentication-check
           csnp-authentication
           no ignore-lsp-errors
           no ignore-narrow-metric
           lsp-lifetime 1200
           lsp-mtu-size 1492
           lsp-refresh-interval 600
           no export-limit
           no export
           no import
           hello-authentication
           psnp-authentication
           no traffic-engineering
           no reference-bandwidth
           no default-route-tag
           no disable-ldp-sync
           no advertise-passive-only
           no advertise-router-capability
           no hello-padding
           no ldp-over-rsvp
           no advertise-tunnel-link
           no ignore-attached-bit
           no suppress-attached-bit
           no iid-tlv-enable
           no poi-tlv-enable
           no prefix-limit
           no loopfree-alternates
           no rib-priority high
           ipv4-routing
           no ipv6-routing
           ipv4-multicast-routing native
           ipv6-multicast-routing native
           no multi-topology
           no unicast-import-disable both
           no multicast-import both
           no strict-adjacency-check
           igp-shortcut
               shutdown
               tunnel-next-hop
                   family ipv4
                       resolution disabled
                       resolution-filter
                           no rsvp
```

```
                        no sr-te
                exit
                family ipv6
                    resolution disabled
                    resolution-filter
                        no rsvp
                        no sr-te
                exit
                family srv4
                    resolution disabled
                    resolution-filter
                        no rsvp
                        no sr-te
                exit
                family srv6
                    resolution disabled
                    resolution-filter
                        no rsvp
                        no sr-te
                exit
            exit
        exit
        timers
            lsp-wait 5000 lsp-initial-wait 10 lsp-second-wait 1000
            sfp-wait 10000 sfp-initial-wait 1000 sfp-second-wait 1000
        exit
        level 1
            advertise-router-capability
            no hello-padding
            no lsp-mtu-size
            no auth-keychain
            no authentication-key
            no authentication-type
            csnp-authentication
            external-preference 160
            hello-authentication
            no loopfree-alternate-exclude
            preference 15
            psnp-authentication
            no wide-metrics-only
            default-metric 10
           default-ipv4-multicast-metric 10
            default-ipv6-unicast-metric 10
            default-ipv6-multicast-metric 10
        exit
        level 2
            advertise-router-capability
            no hello-padding
            no lsp-mtu-size
            no auth-keychain
            no authentication-key
            no authentication-type
            csnp-authentication
            external-preference 165
           hello-authentication
            no loopfree-alternate-exclude
            preference 18
            psnp-authentication
            no wide-metrics-only
            default-metric 10
            default-ipv4-multicast-metric 10
            default-ipv6-unicast-metric 10
            default-ipv6-multicast-metric 10
        exit
```

```
            segment-routing
                shutdown
                adj-sid-hold 15
                no export-tunnel-table
                no prefix-sid-range
                no tunnel-table-pref
                no tunnel-mtu
                mapping-server
                    shutdown
                exit
            exit
            no shutdown
```

4.6.3 Common configuration tasks

To implement IS-IS in your network, you must enable IS-IS on each participating router.

To assign different level to the routers and organize your network into areas, modify the level capability defaults on end systems from level 1/2 to level 1. Routers communicating to other areas can retain the level 1/2 default.

On each router, at least one area ID also called the area address should be configured as well as at least one IS-IS interface.

1. Enable IS-IS.
2. Configure global IS-IS parameters (configure area addresses).
3. Configure IS-IS interface-specific parameters.

4.6.4 Configuring IS-IS components

Use the CLI commands displayed in the following subsections to configure IS-IS components.

4.6.4.1 Enabling IS-IS

IS-IS is disabled by default and must be configured and administratively enabled for the protocol to be active. SR OS also supports multi-instance IS-IS, which allows separate instances of the IS-IS protocol to run independently of the SR OS router.

Use the following command to configure an instance ID for IS-IS instances.

- **MD-CLI**

```
configure router isis isis-instance
```

- **classic CLI**

```
configure router isis
```

Caution: Careful planning is essential to implement commands that can affect the behavior of global and interface levels.

4.6.4.2 Modifying router-level parameters

When IS-IS is enabled, the router operates with both level-1 and level-2 routing. The **level-capability** value can be configured on the global level and also on the interface level. The **level-capability** value determines which level values can be assigned on the router level or on an interface-basis. To operate as only a level-1 router or only a level-2 router, you must explicitly specify the level number.

- Level 1 routes only within an area.
- Level 2 routes to destinations outside an area, toward other eligible level 2 routers.

Use the following command to configure the router capability level:

- **MD-CLI**

```
configure router isis level-capability (1|2|1/2)
```

- **classic CLI**

```
configure router isis level-capability {level-1|level-2|level-1/2}
```

Use the following command to change the default global value for the router to operate as a level 1 router or a level 2 router:

- **MD-CLI**

```
configure router isis level level-number (1|2)
```

- **classic CLI**

```
configure router isis level (1|2)
```

Note: If you modify the level, the protocol shuts down and restarts. This can affect adjacencies and routes.

The following is an example of level-capability and level configuration.

Example: MD-CLI

```
[ex:/configure router "Base" isis 5]
A:admin@node-2# info
    level-capability 2
    level 2 {
        ...
    }
```

Example: classic CLI

```
A:node-2>config>router>isis# info
----------------------------------------------
        shutdown
        level-capability level-2
        level 2
            ...
        exit
----------------------------------------------
```

4.6.4.3 Configuring ISO area addresses

Use the following command to configure the area ID, also called an address. You can configure a maximum of three area IDs per router:

- **MD-CLI**

```
configure router isis area-address
```

- **classic CLI**

```
configure router isis area-id
```

The following example shows the router area ID configuration.

Example: MD-CLI

```
[ex:/configure router "Base" isis 5]
A:admin@node-2# info
    ...
    area-address [49.0180.0001 49.0180.0002 49.0180.0003]
    ...
    }
```

Example: classic CLI

```
A:node-2>config>router>isis# info
----------------------------------------------
        ...
        area-id 49.0180.0001
        area-id 49.0180.0002
        area-id 49.0180.0003
        ...
----------------------------------------------
```

4.6.4.4 Configuring global IS-IS parameters

Commands and command options configured on the global level are inherited at the interface levels. Configurations in the interface and interface-level take precedence over global configurations.

The following example shows a modified global-level configuration.

Example: MD-CLI

```
[ex:/configure router "Base" isis 5]
A:admin@node-2# info
    authentication-key "authkey hash2"
    authentication-type password
    authentication-check true
    level-capability 2
    overload-export-external true
    traffic-engineering true
    area-address [49.0180.0001 49.0180.0002 49.0180.0003]
    }
```

Example: classic CLI

```
A:node-2>config>router>isis# info
----------------------------------------------
        level-capability level-2
        area-id 49.0180.0001
        area-id 49.0180.0002
        area-id 49.0180.0003
        authentication-key "auth-key" hash
        authentication-type password
        overload timeout 90
        traffic-engineering
----------------------------------------------
```

4.6.4.5 Migration to IS-IS multitopology

About this task

To migrate to IS-IS Multitopology (MT) for IPv6, perform the steps described in this topic.

Procedure

Step 1. Use the following command to enable sending and receiving of IPv6 unicast reachability information in IS-IS MT TLVs on all the routers that support MT.

```
configure router isis multi-topology ipv6-unicast
```

Example

MD-CLI

```
[ex:/configure router "Base" isis 0]
A:admin@node-2# info
    ...
    ipv6-routing native
    multi-topology {
        ipv6-unicast true
        ipv4-multicast true
    }
    ...
```

Example

classic CLI

```
A:node-2>config>router>isis# info detail
----------------------------------------------
...
        ipv4-routing
        ipv6-routing native
        multi-topology
            ipv6-unicast
        exit
...
----------------------------------------------
```

Step 2. Use the following commands to ensure that all routers to be configured for MT have the IPv6 reachability information required by MT TLVs.

a. Display the IS-IS IPv6 unicast topology.

```
show router isis topology ipv6-unicast
```

Example

```
===============================================================================
Rtr Base ISIS Instance 0 Topology Table
===============================================================================
Node                              Interface                 Nexthop
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
IS-IS IPv6 paths (MT-ID 2), Level 1
-------------------------------------------------------------------------------
Dut-E.00                          C1/1/8-E1/1/6             Dut-E
-------------------------------------------------------------------------------
IS-IS IPv6 paths (MT-ID 2), Level 2
-------------------------------------------------------------------------------
Dut-E.00                          C1/1/8-E1/1/6             Dut-E
===============================================================================
```

b. Use the following command to display the database details.

```
show router isis database detail
```

Example

```
===============================================================================
Rtr Base ISIS Instance 0 Database (detail)
===============================================================================
Displaying Level 1 database
-------------------------------------------------------------------------------
LSP ID    : ALA-49.00-00                            Level     : L1
Sequence  : 0x22b                Checksum  : 0x60e4   Lifetime  : 1082
Version   : 1                    Pkt Type  : 18       Pkt Ver   : 1
Attributes: L1L2                 Max Area  : 3
SysID Len : 6                    Used Len  : 404      Alloc Len : 1492

TLVs :
Area Addresses  :
  Area Address    : (13) 47.4001.8000.00a7.0000.ffdd.0007
Supp Protocols  :
  Protocols       : IPv4 IPv6
IS-Hostname     :
  Hostname        : ALA-49
TE Router ID    :
  Router ID       : 10.10.10.104
Internal Reach  :
  IP Prefix       : 10.10.10.104/32    (Dir. :Up)  Metric  : 0 (I)
  IP Prefix       : 10.10.4.0/24       (Dir. :Up)  Metric  : 10 (I)
  IP Prefix       : 10.10.5.0/24       (Dir. :Up)  Metric  : 10 (I)
  IP Prefix       : 10.10.7.0/24       (Dir. :Up)  Metric  : 10 (I)
  IP Prefix       : 10.10.0.0/24       (Dir. :Up)  Metric  : 10 (I)
  IP Prefix       : 10.0.0.0/24        (Dir. :Up)  Metric  : 10 (I)
MT IPv6 Reach.  :
  MT ID           : 2
  IPv6 Prefix     : 3ffe::101:100/120
                    Flags : Up Internal Metric : 10
  IPv6 Prefix     : 10::/64
                    Flags : Up Internal Metric : 10
I/f Addresses   :
```

```
  IP Address      : 10.10.10.104
  IP Address      : 10.10.4.3
  IP Address      : 10.10.5.3
  IP Address      : 10.10.7.3
  IP Address      : 10.10.0.16
  IP Address      : 10.0.0.104
I/f Addresses IPv6 :
  IPv6 Address    : 3FFE::101:101
  IPv6 Address    : 10::104
TE IP Reach.    :
  IP Prefix       : 10.10.10.104/32      (Dir. :Up)  Metric  : 0
  IP Prefix       : 10.10.4.0/24         (Dir. :Up)  Metric  : 10
  IP Prefix       : 10.10.5.0/24         (Dir. :Up)  Metric  : 10
  IP Prefix       : 10.10.7.0/24         (Dir. :Up)  Metric  : 10
  IP Prefix       : 10.10.0.0/24         (Dir. :Up)  Metric  : 10
  IP Prefix       : 10.0.0.0/24          (Dir. :Up)  Metric  : 10
Authentication  :
  Auth Type       : Password(1) (116 bytes)

Level (1) LSP Count : 1

Displaying Level 2 database
-------------------------------------------------------------------------------
LSP ID    : ALA-49.00-00                                Level     : L2
Sequence  : 0x22c                Checksum  : 0xb888   Lifetime  : 1082
Version   : 1                    Pkt Type  : 20       Pkt Ver   : 1
Attributes: L1L2                 Max Area  : 3
SysID Len : 6                    Used Len  : 304      Alloc Len : 1492

TLVs :
Area Addresses  :
  Area Address    : (13) 47.4001.8000.00a7.0000.ffdd.0007
Supp Protocols  :
  Protocols       : IPv4 IPv6
IS-Hostname     :
  Hostname        : ALA-49
TE Router ID    :
  Router ID       : 10.10.10.104
Internal Reach  :
  IP Prefix       : 10.10.10.104/32    (Dir. :Up)  Metric  : 0 (I)
  IP Prefix       : 10.10.4.0/24       (Dir. :Up)  Metric  : 10 (I)
  IP Prefix       : 10.10.5.0/24       (Dir. :Up)  Metric  : 10 (I)
  IP Prefix       : 10.10.7.0/24       (Dir. :Up)  Metric  : 10 (I)
  IP Prefix       : 10.10.0.0/24       (Dir. :Up)  Metric  : 10 (I)
  IP Prefix       : 10.0.0.0/24        (Dir. :Up)  Metric  : 10 (I)
MT IPv6 Reach.  :
  MT ID           : 2
  IPv6 Prefix     : 3ffe::101:100/120
                    Flags : Up Internal Metric : 10
  IPv6 Prefix     : 10::/64
                    Flags : Up Internal Metric : 10
I/f Addresses   :
  IP Address      : 10.10.10.104
  IP Address      : 10.10.4.3
  IP Address      : 10.10.5.3
  IP Address      : 10.10.7.3
  IP Address      : 10.10.0.16
  IP Address      : 10.0.0.104
I/f Addresses IPv6 :
  IPv6 Address    : 3FFE::101:101
  IPv6 Address    : 10::104
TE IP Reach.    :
  IP Prefix       : 10.10.10.104/32      (Dir. :Up)  Metric  : 0
  IP Prefix       : 10.10.4.0/24         (Dir. :Up)  Metric  : 10
```

```
  IP Prefix       : 10.10.5.0/24         (Dir. :Up)  Metric  : 10
  IP Prefix       : 10.10.7.0/24         (Dir. :Up)  Metric  : 10
  IP Prefix       : 10.10.0.0/24         (Dir. :Up)  Metric  : 10
  IP Prefix       : 10.0.0.0/24          (Dir. :Up)  Metric  : 10
Authentication  :
  Auth Type       : MD5(54) (16 bytes)

Level (2) LSP Count : 1
-------------------------------------------------------------------------------
Flags : D = Prefix Leaked Down
      : N = Node Flag
      : R = Re-advertisement Flag
      : S = Sub-TLVs Present
      : X = External Prefix Flag
===============================================================================
```

Step 3. Use the following command to configure MT TLVs for IPv6 SPF.

```
configure router isis ipv6-routing mt
```

Example

MD-CLI

```
[ex:/configure router "Base" isis 0]
A:admin@node-2# info
    ...
    ipv6-routing mt
    multi-topology {
        ipv6-unicast true
        ipv4-multicast true
    }
    ...
----------------------------------------------
```

Example

classic CLI

```
A:node-2>config>router>isis# info detail
----------------------------------------------
...
        ipv4-routing
        ipv6-routing mt
        multi-topology
            ipv6-unicast
        exit
...
----------------------------------------------
```

Step 4. Use the following commands to verify the IPv6 routes for IS-IS.

a. Display the IPv6 unicast routes.

```
show router isis routes ipv6-unicast
```

Example

```

===============================================================================
Rtr Base ISIS Instance 0 Route Table
===============================================================================
```

```
Prefix[Flags]                     Metric     Lvl/Typ     Ver.  SysID/Hostname
  NextHop                                                MT     AdminTag/SID[F]
-------------------------------------------------------------------------------
2001:db8:1::3/128                 0          1/Int.      8     Dut-C
   ::                                                      2       0
2001:db8:1::5/128                 123        1/Int.      10    Dut-E
   fe80::e28:1ff:fe01:6-"C1/1/8-E1/1/6"                    2       0
2001:db8:100::/126                123        1/Int.      9     Dut-C
   ::                                                      2       0
-------------------------------------------------------------------------------
No. of Routes: 3 (3 paths)
-------------------------------------------------------------------------------
Flags        : L = LFA nexthop available
SID[F]       : R  = Re-advertisement
               N  = Node-SID
               nP = no penultimate hop POP
               E  = Explicit-Null
               V  = Prefix-SID carries a value
               L  = value/index has local significance
===============================================================================
```

b. Display the IPv6 information in the route table.

```
show router route-table ipv6
```

Example

```
===============================================================================
IPv6 Route Table (Router: Base)
===============================================================================
Dest Prefix                                 Type    Proto    Age         Pref
       Next Hop[Interface Name]                                  Metric
-------------------------------------------------------------------------------
10::/64                                     Local   Local    05h35m28s   0
       to-104                                                    0
-------------------------------------------------------------------------------
No. of Routes: 1
===============================================================================
```

4.6.4.6 Configuring IS-IS interfaces

You must configure at least one IS-IS interface for IS-IS to work. There are no default interfaces applied to the router’s IS-IS instance. An interface belongs to all areas configured on a router. Interfaces cannot belong to separate areas.

4.6.4.6.1 Configuring interfaces with level capability

Describes how to configure IS-IS interfaces with level capability.

About this task

Use the following procedure to configure and enable IP interfaces for IS-IS. You must configure at least one IS-IS interface for IS-IS to work.

Procedure

Step 1. Use the following command to configure IP interfaces to use for IS-IS.

```
configure router interface
```

Example

MD-CLI

```
[ex:/configure router "Base"]
A:admin@node-2# info
    interface "NOK-1-1" {
    }
    interface "NOK-1-2" {
    }
    interface "NOK-1-3" {
    }
    interface "NOK-1-5" {
    }
    interface "system" {
    }
    interface "to-103" {
    }
```

Example

classic CLI

```
A:node-2>config>router# info
#--------------------------------------------------
        interface "NOK-1-1"
            no shutdown
        exit
        interface "NOK-1-2"
            no shutdown
        exit
        interface "NOK-1-3"
            no shutdown
        exit
        interface "NOK-1-5"
            no shutdown
        exit
        interface "system"
            no shutdown
        exit
        interface "to-103"
            no shutdown
        exit
#--------------------------------------------------
```

Step 2. Use the following command to reference the interfaces created in step 1 for IS-IS.

```
configure router isis interface
```

Example

MD-CLI

```
[ex:/configure router "Base" isis 0]
A:admin@Dut-AC# info
    ...
```

```
    interface "NOK-1-1" {
    }
    interface "NOK-1-2" {
    }
    interface "ALA-1-3" {
    }
    interface "ALA-1-5" {
    }
    interface "system" {
    }
    interface "to-103" {
    }
```

Example

classic CLI

```
A:node-2>config>router>isis# interface "NOK-1-2"
*A:node-2>config>router>isis>if#
...
```

Step 3. Use the following commands to configure level 1 and level 2 options and the level-capability for an interface. The **level-capability** value determines which level values are used.

```
configure router isis level
configure router isis interface
configure router isis interface level-capability
configure router isis interface-type
```

Note: For point-to-point interfaces, only values configured under level 1 are used, regardless of the operational level of the interface.

The following example shows global and interface-level configurations.

Example

MD-CLI

```
[ex:/configure router "Base" isis 0]
A:admin@node-2# info
    authentication-key "auth-key= hash2"
    authentication-type password
    ipv6-routing native
    level-capability 2
    traffic-engineering true
    area-address [49.0180.0001 49.0180.0002 49.0180.0003]
    multi-topology {
        ipv6-unicast true
        ipv4-multicast false
    }
    interface "ALA-1-2" {
        level-capability 2
    }
    interface "ALA-1-3" {
        interface-type point-to-point
        level-capability 1
    }
    interface "ALA-1-5" {
        interface-type point-to-point
        level-capability 1
    }
    interface "system" {
```

```
    }
    interface "to-103" {
    }
    level 1 {
        wide-metrics-only true
    }
    level 2 {
        wide-metrics-only true
    }
```

Example

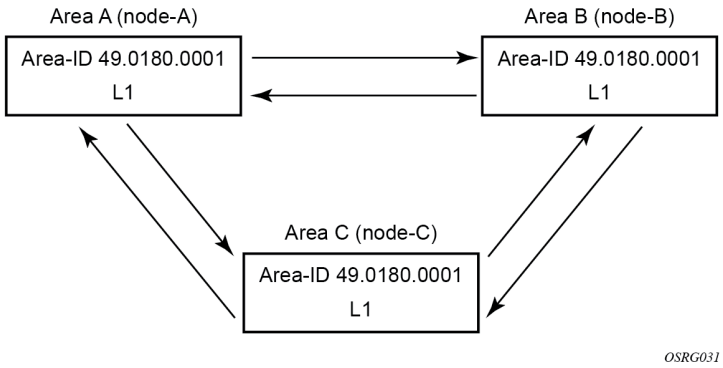
classic CLI

```
A:node-2>config>router>isis# info
----------------------------------------------
        level-capability level-2
        area-id 49.0180.0001
        area-id 49.0180.0002
        area-id 49.0180.0003
        authentication-key "auth-key" hash
        authentication-type password
        traffic-engineering
        level 1
            wide-metrics-only
        exit
        level 2
            wide-metrics-only
        exit
        interface "system"
        exit
        interface "ALA-1-2"
            level-capability level-2
        exit
        interface "ALA-1-3"
            level-capability level-1
            interface-type point-to-point
        exit
        interface "ALA-1-5"
            level-capability level-1
            interface-type point-to-point
        exit
        interface "to-103"
        exit
----------------------------------------------
```

4.6.4.6.2 Configuring a level 1 area

Interfaces are configured in the configure router interface context, as shown in Figure 17: Configuring a level 1 area and the procedure Configuring interfaces with level capability.

Figure 17: Configuring a level 1 area



The following example displays the command usage to configure a level 1 area.

Example: MD-CLI

```
[ex:/configure router "Base" isis 0]
A:admin@node-A# info
    ...
    level-capability 1
    ...
    area-address [49.0180.0001 49.0180.0002 49.0180.0003]
    ...
    }
    interface "A-B" {
        level-capability 1
    }
    interface "A-C" {
        level-capability 1
    }
    interface "B-A" {
        level-capability 1
    }
    interface "B-C" {
        level-capability 1
    }
    interface "C-A" {
        level-capability 1
    }
    interface "C-B" {
        level-capability 1
    }
    ....
```

Example: classic CLI

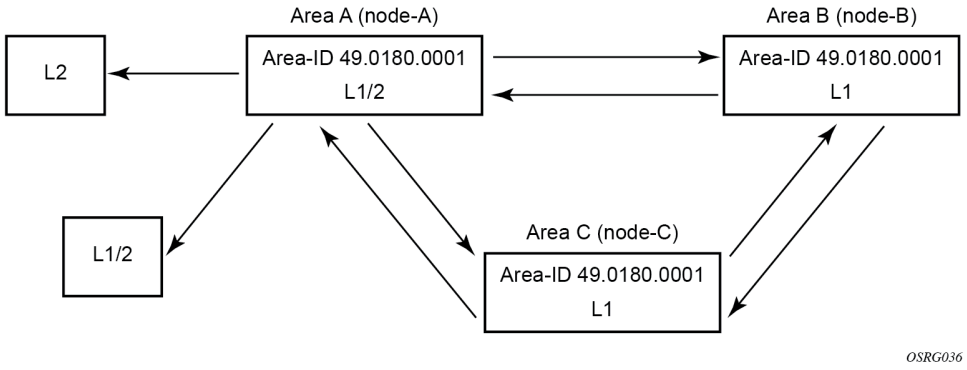
```
A:node-A>config>router>isis# info
----------------------------------------------
        level-capability level-1
        area-id 49.0180.0001
        interface "system"
        exit
        interface "A-B"
        exit
        interface "A-C"
        exit
```

```
-----------------------------------------------

A:node-B>config>router>isis# info
-----------------------------------------------
        level-capability level-1
        area-id 49.0180.0002
        interface "system"
        exit
        interface "B-A"
        exit
        interface "B-C"
        exit
-----------------------------------------------

A:node-C>config>router>isis# info
#---------------------------------------
echo "ISIS"
-----------------------------------------------
        level-capability level-1
        area-id 49.0180.0003
        interface "system"
        exit
        interface "C-A"
        exit
        interface "C-B"
        exit
-----------------------------------------------
```

4.6.4.6.3 Modifying a router's level capability

In the example shown in Configuring a level 1 area, A, B, and C are configured as level 1 systems. Level 1 systems communicate with other level 1 systems in the same area. In the following example, area A is modified to set the level capability to level 1 and 2, as shown in Figure 18: Configuring a level 1/2 area. Now, the level 1 systems in the area with NET 47.0001 forward PDUs to area A for destinations that are not in the local area.

Figure 18: Configuring a level 1/2 area



The following example shows the command usage to change area A from a level 1 system to a level 1/2 system.

Example: MD-CLI

```
[ex:/configure router "Base" isis 0]
```

```
A:admin@node-A# level-capability 1/2

*[ex:/configure router "Base" isis 0]
A:admin@node-A# info
    ...
    level-capability 1/2
    ...
    }
```

Example: classic CLI

Note: The **level-1/2** default configuration option is not displayed in the output of the **info** command.

```
A:node-A>config>router>isis# info
----------------------------------------------
        shutdown
        level-capability level-1
        area-id 49.0180.0001
        ...
A:node-A>config>router>isis# level-capability level-1/2
*A:node-A>config>router>isis# info
----------------------------------------------
        shutdown
        area-id 49.0180.0001
        ...
```

4.6.4.6.4 Configure LFA for IS-IS interfaces

Configure LFA policy maps for interfaces using a route next-hop template

Use the following commands to apply a route next-hop policy template that is configured with the required policies, to all IS-IS prefixes whose primary next hop uses a specific interface. See Application of route next hop policy template to an interface for more information.

- **MD-CLI**

```
configure router isis interface loopfree-alternate policy-map route-nh-template
configure service vprn isis interface loopfree-alternate policy-map route-nh-template
```

- **classic CLI**

```
configure router isis interface lfa-policy-map route-nh-template
configure service vprn isis interface lfa-policy-map route-nh-template
```

Exclude interfaces from LFA SPF

Use the following commands to exclude an interface in IS-IS or an IS-IS level from an LFA SPF. You can also exclude prefixes from a prefix policy that matches on prefixes or on IS-IS tags, for the router or VPRN service. See Excluding interfaces and prefixes from LFA SPF for more information.

- **MD-CLI**

```
configure router isis interface loopfree-alternate exclude
configure service vprn isis interface loopfree-alternate exclude
```

- **classic CLI**

```
configure router isis interface loopfree-alternate-exclude
configure service vprn isis interface loopfree-alternate-exclude
```

4.6.4.7 Configuring IS-IS link groups

IS-IS Link-Groups allows the creation of an administrative grouping of multiple IS-IS member interfaces that should be treated as a common group for ECMP purposes. If the number of operational links in the link-group drops below the operational-member value, then all links associated with that IS-IS link group will have their interface metric increased by the configured offset amounts. As a result, IS-IS will then try to reroute traffic over lower cost paths.

After it is triggered, the higher metric will not be reset to the originally configured IS-IS interface metric values until the number of active interfaces in the link bundle reaches the configured revertive threshold (**revert-members**).

Prerequisite are the following:

- one or more interface members
- a configured operational-member (**oper-members**) value
- a configured revertive-member (**revert-members**) value
- configured offset values for the appropriate address families

4.7 IS-IS configuration management tasks

This section discusses IS-IS configuration management tasks.

4.7.1 Disabling IS-IS

When you administratively disable the IS-IS protocol instance on the router, the configuration settings are not changed, reset, or removed. Use the following command to disable IS-IS on a router:

- **MD-CLI**

```
configure router isis admin-state disable
```

- **classic CLI**

```
configure router isis shutdown
```

4.7.2 Removing IS-IS

When you remove the IS-IS protocol instance, the IS-IS configuration reverts to the default settings.

Use the following command to remove the IS-IS configuration:

- **MD-CLI**

```
config router delete isis
```

- **classic CLI**

```
configure router no isis
```

4.7.3 Modifying global IS-IS configuration

You can modify, disable, or remove global IS-IS configuration without administratively disabling the entities. Changes take effect immediately. Modifying the level capability on the global level causes the IS-IS protocol to restart.

4.7.4 Modifying IS-IS interface configuration

You can modify, disable, or remove interface-level IS-IS configuration without administratively disabling the entities. Changes take effect immediately. Modifying the level capability on the interface causes the IS-IS protocol on the interface to restart.

Use the following command to remove an interface:

- **MD-CLI**

```
configure router delete interface
```

- **classic CLI**

```
configure router no interface
```

Use the following command to disable an interface:

- **MD-CLI**

```
configure router interface admin-state disable
```

- **classic CLI**

```
configure router interface shutdown
```

The following example displays interface IS-IS modification command usage. For specific interface configuration and modification examples also see, Configuring a level 1 area and Modifying a router's level capability.

4.7.5 Configuring authentication using keychains

The use of authentication mechanism is recommended to protect against malicious attack on the communications between routing protocol neighbors. These attacks could aim to either disrupt communications or to inject incorrect routing information into the systems routing table. The use of authentication keys can help to protect the routing protocols from these types of attacks. In addition, the

use of authentication keychains provides the ability to configure authentication keys and make changes to them without affecting the state of the routing protocol adjacencies.

To configure the use of an authentication keychain within IS-IS, use the following steps.

1. Use the following command to configure an authentication keychain. The configured keychain must include at least one valid key entry, using a valid authentication algorithm for the IS-IS protocol.
   - **MD-CLI**

     ```
     configure system security keychains
     ```

   - **classic CLI**

     ```
     configure system security keychain
     ```

2. Associate the configure authentication keychain with IS-IS. Authentication keychains can be used to specify the authentication at the IS-IS global and level contexts, as well as for hello authentication at the interface and interface-level context.
   - **MD-CLI**

     ```
     configure router isis hello-authentication-keychain
     ```

   - **classic CLI**

     ```
     configure router isis auth-keychain
     ```

Use the following command to establish the hello authentication association:

- **MD-CLI**

  ```
  configure router isis hello-authentication
  ```

- **classic CLI**

  ```
  configure router isis hello-auth-keychain
  ```

For a key entry to be valid, it must include a valid key, the current system clock value must be within the begin and end time of the key entry, and the algorithm specified in the key entry must be supported by the IS-IS protocol.

The IS-IS protocol supports the following algorithms:

- clear text password (RFC 5304 and RFC 5310 formats)
- HMAC-MD5 (RFC 5304 and RFC 5310 formats)
- HMAC-SHA-1 (RFC 5310 format)
- HMAC-SHA-256 (RFC 5310 format)

The IS-IS key entry may also include the option to determine how the IS-IS protocol encodes the authentication signature. The**basic** command option results in the use of RFC 5304 format. The default or a value of the **isis-enhanced** command option results in using the RFC 5310 format.

Use the following command to configure the encoding of the IS-IS protocol authentication signature:

- **MD-CLI**

```
configure system security keychains keychain bidirectional entry option
```

- **classic CLI**

```
configure system security keychain direction bi entry
```

The error handling is described below.

- If a keychain exists but there are no active key entries with an authentication type that is valid for the associated protocol then inbound protocol packets are not authenticated and discarded and no outbound protocol packets should be sent.
- If keychain exists, but the last key entry has expired, a log entry is raised indicating that all keychain entries have expired. The IS-IS protocol requires that the protocol not revert to an unauthenticated state and requires that the old key is not to be used, therefore, after the last key has expired, all traffic is discarded.

4.7.6 Guidelines for configuring route leaking from level 2 to level 1 areas

IS-IS allows a two-level hierarchy to route PDUs. Level 1 areas can be interconnected by a contiguous level 2 backbone. The level 1 link-state database contains information only about that area. The level 2 link-state database contains information about the level 2 system and each of the level 1 systems in the area. A level 1/2 router contains information about both level 1 and level 2 databases. A level 1/2 router advertises information about its level 1 area toward the other level 1/2 or level 2 (only) routers.

Packets with destinations outside the level 1 area are forwarded toward the closest level 1/2 router which, in turn, forwards the packets to the destination area.

Sometimes, the shortest path to an outside destination is not through the closest level 1/2 router, or, the only level 1/2 system to forward packets out of an area is not operational. Route leaking provides a mechanism to leak level 2 information to level 1 systems to provide routing information about inter-area routes. A level 1 router then has more options to forward packets.

See Configuring route leaking from level 2 to level 1 areas for information about how to configure route leaking.

4.7.7 Configuring route leaking from level 2 to level 1 areas

This topic describes how to configure policies to leak routes from level 2 to level 1.

Prerequisites

- □ Review the guidelines in Guidelines for configuring route leaking from level 2 to level 1 areas.
- □ In the classic CLI, when configuring policy-options you must use the **begin** command to enter edit mode and the **commit** command to save the configuration.

About this task

This task describes how to configure policies, including a prefix list and policy statement, to leak routes from level 2 to level 1 areas.

Procedure

Step 1. Use the commands in the following contexts to configure a policy to leak routes from level 2 into level 1 areas:

- **MD-CLI**

  Note: Use the **commit** command to save the configuration.

- **classic CLI**

  Note: Use the **begin** command to enter edit mode for the policy-options configuration and the **commit** command to save the configuration.

Example

MD-CLI

```
[ex:/configure policy-options]
A:admin@node-2# info
    prefix-list "loops" {
        prefix 10.1.1.0/24 type longer {
        }
    }
    policy-statement "leak" {
        entry 10 {
            from {
                level 2
                prefix-list "loops"
            }
            to {
                level 1
            }
            action {
                action-type accept
            }
        }
    }
```

Example

classic CLI

```
A:node-2>config>router>policy-options# info
----------------------------------------------
            prefix-list "loops"
                    prefix 10.1.1.0/24 longer
            exit
            policy-statement "leak"
                entry 10
                    from
                        prefix-list "loops"
                        level 2
                    exit
                    to
                        level 1
                    exit
                    action accept
                    exit
                exit
```

```
            exit
-----------------------------------------------
```

Step 2. Use the following command to apply the policy statement created in step 1 to leak routes from level 2 info level 1 systems on the node.

- **MD-CLI**

```
configure router isis export-policy
```

- **class CLI**

```
configure router isis export
```

Example

MD-CLI

```
[ex:/configure router "Base" isis 0]
A:admin@node-A# info
    authentication-key "auth-key= hash2"
    authentication-type password
    ...
    area-address [49.0180.0001 49.0180.0002 49.0180.0003]
    authentication-check false
    export-policy "leak"
    ...
```

Example

classic CLI

```
A:node-A>config>router>isis# info
----------------------------------------------
        area-id 49.0180.0001
        area-id 49.0180.0002
        area-id 49.0180.0003
        authentication-key "//auth-key" hash
        authentication-type password
        no authentication-check
        export "leak"
        ...
----------------------------------------------
```

Step 3. After exporting the policy, use the commands in the following context to create a policy to redistribute external IS-IS routes from level 1 systems into the level 2 backbone (see Redistributing external IS-IS routers).

- **MD-CLI**

```
configure policy-options prefix-list
configure policy-options policy-statement entry
```

- **classic CLI**

```
configure router policy-options prefix-list
configure router policy-options policy-statement entry
```

Example

MD-CLI

```
[ex:/configure policy-options]
A:node-AC# info
    prefix-list "loops" {
        prefix 10.1.1.0/24 type longer {
        }
    }
    policy-statement "leak" {
        entry 10 {
            from {
                level 2
                prefix-list ["loops"]
            }
            to {
                level 1
            }
            action {
                action-type accept
            }
        }
    }
    policy-statement "isis-ext" {
        entry 10 {
            from {
                external true
            }
            to {
                level 2
            }
            action {
                action-type accept
            }
        }
    }
```

Example

classic CLI

```
A:node-AC>config>router>policy-options# info
----------------------------------------------
            prefix-list "loops"
                    prefix 10.1.1.0/24 longer
            exit
            policy-statement "leak"
                entry 10
                    from
                        prefix-list "loop"
                        level 2
                    exit
                    to
                        level 1
                    exit
                    action accept
                    exit
                exit
            exit
            policy-statement "isis-ext"
                entry 10
                    from
```

```
                        external
                    exit
                    to
                        level 2
                    exit
                    action accept
                    exit
                exit
            exit
-----------------------------------------------
```

4.7.8 Redistributing external IS-IS routers

IS-IS does not redistribute level 1 external routes into level 2 by default. You must explicitly apply the policy to redistribute external IS-IS routes. Use the commands in the following context to create and apply a policy to redistribute external IS-IS routes:

- **MD-CLI**

  Note: Use the **commit** command to save the configuration.

```
configure policy-options prefix-list
configure policy-options policy-statement entry
```

- **class CLI**

  Note: Use the **begin** command to enter edit mode for the policy-options configuration and the **commit** command to save the configuration.

```
configure router policy-options prefix-list
configure router policy-options policy-statement entry
```

See Route policies for more information about creating and using route policies.

The following example shows a policy-statement configuration.

Example: MD-CLI

```
[ex:/configure policy-options]
A:node-AC# info
    prefix-list "loops" {
        prefix 10.1.1.0/24 type longer {
        }
    }
    policy-statement "isis-ext" {
        entry 10 {
            from {
                external true
            }
            to {
                level 2
            }
            action {
                action-type accept
            }
        }
    }
```

```
    policy-statement "leak" {
        entry 10 {
            from {
                level 2
                prefix-list ["loops"]
            }
            to {
                level 1
            }
            action {
                action-type accept
            }
        }
    }
```

Example: classic CLI

```
A:node-AC>config>router>policy-options# info
----------------------------------------------
            prefix-list "loops"
                    prefix 10.1.1.0/24 longer
            exit
            policy-statement "leak"
                entry 10
                    from
                        prefix-list "loops"
                        level 2
                    exit
                    to
                        level 1
                    exit
                    action accept
                    exit
                exit
            exit
            policy-statement "isis-ext"
                entry 10
                    from
                        external
                    exit
                    to
                        level 2
                    exit
                    action accept
                    exit
                exit
            exit
----------------------------------------------
```

4.7.9 Specifying MAC addresses for all IS-IS routers

Use the following commands to specify a MAC address for all L1 IS-IS routers.

```
configure router isis all-l1isis
configure service vprn isis all-l1isis
```

Use the following commands to specify the MAC address for all L2 IS-IS routers.

```
configure router isis all-l2isis
```

```
configure service vprn isis all-l2isis
```

5 BGP

5.1 BGP overview

Border Gateway Protocol (BGP) is an inter-Autonomous System routing protocol. An Autonomous System (AS) is a set of routers managed and controlled by a common technical administration. BGP-speaking routers establish BGP sessions with other BGP-speaking routers and use these sessions to exchange BGP routes. A BGP route provides information about a network path that can reach an IP prefix or other type of destination. The path information in a BGP route includes the list of ASes that must be traversed to reach the route source; this allows inter-AS routing loops to be detected and avoided. Other path attributes that may be associated with a BGP route include the Local Preference, Origin, Next-Hop, Multi-Exit Discriminator (MED) and Communities. These path attributes can be used to implement complex routing policies.

The primary use of BGP was originally Internet IPv4 routing but multiprotocol extensions to BGP have greatly expanded its applicability. Now BGP is used for many purposes, including:

- Internet IPv6 routing
- inter-domain multicast support
- L3 VPN signaling (unicast and multicast)
- L2 VPN signaling (BGP autodiscovery for LDP-VPLS, BGP-VPLS, BGP-VPWS, multisegment pseudowire routing, EVPN)
- setup of inter-AS MPLS LSPs
- distribution of flow specification rules (filters/ACLs)

The next sections provide information about BGP sessions, BGP network design, BGP messages and BGP path attributes.

5.2 BGP sessions

A BGP session is a TCP connection formed between two BGP routers over which BGP messages are exchanged. The three types of BGP sessions are as follows:

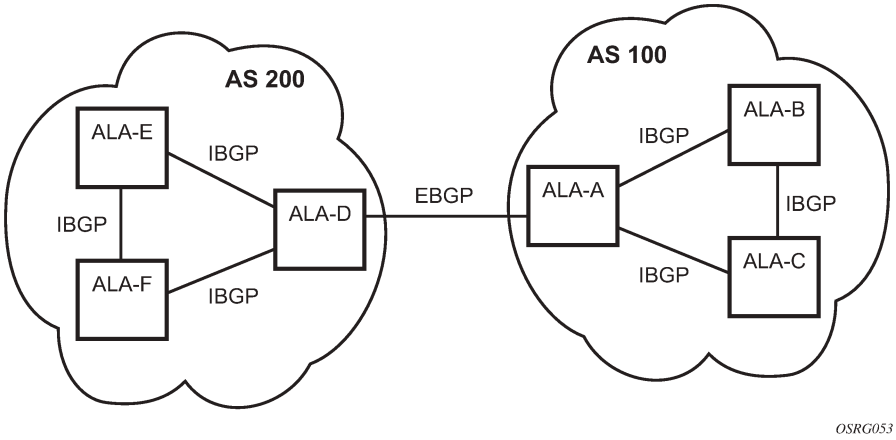
- internal BGP (IBGP)
- external BGP (EBGP)
- confederation external BGP (confed-EBGP)

An IBGP session is formed when the two BGP routers belong to the same Autonomous System (AS). Routes received from an IBGP peer are not advertised to other IBGP peers unless the router is a route reflector. The two routers that form an IBGP session are usually not directly connected. Figure 19: BGP sessions shows an example of two Autonomous Systems that use BGP to exchange routes. In this example, the router ALA-A forms IBGP sessions with ALA-B and ALA-C.

An EBGP session is formed when the two BGP routers belong to different Autonomous Systems. Routes received from an EBGP peer can be advertised to any other peer. The two routers that form an EBGP

session are often directly connected but multihop EBGP sessions are also possible. When a route is advertised to an EBGP peer the Autonomous System numbers of the advertising router are added to the AS Path attribute. In the example of Figure 19: BGP sessions, the router ALA-A forms an EBGP session with ALA-D.

Figure 19: BGP sessions



A confederation EBGP session is formed when the two BGP routers belong to different member Autonomous Systems of the same confederation. See BGP confederations for more information about BGP confederations.

SR OS supports both statically configured and dynamic (unconfigured) BGP sessions. Dynamic sessions are allowed when they are detected by either of the following mechanisms:

- The source IP address of an incoming BGP TCP connection matches an IP prefix associated with dynamic BGP sessions. Use the following command to configure this prefix.

```
configure router bgp group dynamic-neighbor match prefix
```

- An ICMPv6 router advertisement message is received from a potential BGP router on an interface listed as a dynamic-neighbor interface. Use the following command to configure an interface for dynamic neighbors.

```
configure router bgp group dynamic-neighbor interface
```

Use the following command to configure a statically configured BGP session:

- **MD-CLI**

```
configure router bgp neighbor
```

- **classic CLI**

```
configure router bgp group neighbor
```

This command accepts either an IPv4 or IPv6 address, which allows the session transport to be IPv4 or IPv6. By default, the router is the active side of TCP connections to statically configured remote peers, meaning that as soon as a session leaves the Idle state, the router attempts to set up an outgoing TCP connection to the remote neighbor in addition to listening on TCP port 179 for an incoming connection from

the peer. If required, a statically configured BGP session can be configured for passive mode so that the router only listens for an incoming connection and does not attempt to set up the outgoing connection.

The source IP address used to set up the TCP connection to the statically configured or dynamic peer can be configured explicitly. Use the following command to configure the source IP address at the group level.

```
configure router bgp group local-address
```

Use the following command to configure the source IP address at the neighbor level:

- **MD-CLI**

  ```
  configure router bgp neighbor local-address
  ```

- **classic CLI**

  ```
  configure router bgp group neighbor local-address
  ```

If the **local-address** command is not configured, the source IP address is determined as follows:

- If the neighbor's IP address belongs to a local subnet, the source IP address is this router's IP address on that subnet.
- If the neighbor's IP address does not belong to a local subnet, the source IP address is this router's system IP address.

In addition, it is possible to configure the local address with the name of the router interface. To configure the BGP local address to use the router interface's IP address information, the **local-address** command is used in conjunction with the configured router interface name. Configuring the router interface as the local address is available at both the group level and the neighbor level.

When the router interface is configured as the local address, BGP inherits the address from the interface as follows:

- **BGPv4 sessions**

  The primary IPv4 address configured on the interface is used as the local address.

- **BGPv6 sessions**

  The primary IPv6 address configured on the interface is used as the local address.

If the corresponding IPv4 or IPv6 address is not configured on the router interface, the BGP sessions that have this interface set as the local address are kept down until an interface address is configured on the router interface.

If the primary IPv4 or IPv6 address is changed on the router interface and that interface is being used as the local address for BGP, then BGP bounces the link. This removes all routes advertised using the previous address and starts advertising those routes again using the newly configured IP address.

5.2.1 BGP session states

A BGP session is in one of the following states at any moment in time:

- **Idle**

  This is the state of a BGP session when it is administratively disabled. In this state no incoming TCP connection is accepted from the peer. When the session is administratively enabled it transitions out of the idle state immediately. When the session is restarted automatically it may not leave the idle state

immediately if **damp-peer-oscillations** is configured, **damp-peer-oscillations** holds a session in the idle state for exponentially increasing amounts of time if the session is unstable and resets frequently.

- **Connect**

  This is the state of a BGP session when the router, acting in active mode, is attempting to establish an outbound TCP connection with the remote peer.

- **Active**

  This is the state of a BGP session when the router is listening for an inbound TCP connection attempt from the remote peer.

- **OpenSent**

  This is the state of a BGP session when the router has sent an OPEN message to its peer in reaction to successful setup of the TCP connection and is waiting for an OPEN message from the peer.

- **OpenConfirm**

  This is the state of a BGP session after the router has received an acceptable OPEN message from the peer and sent a KEEPALIVE message in response and is waiting for a KEEPALIVE message from the peer. TCP connection collision procedures may be performed at this stage. For more details, see the RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*.

- **Established**

  This is the state of a BGP session after the router has received a KEEPALIVE message from the peer. In this state BGP can advertise and withdraw routes by sending UPDATE messages to its peer.

5.2.2 Detecting BGP session failures

If a router suspects that its peer at the other end of an established session has experienced a complete failure of both its control and data planes the router should divert traffic away from the failed peer as quickly as possible to minimize traffic loss. There are various mechanisms that the router can use to detect such failures, including:

- BGP session hold timer expiry (for more details about this mechanism, see Keepalive message)
- peer tracking
- BFD
- fast external failover

When any one or these mechanisms is triggered the session immediately returns to the *Idle* state and a new session is attempted. Peer tracking, BFD and fast external failover are described in more detail in the following sections.

5.2.2.1 Peer tracking

When peer tracking is enabled on a session, the neighbor IP address is tracked in the routing table. If a failure occurs and there is no longer any IP route matching the neighbor address, or if the longest prefix match (LPM) route is rejected by the peer-tracking policy (configurable for the BGP router or VPRN service using the **peer-tracking-policy** command), after a 1-second delay the session is taken down. By default, peer-tracking is disabled on all sessions. The default peer-tracking policy allows any type of route to match the neighbor IP address, except aggregate routes and LDP shortcut routes.

Peer tracking was introduced when BFD was not yet supported for peer failure detection. Now that BFD is available, peer-tracking has less value and is used less often.

Note: Use peer tracking with caution. Peer tracking can tear a session down even if the loss of connectivity turns out to be short-lived. For example, while the IGP protocol is re-converging. Next-hop tracking, which is always enabled, handles such temporary connectivity issues much more effectively.

5.2.2.2 Bidirectional Forwarding Detection

SR OS supports setting up an asynchronous-mode BFD session to a BGP neighbor, so that the failure of the BFD session triggers an immediate teardown of the BGP session. When BFD is enabled on a BGP session, a one-hop or multihop BFD session is set up using the neighbor IP address, and the BFD settings are derived from the BFD configuration of the interface associated with the address. For multihop sessions, this is usually the system interface. With a 10 ms transit interval and a multiplier of 3, BFD can detect peer failure in as quickly as 30 ms.

By default, BGP only reacts to BFD sessions transitioning from up to down and does not wait for the BFD session to come back up to reestablish the affected BGP session or to consider the address of the peer (as a BGP next hop) reachable. However, when BFD Strict-Mode is active, that is, both sides have advertised the associated BGP capability and BFD is configured appropriately on each side, BGP blocks the BGP session from transitioning to established until the BFD session to the peer is up.

Strict-BFD cannot be used to control BGP session-state changes, but the user can use the **next-hop-reachability** command to configure the BGP next-hop resolution process to keep a next hop unresolved for as long as the associated BFD session stays down, and not to recover automatically based on route or tunnel reachability. When the **next-hop-reachability** command is enabled, routes received from one peer with a BGP next-hop address equal to the address of another peer are not affected by the BFD session to the other peer. The **next-hop-reachability** command only affects routes belonging to the following address families:

- IPv4
- IPv6
- IPv4 VPN
- IPv6 VPN
- labeled unicast IPv4
- labeled unicast IPv6
- EVPN
- IPv4 multicast
- IPv6 multicast
- IPv4 VPN multicast
- IPv6 VPN multicast

5.2.2.3 Fast external failover

Fast external failover applies only to single-hop EBGP sessions. When fast external failover is enabled on a single-hop EBGP session and the interface associated with the session goes down the BGP session is

immediately taken down as well, even if other mechanisms such as the hold-timer have not yet indicated a failure.

5.2.3 High availability BGP sessions

A BGP session reset can be very disruptive – each router participating in the failed session must delete the routes it received from its peer, recalculate new best paths, update forwarding tables (depending on the types of routes), and send route withdrawals and advertisements to other peers. It makes sense then that session resets should be avoided as much as possible and when a session reset cannot be avoided the disruption to the network should be minimized.

To support these objectives, the BGP implementation in SR OS supports two key features:

- BGP high availability (HA)
- BGP graceful restart (GR)

BGP HA refers to the capability of a router with redundant CPMs to keep established BGP sessions up whenever a planned or unplanned CPM switchover occurs. A planned CPM switchover can occur during In-Service Software Upgrade (ISSU). An unplanned CPM switchover can occur if there is an unexpected failure of the primary CPM.

BGP HA is always enabled on routers with redundant CPMs; it cannot be disabled. BGP HA keeps the standby CPM in-sync with the primary CPM, with respect to BGP and associated TCP state, so that the standby CPM is ready to take over for the primary CPM at any time. The primary CPM is responsible for building and sending the BGP messages to peers but the standby CPM reliably receives a copy of all outgoing UPDATE messages so that it has a synchronized view of the RIB-OUT.

5.2.3.1 BGP graceful restart

Some BGP routers do not have redundant control plane processor modules or do not support BGP HA with the same quality or coverage as 7450 ESS, 7750 SR, or 7950 XRS routes. When dealing with such routers or specific error conditions, BGP graceful restart (GR) is a good option for minimizing the network disruption caused by a control plane reset.

BGP GR assumes that the router restarting its BGP sessions has the ability and architecture to continue packet forwarding throughout the control plane reset. If this is the case, then the peers of the restarting router act as helpers and "hide" the control plane reset from the rest of the network so that forwarding can continue uninterrupted. Forwarding based on stale routes and hiding the "staleness" from other routers is considered acceptable because the duration of the control plane outage is expected to be relatively short (a few minutes). For BGP GR to be used on a session, both routers must advertise the BGP GR capability during the OPEN message exchange; see the BGP advertisement section for more details.

BGP GR is enabled on one or more BGP sessions by configuring the **graceful-restart** command in the global, group, or neighbor context. The command causes GR mode to be supported for the following active families:

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

- label-IPv6
- L2-VPN
- route-target (RTC)
- flow-IPv4 (IPv4 FlowSpec)
- flow-IPv6 (IPv6 FlowSpec)

Helper mode is activated when one of the following events affects an 'Established' session:

- TCP socket error
- new inbound TCP connection from the peer
- hold timer expiry
- peer unreachable
- BFD down
- sent or received NOTIFICATION messages, only if notifications are enabled using the following commands, and the peer set the "N" bit in its GR capability, and the NOTIFICATION is not a 'Cease' with subcode 'Hard Reset':
  - **MD-CLI**

    ```
    configure router bgp graceful-restart gr-notification
    configure router bgp group graceful-restart gr-notification
    configure router bgp neighbor graceful-restart gr-notification
    ```

  - **classic CLI**

    ```
    configure router bgp graceful-restart enable-notification
    configure router bgp group graceful-restart enable-notification
    configure router bgp group neighbor graceful-restart enable-notification
    ```

As soon as the failure is detected, the helping 7450 ESS, 7750 SR, or 7950 XRS router marks all the routes received from the peer as stale and starts a restart timer. The stale state is not factored into the BGP decision process, and it is not made visible to other routers in the network. The restart timer derives its initial value from the Restart Time carried in the last GR capability of the peer. The default advertised Restart Time is 300 seconds, but it can be changed using the **restart-time** command.

When the restart timer expires, helping stops if the session is not yet re-established. If the session is re-established before the restart timer expires and the new GR capability from the restarting router indicates that the forwarding state has been preserved, then helping continues and the peers exchange routes per the normal procedure.

When each router has advertised all its routes for a specific address family, it sends an End-of-RIB marker (EOR) for the address family. The EOR is a minimal UPDATE message with no reachable or unreachable Network Layer Reachability Information (NLRI) for the AFI or SAFI. When the helping router receives an EOR, it deletes all remaining stale routes of the AFI or SAFI that were not refreshed in the most recent set of UPDATE messages. The maximum amount of time that routes can remain stale (before being deleted if they are not refreshed) is configurable using the **stale-routes-time**.

Note: If a second reset occurs before GR has successfully completed, the router always aborts the GR helper process, regardless of the failure trigger.

5.2.3.2 BGP long-lived graceful restart

SR OS supports Long-Lived Graceful Restart (LLGR). LLGR is supported for the same address families as normal GR, as described in BGP graceful restart.

The LLGR procedures adhere to *draft-uttaro-idr-bgp-persistence-03*. LLGR is intended to handle more serious and longer-term outages than ordinary GR.

SR OS routers support LLGR in the context of both the restarting router (which experienced a restart or failure) and the helper or receiving router (which is a peer of the failed router). Both functionalities are enabled and disabled at the same time by adding the **long-lived** command under a **graceful-restart** configuration context.

When **long-lived** is applied to a session (and capability negotiation is not disabled), the OPEN message sent to the peer includes both the GR capability and the LLGR capability. Both capabilities list the same set of AFI/SAFI.

5.2.3.2.1 LLGR operations

If a BGP session protected by LLGR goes down because of a restart or failure of the peer, then the SR OS router activates GR+LLGR helper mode for all the protected AFI/SAFI. In GR+LLGR helper mode, the received routes of a particular AFI/SAFI are retained as stale routes for a maximum duration of:

restart-time + *LLGR-stale-time*

where:

restart-time is signaled in the GR capability of the peer (but overridden, if necessary, by the locally-configured **helper-override-restart-time** command).

LLGR-stale-time is signaled in the LLGR capability of the peer (but overridden, if necessary, by a locally-configured **helper-override-stale-time** command).

While the restart-timer is running, the SR OS router acts in the normal GR helper role. When the restart-timer elapses, the LLGR phase begins. When LLGR starts, the following tasks occur.

1. The *LLGR-stale-time* starts to count down.
2. Stale routes marked with the NO_LLGR community are immediately deleted.
3. Remaining stale routes are not preferred. The BGP best path selection algorithm is rerun with a new first step that prefers valid, non-stale LLGR routes over any stale LLGR routes.
4. If a de-preferenced stale route remains, the best and valid NH-reachable path for the NLRI is re-advertised, with an added LLGR_STALE community, to peers that signaled support for the LLGR capability. The route may be withdrawn or re-advertised toward peers that do not support LLGR, subject to the configuration of the **advertise-stale-to-all-neighbors** command and **without-no-export** command option.

LLGR ends for a particular AFI/SAFI when the LLGR-stale-time reaches zero. At that time, all remaining stale routes of the AFI/SAFI are deleted. The LLGR-stale-time is not stopped by re-establishment of the session with the failed peer; it continues until the EoR marker is received for the AFI/SAFI.

Stale routes may be deleted before the expiration of the LLGR-stale-time. If the session with the failed peer comes back up and one of the following is true, then the stale routes should be deleted immediately.

- The GR or LLGR capability is missing.
- The AFI/SAFI is missing from the LLGR capability.
- The F bit is equal to 0 for the AFI/SAFI.

5.2.3.2.2 Receiving routes with LLGR_STALE community

When a router running SR OS Release 15.0.R4 or later receives a BGP route of any AFI/SAFI, with the LLGR_STALE community, the decision process considers the route less preferred than any valid, non-stale LLGR route for that NLRI. This logic applies even if the router is not configured as long-lived. If a route with an LLGR_STALE community is selected as the best path, then it is advertised to peers according to the configuration of the **advertise-stale-to-all-neighbors** command; if this command is absent (or the **long-lived** context is absent), then the route is advertised only to peers that advertised the LLGR capability.

5.2.4 BGP session security

5.2.4.1 TCP MD5 authentication

The operation of a network can be compromised if an unauthorized system is able to form or hijack a BGP session and inject control packets by falsely representing itself as a valid neighbor. This risk can be mitigated by enabling TCP MD5 authentication on one or more of the sessions. When TCP MD5 authentication is enabled on a session every TCP segment exchanged with the peer includes a TCP option (19) containing a 16-byte MD5 digest of the segment (more specifically the TCP/IP pseudo-header, TCP header and TCP data). The MD5 digest is generated and validated using an authentication key that must be known to both sides. If the received digest value is different from the locally computed one then the TCP segment is dropped, thereby protecting the router from spoofed TCP segments.

5.2.4.2 TTL security mechanism

The TTL security mechanism (GTSM) relies on a simple concept to protect BGP infrastructure from spoofed IP packets. It recognizes the fact that the vast majority of EBGP sessions are established between directly-connected routers and therefore the IP TTL values in packets belonging to these sessions should have predictable values. If an incoming packet does not have the expected IP TTL value it is possible that it is coming from an unauthorized and potentially harmful source.

TTL security is enabled using the **ttl-security** command. This command requires a minimum TTL value to be specified. When TTL security is enabled on a BGP session the IP TTL values in packets that are supposedly coming from the peer are compared (in hardware) to the configured minimum value and if there is a discrepancy the packet is discarded and a log is generated. TTL security is used most often on single-hop EBGP sessions but it can be used on multihop EBGP and IBGP sessions as well.

To enable TTL security on a single-hop EBGP session, configure **ttl-security** and **multihop** to a value of 255. To enable TTL security on a multihop EBGP session, configure **ttl-security** and **multihop** to match the expected TTL of (255 - hop count). The TTL value for both EBGP peers must be manually configured to the same value, as there is no TTL negotiation.

Note: IP packets sent to an IBGP peer are originated with an IP TTL value of 64. IP packets to an EBGP peer are originated with an IP TTL value of 1, except if **multihop** is configured; in that case, the TTL value is taken from the **multihop** command.

5.2.5 BGP address family support for different session types

When the base router has a neighbor identified by an IPv4 address, and therefore the transport of the BGP session uses IPv4 TCP, all MP-BGP address families available in SR OS are supported by that session.

When the base router has a neighbor identified by an IPv6 address, and therefore the transport of the BGP session uses IPv6 TCP, the following MP-BGP address families are supported:

- ipv4
- ipv6
- mcast-ipv4
- mcast-ipv6
- vpn-ipv4
- vpn-ipv6
- evpn
- flow-ipv6
- label-ipv4
- label-ipv6
- bgp-ls

When a VPRN has a neighbor identified by an IPv4 address, and therefore the transport is IPv4 TCP, the following MP-BGP address families are supported:

- ipv4
- ipv6
- mcast-ipv4
- mcast-ipv6
- flow-ipv4
- flow-ipv6
- label-ipv4

When a VPRN has a neighbor identified by an IPv6 address, and therefore the transport is IPv6 TCP, the following MP-BGP address families are supported:

- ipv4
- ipv6
- mcast-ipv4
- mcast-ipv6
- flow-ipv6

5.2.6 BGP groups

In SR OS, every neighbor (and therefore BGP session) is configured under a group. A group is a CLI construct that saves configuration effort when multiple peers have a similar configuration; in this situation the common configuration commands can be configured once at the group level and need not be repeated for every neighbor. A single BGP instance can support many groups and each group can support many

peers. Most SR OS commands that are available at the **neighbor** level are also available at the **group** level.

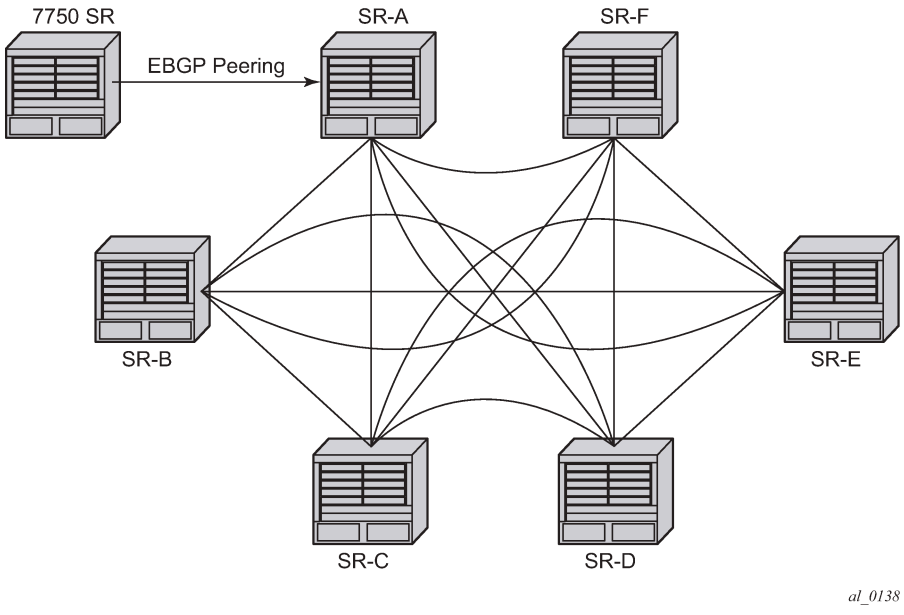
5.3 BGP design concepts

BGP assumes that all routers within an Autonomous System can reach destinations external to the Autonomous System using efficient, loop-free intra-AS forwarding paths. This generally requires that all the routers within the AS have a consistent view of the best path to every external destination. This is especially true when each BGP router in the AS makes its own forwarding decisions based on its own BGP routing table. The basic BGP specification does not store any intra-AS path information in the AS Path attribute so basic BGP has no way to detect routing loops within an AS that arise from inconsistent best path selections.

There are 3 solutions for dealing the issues described above.

- Create a full-mesh of IBGP sessions within the AS as shown in Figure 20: Fully meshed BGP configuration. This ensures routing consistency but does not scale well because the number of sessions increases exponentially with the number of BGP routers in the AS.
- Use BGP route reflectors in the AS. Route reflection is described in the section titled Route reflection. BGP route reflectors allow for routing consistency with only a partial mesh of IBGP sessions within the AS.
- Create a confederation of autonomous systems. BGP confederations are described in the section titled BGP confederations.

Figure 20: Fully meshed BGP configuration



5.3.1 Route reflection

In a standard BGP configuration a BGP route learned from one IBGP peer is not re-advertised to another IBGP peer. This rule exists because of the assumption of a full IBGP mesh within the AS. As discussed in the previous section a full IBGP mesh imposes specific scaling challenges. BGP route reflection eliminates the need for a full IBGP mesh by allowing routers configured as route reflectors to re-advertise routes from one IBGP peer to another IBGP peer.

A route reflector provides route reflection service to IBGP peers called clients. Other IBGP peers of the RR are called non-clients. An RR and its client peers form a cluster. A large AS can be sub-divided into multiple clusters, each identified by a unique 32-bit cluster ID. Each cluster contains at least one route reflector which is responsible for redistributing routes to its clients. The clients within a cluster do not need to maintain a full IBGP mesh between each other; they only require IBGP sessions to the route reflectors in their cluster. If the clients within a cluster are fully meshed, consider using the following commands to disable client reflection of routes. The non-clients in an AS must be fully meshed with each other.

- **MD-CLI**

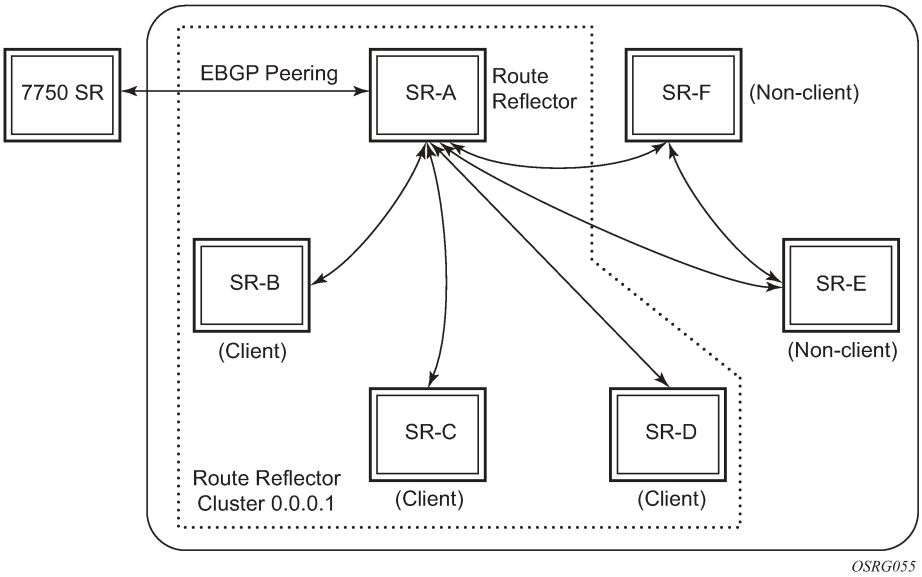
```
configure router bgp client-reflect
configure router bgp neighbor client-reflect
configure router bgp group client-reflect
```

- **classic CLI**

```
configure router bgp disable-client-reflect
configure router bgp group neighbor disable-client-reflect
configure router bgp group disable-client-reflect
```

Figure 21: BGP configuration with route reflectors depicts the same network as Figure 20: Fully meshed BGP configuration but with route reflectors deployed to eliminate the IBGP mesh between SR-B, SR-C, and SR-D. SR-A, configured as the route reflector, is responsible for reflection routes to its clients SR-B, SR-C, and SR-D. SR-E and SR-F are non-clients of the route reflector. As a result, a full mesh of IBGP sessions must be maintained between SR-A, SR-E and SR-F.

Figure 21: BGP configuration with route reflectors



A router becomes a route reflector whenever it has one or more client IBGP sessions. Use the BGP **cluster** command to create a client IBGP session, which also indicates the cluster ID of the client. The typical practice is to use the router ID as the cluster ID, although this is not necessary.

Basic route reflection operation (without Add-Path configured) is summarized as follows.

- If the best and valid path for an NLRI is learned from a client and client reflection of routes is not disabled, advertise that route to all clients, non-clients, and EBGP peers (as allowed by policy). If the client that advertised the best and valid path is a neighbor to which the **split-horizon** command (at the BGP group or neighbor level) applies, the route is not advertised back to the sending client. In the route that is reflected to clients and non-clients, the route reflector does the following:
  - adds an ORIGINATOR_ID attribute if it did not already exist; the ORIGINATOR_ID indicates the BGP identifier (router ID) of the client that originated the route
  - prepends the cluster ID of the client that advertised the route and then the cluster ID of the client receiving the route (if applicable) to the CLUSTER_LIST attribute, creating the attribute if it did not previously exist
- If the best and valid path for an NLRI is learned from a client and client reflection of routes is disabled, advertise that route to all clients in other clusters, non-clients, and EBGP peers (as allowed by policy). In the route that is reflected to clients in other clusters and non-clients, the router reflector does the following:
  - adds an ORIGINATOR_ID attribute if it did not already exist; the ORIGINATOR_ID indicates the BGP identifier (router ID) of the client that originated the route
  - prepends the cluster ID of the client receiving the route to the CLUSTER_LIST attribute, creating the attribute, if it did not previously exist
- If the best and valid path for an NLRI is learned from a non-client, advertise that route to all clients and EBGP peers (as allowed by policy). In the route that is reflected to clients, the router reflector does the following:

  - adds an ORIGINATOR_ID attribute if it did not already exist; the ORIGINATOR_ID indicates the BGP identifier (router ID) of the non-client that originated the route
  - prepends the cluster ID of the client receiving the route to the CLUSTER_LIST attribute, creating the attribute if it did not previously exist
- If the best and valid path for an NLRI is learned from an EBGP peer, advertise that route to all clients, non-clients and other EBGP peers (as allowed by policy). The ORIGINATOR_ID and CLIUSTER_LIST attributes are not added to the route.
- If the best and valid path for an NLRI is locally originated by the RR (for example, it was learned through means other than BGP), it advertises that route to all clients, non-clients and EBGP peers (as allowed by policy). The ORIGINATOR_ID and CLUSTER_LIST attributes are not added to the route.

The ORIGINATOR_ID and CLUSTER_LIST attributes allow BGP to detect the looping of a route within the AS. If any router receives a BGP route with an ORIGINATOR_ID attribute containing its own BGP identifier, the route is considered invalid. In addition, if a route reflector receives a BGP route with a CLUSTER_LIST attribute containing a locally configured cluster ID, the route is considered invalid. Invalid routes are not installed in the route table and not advertised to other BGP peers.

5.3.2 BGP confederations

BGP confederations are another alternative for avoiding a full mesh of BGP sessions inside an Autonomous System (AS). A BGP confederation is a group of ASs managed by a single technical administration that appear as a single AS to BGP routers outside the confederation. The single externally visible AS is called the confederation ID. Each AS in the group is called a member AS and the ASN of each member AS is visible only within the confederation. For this reason, member ASNs are often private ASNs.

Within a confederation EBGP-type, sessions can be setup between BGP routers in different member ASs. These confederation-EBGP sessions avoid the need for a full mesh between routers in different member ASs. Within each member AS the BGP routers must be fully-meshed with IBGP sessions or route reflectors must be used to ensure routing consistency.

In SR OS, a confederation EBGP session is formed when the ASN of the peer is different from the local ASN and the peer ASN appears as a member AS in the **configure router confederation** context. You can configure the confederation ID and up to 15 members that are part of a confederation.

When a route is advertised to a confederation-EBGP peer the advertising router prepends its local ASN, which is its member ASN, to a confederation-specific sub-element in the AS_PATH that is created if it does not already exist. The extensions to the AS_PATH are used for loop detection but they do not influence best path selection (that is, they do not increase the AS Path length used in the BGP decision process). The MED, NEXT_HOP and LOCAL_PREF attributes in the received route are propagated unchanged by default. The ORIGINATOR_ID and CLUSTER_LIST attributes are not included in routes to confederation-EBGP peers.

When a route is advertised to an EBGP peer outside the confederation the advertising router removes all member AS elements from the AS_PATH and prepends its confederation ID instead of its local/member ASN.

5.4 BGP messages

BGP protocol operation relies on the exchange of BGP messages between peers. 7450 ESS, 7750 SR, 7950 XRS, and most other routers, support the following message types: Open message, Update message, Keepalive message, Notification message, and Route refresh message.

The minimum BGP message length is 19 bytes and the maximum is 4096 bytes. BGP messages appear as a stream of bytes to the underlying TCP transport layer, and so there is no direct association between a BGP message and a TCP segment. One TCP segment can include parts of one or more BGP messages. Immediately after session setup, the initial value for the maximum TCP segment size that can be sent toward a specific peer is the minimum of the following:

- the MSS option value in the TCP SYN received from the peer, when the connection was established
- the IP MTU of the initial outgoing interface used to route packets to the peer, minus 40 bytes (IPv4) or minus 60 bytes (IPv6)
- the TCP MSS value in the BGP configuration of the peer, if there is a configuration and it specifies a value instead of the **ip-stack** option

As time elapses, the maximum sending segment size can fall below the initial value if path MTU discovery (PMTUD) is active on the session. PMTUD lowers the segment size when ICMP unreachable or packet-too-big messages are received. These messages indicate that the IP MTU of the link could not forward the unfragmentable packet and this IP MTU minus 40 (IPv4) or minus 60 (IPv6) bytes sets the new maximum segment size value.

5.4.1 Open message

After a TCP connection is established between two BGP routers the first message sent by each one is an Open message. If the received Open message is acceptable a Keepalive message confirming the Open is sent back. For more information, see BGP session states.

An Open message contains the following information:

- **Version**

  The current BGP version number is 4.

- **Autonomous system number**

  The 2-byte AS of the sending router. If the sending router has an ASN greater than 65535, this field has the special value 23456 (AS_TRANS). On a 7450, 7750, or 7950 router, the ASN in the Open message is based on the confederation ID (if the peer is external to the confederation), the global AS (configured using the **autonomous-system** command) or a session-level override of the global AS called the local AS (configured using the **local-as** command). More details about the use of local-AS are described in the section titled Using local AS for ASN migration. More details about 4-byte AS numbers are described in the section titled 4-octet autonomous system numbers.

- **Hold time**

  The proposed maximum time BGP waits between successive messages (Keepalive and/or Update) from its peer before closing the connection. The actual hold time is the minimum of the configured **hold-time** for the session and the hold-time in the peer's Open message. If this minimum is below the threshold configured with the following command option, the connection attempt is rejected:

– **MD-CLI**

```
configure router bgp neighbor hold-time minimum-hold-time
```

– **classic CLI**

```
configure service vprn bgp hold-time min
```

Note: Changes to the configured **hold-time** trigger a session reset.

- **BGP identifier**

  The router ID of the BGP speaker. In Open messages, the BGP ID is derived from following:

  1. The router ID configured under BGP.
  2. If the router ID is not configured under BGP, the BGP ID comes from the configuration under the **configure router** or **configure service vprn** context.
  3. If the router ID is not configured for the router or service VPRN, the system interface IPv4 address is used.

  Note: A change of the router ID in the **configure router bgp** context causes all BGP sessions to be reset immediately while other changes resulting in a new BGP identifier only take effect after BGP is shutdown and re-enabled.

- **Options**

  Each of the option capabilities is TLV-encoded with a unique type code. The only optional capability that has been defined is the optional type. The optional type supports the process of BGP advertisement (see BGP advertisement for more information). The BGP router terminates the session if it receives an Open message with an unsupported optional type. Unless you disable the capability negotiation using the following command, the router always sends an optional type in its Open message:

  – **MD-CLI**

  ```
  configure router bgp group capability-negotiation false
  configure router bgp neighbor capability-negotiation false
  ```

  – **classic CLI**

  ```
  configure router bgp group disable-capability-negotiation
  configure router bgp group neighbor disable-capability-negotiation
  ```

5.4.1.1 Changing the autonomous system number

If the AS number is changed at the router level (**configure router**), the new AS number is not used until the BGP instance is restarted either by administratively disabling and enabling the BGP instance or by rebooting the system with the new configuration.

On the other hand, if the AS number is changed in the BGP configuration (**configure router bgp**), the effects are as follows.

- A change of the local-AS at the global level causes the BGP instance to restart with the new local AS number.
- A change of the local-AS at the **group** level causes BGP to re-establish sessions with all peers in the group using the new local AS number.
- A change of the local-AS at the **neighbor** level causes BGP to re-establish the session with the new local AS number.

5.4.1.2 Changing a confederation number

Changing the confederation value on an active BGP instance does not restart the protocol. The change takes effect when the BGP protocol is (re) initialized.

5.4.1.3 BGP advertisement

BGP advertisement allows a BGP router to indicate to a peer, using the optional type, the features that it supports so that they can coordinate and use only the features that both support. Each capability is TLV-encoded with a unique optional type code. SR OS supports the following capability codes:

- multiprotocol BGP (code 1)
- route refresh (code 2)
- outbound route filtering (code 3)
- graceful restart (code 64)
- 4-octet AS number (code 65)
- add-path (code 69)

5.4.2 Update message

Update messages are used to advertise and withdraw routes. An update message provides the following information:

- **Withdrawn routes length**

  The length of the withdrawn routes field that is described next (may be 0).
- **Withdrawn routes**

  The IPv4 prefixes that are no longer considered reachable by the advertising router.
- **Total path attribute length**

  The length of the path attributes field that is discussed next (may be 0).
- **Path attributes**

  The path attributes presented in variable length TLV format. The path attributes apply to all the NLRI in the UPDATE message.
- **Network layer reachability information (NLRI)**

  The IPv4 prefixes that are considered reachable by the advertising router.

For fast routing convergence, as many NLRI as possible are packed into a single Update message as possible. This requires identifying all the routes that share the same path attribute values.

5.4.3 Keepalive message

After a session is established, each router sends periodic Keepalive messages to its peer to test that the peer is still alive and reachable. If no Keepalive or Update message is received from the peer for the negotiated hold-time duration, the session is terminated. The period between one Keepalive message and the next is 1/3 of the negotiated hold-time duration or the value configured with the **keepalive** command, whichever is less. If the active hold-time or keepalive interval is zero, Keepalive messages are not sent. The default hold-time is 90 seconds and the default keepalive interval is 30 seconds.

A peer (reachability) failure is often detected through faster mechanisms than hold-timer expiry, as described in Detecting BGP session failures.

5.4.4 Notification message

When a non-recoverable error related to a particular session occurs a Notification message is sent to the peer and the session is terminated (or restarted if GR is enabled for this scenario). For more details, see BGP graceful restart. The Notification message provides the following information:

- **Error code**

  This indicates the type of error: message header error, Open message error, Update message error, Hold timer expired, Finite State Machine error, or Cease.

- **Error subcode**

  This provides more specific information about the error. The meaning of the subcode is specific to the error code.

5.4.4.1 Update message error handling

The approach to handling UPDATE message errors has evolved. The original BGP protocol specification called for all UPDATE message errors to be handled the same way, by sending a notification to the peer and immediately closing the BGP session. This error handling approach was motivated by the goal to ensure protocol "correctness" above all else. But it ignored several important points, including:

- Not all UPDATE message errors truly have the same severity. If the NLRI cannot be extracted and parsed from an UPDATE message then this is indeed a "critical" error. But other errors such as incorrect attribute flag settings, missing mandatory path attributes, incorrect next-hop length/format, and so on. can be considered "non-critical" and handled differently.
- Session resets are extremely costly in terms of their impact on the stability and performance of the network. For many types of UPDATE message errors, a session reset does not solve the problem because the root cause remains (for example, software error, hardware error or misconfiguration). If a session reset is absolutely necessary, then the operator should have some control over the timing.
- Some degree of protocol "incorrectness" is tolerable for a short period of time as long as the network operator is fully aware of the issue. In this context, "incorrectness" typically means a BGP RIB inconsistency between routers in the same AS. Such inconsistency has become less and less of an issue over time as edge-to-edge tunneling of IP traffic (for example, BGP shortcuts, IP VPN) has reduced the number of deployments where IP traffic is forwarded hop-by-hop.

In recognition of these points, the IETF's IDR working group documented a revised set of error handling procedures in RFC 7606, and gradually all BGP implementations have moved toward the implementation of this RFC. All peers, regardless of **update-fault-tolerance** configuration, are automatically covered by

the procedures documented in RFC 7606. If absolutely necessary, it is possible to disable the RFC 7606 procedures on specific BGP sessions by configuring **legacy-mode** to **true** and removing the **update-fault-tolerance** configuration on these sessions. However, this is not recommended for most situations.

```
configure router bgp error-handling legacy-mode
configure service vprn bgp error-handling legacy-mode
```

If **legacy-mode** is set to **true**, the **update-fault-tolerance** command allows the user to decide whether the router should apply RFC 7606 or legacy error-handling procedures to UPDATE message errors. If the **update-fault-tolerance** command is configured, noncritical errors are handled using the "treat-as-withdraw" or "attribute-discard" approach to error handling. These approaches do not cause a session reset. If the **update-fault-tolerance** command is not configured, legacy procedures apply and all errors (critical and non critical) trigger a session reset.

If the **legacy-mode** is configured to **true**, and the **update-fault-tolerance** configuration is removed from a BGP session, the BGP session is reset if a non critical error was already encountered.

5.4.5 Route refresh message

A BGP router can send a Route Refresh message to its peer only if both have advertised the route refresh capability (code 2). The Route Refresh message is a request for the peer to re-send all or some of its routes associated with a particular pair of AFI/SAFI values. AFI/SAFI values are the same ones used in the MP-BGP capability (see the section titled Multiprotocol BGP attributes).

7450, 7750, and 7950 routers only send Route Refresh messages for AFI/SAFI associated with VPN routes that carry Route Target (RT) Extended Communities, such as VPN-IPv4, VPN-IPv6, L2-VPN, MVPN-IPv4 and MVPN-IPv6 routes. By default, routes of these types are discarded if, at the time they are received, there is no VPN that imports any of the route targets they carry. If at a later time a VPN is added or reconfigured (in terms of the route targets that it imports), a Route Refresh message is sent to all relevant peers, so that previously discarded routes can be relearned.

Note: Route refresh messages are not sent for VPN-IPv4 and VPN-IPv6 routes if **mp-bgp-keep** is configured; in this situation received VPN-IP routes are kept in the RIB-IN regardless of whether they match a VRF import policy or not.

5.5 BGP path attributes

Path attributes are fundamental to BGP. A BGP route for a particular NLRI is distinguished from other BGP routes for the same NLRI by its set of path attributes. Each path attribute describes some property of the path and is encoded as a TLV in the Path Attributes field of the Update message. The type field of the TLV identifies the path attribute and the value field carries data specific to the attribute type. There are 4 different categories of path attributes:

- **Well-known mandatory**

  These attributes must be recognized by all BGP routers and must be present in every update message that advertises reachable NLRI toward a specific type of neighbor (EBGP or IBGP).

- **Well-known discretionary**

  These attributes must be recognized by all BGP routers but are not required in every update message.

- **Optional transitive**

These attributes are allowed to be unrecognized by some BGP routers. If a BGP router does not recognize one of these attributes it accepts it, passes it on to other BGP peers, and sets the Partial bit to 1 in the attribute flags byte.

- **Optional non-transitive**

  These attributes are allowed to be unrecognized by some BGP routers. If a BGP router does not recognize one of these attributes it is quietly ignored and not passed on to other BGP peers.

SR OS supports the following path attributes, which are described in detail in upcoming sections:

- ORIGIN (well-known mandatory)
- AS_PATH (well-known mandatory)
- NEXT_HOP (well-known, required only in Update messages with IPv4 prefixes in the NLRI field)
- MED (optional non-transitive)
- LOCAL_PREF (well-known, required only in Update messages sent to IBGP peers)
- ATOMIC_AGGR (well-known discretionary)
- AGGREGATOR (optional transitive)
- COMMUNITY (optional transitive)
- ORIGINATOR_ID (optional non-transitive)
- CLUSTER_LIST (optional non-transitive)
- MP_REACH_NLRI (optional non-transitive)
- MP_UNREACH_NLRI (optional non-transitive)
- EXT_COMMUNITY (optional transitive)
- AS4_PATH (optional transitive)
- AS4_AGGREGATOR (optional transitive)
- CONNECTOR (optional transitive)
- PMSI_TUNNEL (optional transitive)
- TUNNEL_ENCAPSULATION (optional transitive)
- AIGP (optional non-transitive)
- BGP-LS (optional non-transitive)
- LARGE_COMMUNITY (optional transitive)

5.5.1 Origin

The ORIGN path attribute indicates the origin of the path information. There are three supported values:

- IGP (0)
- EGP (1)
- Incomplete (2)

When a router originates a VPN-IP prefix (from a non-BGP route), it sets the value of the origin attribute to IGP. When a router originates an BGP route for an IP prefix by exporting a non-BGP route from the routing

table, it sets the value of the origin attribute to Incomplete. Route policies (BGP import and export) can be used to change the origin value.

5.5.2 AS path

The AS_PATH attribute provides the list of Autonomous Systems through which the routing information has passed. The AS_PATH attribute is composed of segments. There can be up to 4 different types of segments in an AS_PATH attribute: AS_SET, AS_SEQUENCE, AS_CONFED_SET and AS_CONFED_SEQUENCE. The AS_SET and AS_CONFED_SET segment types result from route aggregation. AS_CONFED_SEQUENCE contains an ordered list of member AS through which the route has passed inside a confederation. AS_SEQUENCE contains an ordered list of AS (including confederation IDs) through which the route has passed on its way to the local AS/confederation.

The AS numbers in the AS_PATH attribute are all 2-byte values or all 4-byte values (if the 4-octet ASN capability was announced by both peers).

A BGP router always prepends its AS number to the AS_PATH attribute when advertising a route to an EBGP peer. The specific details for a 7450, 7750, or 7950 router are described below.

- When a route is advertised to an EBGP peer and the advertising router is not part of a confederation.
  - The global AS (configured using the **autonomous-system** command) is prepended to the AS_PATH if **local-as** is not configured.
  - The local AS followed by the global AS are prepended to the AS_PATH if **local-as** is configured.
  - Only the local AS (not the global AS) is prepended to the AS_PATH if the following command is configured:
    - **MD-CLI**

      ```
      configure router bgp local-as prepend-global-as false
      ```

    - **classic CLI**

      ```
      configure router bgp local-as no-prepend-global-as
      ```

  - Some or all private and reserved AS numbers (64512 to 65535 and 4200000000 to 4294967295 inclusive) can be removed or replaced from the AS_PATH if the **remove-private** command is configured.
- When a route is advertised to an EBGP peer outside a confederation.
  - The confederation ID is prepended to the AS_PATH if **local-as** is not configured.
  - The local AS followed by the confederation ID are prepended to the AS_PATH if **local-as** is configured (the command option to not prepend the global AS has no effect in this scenario).
  - Member AS numbers are removed from the AS_PATH as described in the section titled BGP confederations.
  - Some or all private and reserved AS numbers (64512 to 65535 and 4200000000 to 4294967295 inclusive) can be removed or replaced from the AS_PATH if the **remove-private** command is configured.
- When a route is advertised to a confederation-EBGP peer.

  - If the route came from an EBGP peer and **local-as** was configured on this session (without the **private** option) this local AS number is prepended to the AS_PATH in a regular AS_SEQUENCE segment.
  - The global AS (configured using the **autonomous-system** command) is prepended, as a member AS, to the AS_PATH if **local-as** is not configured.
  - The local AS followed by the global AS are prepended, as member AS, to the AS_PATH if **local-as** is configured.
  - Only the local AS is prepended, as a member AS, to the AS_PATH if **local-as no-prepend-global-as** is configured.
  - Some or all private and reserved AS numbers (64512 to 65535 and 4200000000 to 4294967295 inclusive) can be removed or replaced from the AS_PATH if the **remove-private** command is configured.
- When a route is advertised to an IBGP peer.
  - No information is added to the AS_PATH if the route is locally originated or if it came from an IBGP peer.
  - The local AS number is prepended to the AS_PATH if the route came from an EBGP peer and **local-as** is configured without the **private** option.
  - The local AS number is prepended, as a member AS, to the AS_PATH if the route came from a confederation-EBGP peer and **local-as** is configured without the **private** option.
  - Some or all private and reserved AS numbers (64512 to 65535 and 4200000000 to 4294967295 inclusive) can be removed or replaced from the AS_PATH if the **remove-private** command is configured.

BGP import policies can be used to prepend an AS number multiple times to the AS_PATH, whether the route is received from an IBGP, EBGP or confederation EBGP peer. The AS path prepend action is also supported in BGP export policies applied to these types of peers, regardless of whether the route is locally originated or not. AS path prepending in export policies occurs before the global and/or local ASs (if applicable) are added to the AS_PATH.

When a BGP router receives a route containing one of its own autonomous system numbers (local or global or confederation ID) in the AS_PATH the route is normally considered invalid for reason of an AS path loop. However, SR OS provides a **loop-detect** command that allows this check to be bypassed. If it known that advertising specific routes to an EBGP peer results in an AS path loop condition and yet there is no loop (assured by other mechanisms, such as the Site of Origin (SOO) extended community), then **as-override** can be configured on the advertising router instead of disabling loop detection on the receiving router. The **as-override** command replaces all occurrences of the peer AS in the AS_PATH with the advertising router's local AS.

5.5.2.1 AS override

The AS override feature can be used in VPRN scenarios where a customer is running BGP as the PE-CE protocol and some or all of the CE locations are in the same Autonomous System (AS). With normal BGP, two sites in the same AS would not be able to reach each other directly because there is an apparent loop in the AS path.

When the AS override option is configured on a PE-CE EBGP session, the PE rewrites the customer ASN in the AS path with the VPRN AS number as the route is advertised to the CE.

5.5.2.2 Using local AS for ASN migration

The description in the previous section does fully describe the reasons for using **local-as**. This BGP feature facilitates the process of changing the ASN of all the routers in a network from one number to another. This may be necessary if one network operator merges with or acquires another network operator and the two BGP networks must be consolidated into one autonomous system.

For example, suppose the operator of the ASN 64500 network merges with the operator of the ASN 64501 network and the new merged entity decides to renumber ASN 64501 routers as ASN 64500 routers, so that the entire network can be managed as one autonomous system. The migration can be carried out using the following sequence of steps.

1. Change the global AS of the route reflectors that used to be part of ASN 64501 to the new value 64500.
2. Change the global AS of the RR clients that used to be part of ASN 64501 to the new value 64500.
3. Configure the following command on every EBGP session of each RR client migrated in step 2:
   - **MD-CLI**

     ```
     configure router bgp local-as as-number 64501
     configure router bgp local-as prepend-global-as false
     configure router bgp local-as private true
     ```

   - **classic CLI**

     ```
     configure router bgp local-as 64501 private no-prepend-global-as
     ```

This migration procedure has several advantages. First, customers, settlement-free peers and transit providers of the previous ASN 64501 network still perceive that they are peering with ASN 64501 and can delay switching to ASN 64500 until the time is convenient for them. Second, the AS path lengths of the routes exchanged with the EBGP peers are unchanged from before so that best path selections are preserved.

5.5.2.3 4-octet autonomous system numbers

When BGP was developed, it was assumed that 16-bit (2-octet) ASNs would be sufficient for global Internet routing. In theory a 16-bit ASN allows for 65536 unique autonomous systems but some of the values are reserved (0 and 64000-65535). Of the assignable space less than 10% remains available. When a new AS number is needed it is now simpler to obtain a 4-octet AS number. 4-octet AS numbers have been available since 2006. A 32-bit (4-octet) ASN allows for 4,294,967,296 unique values (some of which are again, reserved).

When 4-octet AS numbers became available it was recognized that not all routers would immediately support the ability to parse 4-octet AS numbers in BGP messages so two optional transitive attributes called AS4_PATH and AS4_AGGREGATOR were introduced to allow a gradual migration.

A BGP router that supports 4-octet AS numbers advertises this capability in its OPEN message. The capability information includes the AS number of the sending BGP router, encoded using 4 bytes (recall the ASN field in the OPEN message is limited to 2 bytes). By default, OPEN messages sent by 7450 ESS, 7750 SR, or 7950 XRS routers always include the 4-octet ASN capability. You can change this using the following command:

- **MD-CLI**

```
configure router bgp asn-4-byte false
```

- **classic CLI**

```
configure router bgp disable-4byte-asn
```

If a BGP router and its peer have both announced the 4-octet ASN capability, then the AS numbers in the AS_PATH and AGGREGATOR attributes are always encoded as 4-byte values in the UPDATE messages they send to each other. These UPDATE messages should not contain the AS4_PATH and AS4_AGGREGATOR path attributes.

If one of the routers involved in a session announces the 4-octet ASN capability and the other one does not, then the AS numbers in the AS_PATH and AGGREGATOR attributes are encoded as 2-byte values in the UPDATE messages they send to each other.

When a 7450 ESS, 7750 SR, or 7950 XRS router advertises a route to a peer that did not announce the 4-octet ASN capability.

- If there are any AS numbers in the AS_PATH attribute that cannot be represented using 2 bytes (because they have a value greater than 65535) they are substituted with the special value 23456 (AS_TRANS) and an AS4_PATH attribute is added to the route if it is not already present. The AS4_PATH attribute has the same encoding as the AS_PATH attribute that would be sent to a 4-octet ASN capable router (that is, each AS number is encoded using 4 octets) but it does not carry segments of type AS_CONFED_SEQUENCE or AS_CONFED_SET.
- If the AS number in the AGGREGATOR attribute cannot be represented using 2 bytes (because its value is greater than 65535) it is substituted with the special value 23456 and as AS4_AGGREGATOR attribute is added to the route if it is not already present. The AS4_AGGREGATOR is the same as the AGGREGATOR attribute that would be sent to a 4-octet ASN capable router (that is, the AS number is encoded using 4 octets).

When a 7450 ESS, 7750 SR, or 7950 XRS router receives a route with an AS4_PATH attribute it attempts to reconstruct the full AS path from the AS4_PATH and AS_PATH attributes, regardless of whether the 4byte ASN is disabled or not. The reconstructed path is the AS path displayed in BGP show commands. If the length of the received AS4_PATH is N and the length of the received AS_PATH is N+t, then the reconstructed AS path contains the t leading elements of the AS_PATH followed by all the elements in the AS4_PATH.

5.5.3 Next-hop

The NEXT_HOP attribute indicates the IPv4 address of the BGP router that is the next-hop to reach the IPv4 prefixes in the NLRI field. If the Update message is advertising routes other than IPv4 unicast routes the next-hop of these routes is encoded in the MP_REACH_NLRI attribute. For more details, see Multiprotocol BGP attributes.

The rules for deciding what next-hop address types to accept in a received BGP route and what next-hop address types to advertise as a BGP next-hop are address family dependent. The following sections summarize the key details.

5.5.3.1 Unlabeled IPv4 unicast routes

By default, IPv4 routes are advertised with IPv4 next-hops but on IPv6-TCP transport sessions they can be advertised with IPv6 next-hops if the **advertise-ipv6-next-hops** command (with the IPv4 option) applies to the session. To receive IPv4 routes with IPv6 next-hop addresses from a peer, the **extended-nh-encoding** command (with the IPv4 option) must be applied to the session. This advertises the corresponding RFC 5549, *Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop*, capability to the peer.

Whenever **next-hop-self** applies to an IPv4 route, the next hop is set as follows.

- If the peer whose routes are being advertised is an IPv4 transport peer (in other words, the neighbor address is IPv4), the BGP next-hop is the IPv4 local address used to setup the session.
- If the peer toward which the routes are being advertised is an IPv6 transport peer (in other words, the neighbor address is IPv6), and the **advertise-ipv6-next-hops** command (with the **ipv4** option) applies to the session, and the peer toward which the routes are being advertised opened the session by announcing an Extended NH Encoding capability for AFI =1, SAFI = 1 and next-hop AFI = 2, then the BGP next-hop is the IPv6 local address used to setup the session. Otherwise, for all other cases, the BGP next-hop is the IPv4 address of the system interface. If the system interface does not have an IPv4 address, the route is not advertised unless an export policy sets a valid IPv4 next-hop.

When an IPv4 BGP route is advertised to an EBGP peer, **next-hop-self** always applies except if the **third-party-nexthop** command is applied. Configuring **third-party-nexthop** allows an IPv4 route received from one EBGP peer to be advertised to another EBGP that is in the same IP subnet with an unchanged BGP next-hop.

When an IPv4 BGP route is re-advertised to an IBGP or confederation EBGP peer, the advertising router does not modify the BGP next-hop unless one of the following applies.

- The BGP **next-hop-self** command is applied to the IBGP or confederation EBGP peer. This causes **next-hop-self** to be applied to all IPv4 routes advertised to the peer, regardless of the peer type from which they were received (IBGP, confederation-EBGP, or EBGP).
- IPv4 routes are matched and accepted by a route policy entry, and this entry has a **next-hop-self** action. This applies **next-hop-self** as described above to only those routes matched by the policy entry.
- IPv4 routes are matched and accepted by route policy entry, and this entry has a next-hop IP address action. This changes the BGP next-hop of only the matched routes to the IP address, if the IP address is an IPv4 address or an IPv6 address and the necessary conditions exist. The **advertise-ipv6-next-hops** command is configured appropriately and the peer opened the session with the correct RFC 5549 capability.

When an IPv4 BGP route is locally originated and advertised to an IBGP or confederation EBGP peer, the BGP next-hop is, by default, copied from the next hop of the route that was imported into BGP, with specified exceptions (for example, black-hole next-hop). When a static route with indirect next hop is re-advertised as a BGP route, the BGP next-hop is a copy of the indirect address. However, with route table import policies, BGP can be instructed to take the resolved next hop of the static route as the BGP next-hop address.

5.5.3.2 Unlabeled IPv6 unicast routes

SR OS routers never send or receive IPv6 routes with 32-bit IPv4 next-hop addresses.

When an IPv6 BGP route is advertised to an EBGP peer, next-hop-self always applies (except if the third-party-nexthop command is applied, as described in the following note). Next-hop-self results in one of the following outcomes.

- If the EBGP session uses IPv4 transport, the BGP next-hop encodes the local-address used for setup of the session as an IPv4-compatible IPv6 address (all zeros in the first 96 bits followed by the 32 bit IPv4 local-address).
- If the EBGP session uses IPv6 transport, the BGP next-hop is the local-address used to setup the session and this cannot be overridden, even by BGP export policy.

Note: Configuring **third-party-nexthop** allows an IPv6 route received from one EBGP peer to be advertised to another EBGP that is in the same IP subnet with an unchanged BGP next-hop.

When an IPv6 BGP route is re-advertised to an IBGP or confederation-EBGP peer, the advertising router does not modify the BGP next-hop by default; however, this can be changed as follows.

- If the BGP **next-hop-self** command is applied to the IBGP peer or confederation-EBGP peer, then this changes the BGP next-hop to the local-address used to setup the session (if the transport to the peer is IPv6) or to an IPv4-compatible IPv6 address derived from the IPv4 local-address used to setup the session (if the transport to the peer is IPv4). This command applies to all routes advertised to the peer, regardless of the peer type from which they were received (IBGP, confed-EBGP, or EBGP).
- If IPv6 routes are matched and accepted by an export policy applied to an IBGP or confederation-EBGP session, and the matching policy entry has a **next-hop-self** action, this changes the BGP next-hop of only the matched routes to the local-address used to setup the session (if the transport to the peer is IPv6) or to an IPv4-compatible IPv6 address derived from the IPv4 local-address used to setup the session (if the transport to the peer is IPv4).
- If IPv6 routes are matched and accepted by an export policy applied to an IBGP or confederation-EBGP session, and the matching policy entry has a **next-hop** IP address action, this changes the BGP next-hop of only the matched routes to the IP address, if the IP address is an IPv6 address. If the IP address is an IPv4 address the matched routes are treated as though they were rejected by the policy entry.

When an IPv6 BGP route is locally originated and advertised to an IBGP or confederation- EBGP peer, the BGP next-hop is, by default, copied from the next-hop of the route that was imported into BGP, with specified exceptions (for example, black-hole next-hop). When a static route with indirect next-hop is re-advertised as a BGP route, the BGP next-hop is a copy of the indirect address, however with route-table-import policies BGP can be instructed to take the resolved next-hop of the static route as the BGP next-hop address.

5.5.3.3 VPN-IPv4 routes

SR OS routers can send and receive VPN-IPv4 routes with IPv4 next-hops. They can also be configured (using the **extended-nh-encoding** command) to receive VPN-IPv4 routes with IPv6 next-hop addresses from selected BGP peers by signaling the corresponding Extended NH Encoding BGP capability to those peers during session setup. If the capability is not advertised to a peer, such routes are not accepted from that peer. Also, if the SR OS router does not receive an Extended NH Encoding capability advertisement for [NLRI AFI=1, NLRI SAFI=128, next-hop AFI=2] from a peer, it does not advertise VPN-IPv4 routes with IPv6 next-hops to that peer.

When a VPN-IPv4 BGP route is advertised to an EBGP peer, the **next-hop-self** command applies in the following cases:

- The following command is configured:
  - **MD-CLI**

```
configure router bgp inter-as-vpn
```

  - **classic CLI**

```
configure router bgp enable-inter-as-vpn
```

- The following command and the **next-hop-self** command are configured toward the EBGP peer:
  - **MD-CLI**

```
configure router bgp subconfed-vpn-forwarding
```

  - **classic CLI**

```
configure router bgp enable-subconfed-vpn-forwarding
```

Otherwise, there is no change to the next-hop and the **next-hop-self** command results in one of the following outcomes:

- If the EBGP session uses IPv4 transport, the BGP next-hop is taken from the value of the local address used to set up the session.
- If the EBGP peer has opened an IPv6-transport session by advertising an extended NH encoding capability with (NLRI AFI=1, NLRI SAFI=128, next-hop AFI=2) and, in the configuration of the local router, the session is associated with an **advertise-ipv6-next-hops vpn-ipv4** command, then the BGP next-hop is set to the value of the IPv6 local address used to set up the session. Otherwise, the BGP next-hop is set to the IPv4 address of the system interface.

If inter-AS-VPN is enabled, the **next-hop-self** command applies automatically if a VPN-IPv4 BGP route is received from an EBGP peer and readvertised to an IBGP or confederation-EBGP peer. If the inter-AS-VPN is not enabled, but subconfederation VPN forwarding is enabled, the **next-hop-self** command can be set manually toward any IBGP or confederation-EBGP peer with the same label-swap forwarding behavior as provided by the inter-AS-VPN command. In either case, the **next-hop-self** command results in one of the following outcomes.

- If the IBGP or confederation-EBGP session uses IPv4 transport, then the BGP next-hop is taken from the value of the local address used to set up the session.
- If the IBGP or confederation-EBGP peer has opened an IPv6-transport session by advertising an extended NH encoding capability with (NLRI AFI=1, NLRI SAFI=128, next-hop AFI=2) and, in the configuration of the local router, the session is associated with an **advertise-ipv6-next-hops vpn-ipv4** command, then the BGP next-hop is set to the value of the IPv6 local address used to set up the session. Otherwise, the BGP next-hop is set to the IPv4 address of the system interface.

When a VPN-IPv4 BGP route is reflected from one IBGP peer to another IBGP peer, the RR does not modify the next-hop by default. However, if the **next-hop-self** command is applied to the IBGP peer receiving the route and the following command is enabled, a next-hop-self and label swap behavior can be provided:

- **MD-CLI**

```
configure router bgp rr-vpn-forwarding
```

- **classic CLI**

```
configure router bgp enable-rr-vpn-forwarding
```

Alternatively, the same behavior can be achieved by configuring the following command and setting the **next-hop-self** command toward the targeted IBGP peer.

- **MD-CLI**

```
configure router bgp subconfed-vpn-forwarding
```

- **classic CLI**

```
enable-subconfed-vpn-forwarding
```

In either case, next-hop-self results in one of the following outcomes:

- If the IBGP session receiving the reflected route uses IPv4 transport, the BGP next-hop is taken from the value of the local address used to set up the session.
- If the IBGP session receiving the reflected route has opened an IPv6-transport session by advertising an extended NH encoding capability with (NLRI AFI=1, NLRI SAFI=128, next-hop AFI=2) and, in the configuration of the local router, the session is associated with an **advertise-ipv6-next-hops vpn-ipv4** command, then the BGP next-hop is set to the value of the IPv6 local address used to set up the session. Otherwise, the BGP next-hop is set to the IPv4 address of the system interface.

When a VPN-IPv4 BGP route is reflected from one IBGP peer to another IBGP peer, the following command is configured, and the VPN-IPv4 route is matched and accepted by an export policy entry with a **next-hop** IP address action, the BGP next-hop of the matched routes changes to the **next-hop** IP address:

- **MD-CLI**

```
configure router bgp rr-vpn-forwarding
```

- **classic CLI**

```
configure router bgp enable-rr-vpn-forwarding
```

The exception to this is if the **next-hop** IP address is an IPv6 address and the receiving IBGP peer did not advertise an extended NH encoding capability with (NLRI AFI=1, NLRI SAFI=128, next-hop AFI=2), or in the configuration of the local router, the session is not associated with an **advertise-ipv6-next-hops vpn-ipv4** command. In this case, the route is treated as though it was rejected by the policy entry.

5.5.3.4 VPN-IPv6 routes

SR OS routers never send or receive VPN-IPv6 routes with 32-bit IPv4 next-hop addresses.

When a VPN-IPv6 BGP route is advertised to an EBGP peer, the **next-hop-self** command applies in the following cases:

- The following command is configured:
  - **MD-CLI**

```
configure router bgp inter-as-vpn
```

  - **classic CLI**

```
configure router bgp enable-inter-as-vpn
```

- The following command and the **next-hop-self** command are configured toward the EBGP peer:

– **MD-CLI**

```
configure router bgp subconfed-vpn-forwarding
```

– **classic CLI**

```
configure router bgp enable-subconfed-vpn-forwarding
```

Otherwise, there is no change to the next-hop. The **next-hop-self** command results in one of the following outcomes:

- If the EBGP session uses IPv4 transport, then the BGP next-hop is set to the IPv4 local address used to set up the session but encoded as an IPv4-mapped IPv6 address (for example, with the IPv4 address in the least significant 32 bits of a ::FFFF/96 prefix).
- If the EBGP session uses IPv6 transport and it is associated with an **advertise-ipv6-next-hops vpn-ipv6** command, the BGP next-hop is set to the value of the IPv6 local address used to set up the session. Otherwise, the BGP next-hop is set to the IPv4 address of the system interface encoded as an IPv4-mapped IPv6 address (for example, with the IPv4 address in the least significant 32 bits of a ::FFFF/96 prefix).

If inter-AS-VPN is enabled, the **next-hop-self** command applies automatically if a VPN-IPv6 BGP route is received from an EBGP peer and re-advertised to an IBGP or confederation-EBGP peer. If inter-AS-VPN is not enabled, but the subconfederation VPN forwarding is enabled, the next-hop can be set manually toward any IBGP or confederation EBGP peer, with the same label-swap forwarding behavior as provided by the inter-AS-VPN configuration. In either case, the **next-hop-self** command results in one of the following outcomes.

- If the IBGP or confederation EBGP session uses IPv4 transport, then the BGP next-hop is set to the IPv4 local address used to set up the session but encoded as an IPv4-mapped IPv6 address (for example, with the IPv4 address in the least significant 32 bits of a ::FFFF/96 prefix).
- If the IBGP or confederation EBGP session uses IPv6 transport and it is associated with an **advertise-ipv6-next-hops vpn-ipv6** command, the BGP next-hop is set to the value of the IPv6 local address used to set up the session. Otherwise, the BGP next-hop is set to the IPv4 address of the system interface encoded as an IPv4-mapped IPv6 address (for example, with the IPv4 address in the least significant 32 bits of a ::FFFF/96 prefix).

When a VPN-IPv6 BGP route is reflected from one IBGP peer to another IBGP peer, the RR does not modify the next-hop by default. However, if the **next-hop-self** command is applied to the IBGP peer receiving the route and the following command is configured, a **next-hop-self** command and label swap behavior can be provided:

- **MD-CLI**

```
configure router bgp rr-vpn-forwarding
```

- **classic CLI**

```
configure router bgp enable-rr-vpn-forwarding
```

Alternatively, the same behavior can be achieved by configuring the following command and setting the **next-hop-self** command toward the targeted IBGP peer:

- **MD-CLI**

```
configure router bgp subconfed-vpn-forwarding
```

- **classic CLI**

```
configure router bgp enable-subconfed-vpn-forwarding
```

In either case, the **next-hop-self** command results in one of the following outcomes:

- If the IBGP session receiving the reflected route uses IPv4 transport then the BGP next-hop is set to the IPv4 local-address used to set up the session but encoded as an IPv4-mapped IPv6 address (for example, with the IPv4 address in the least significant 32 bits of a ::FFFF/96 prefix).
- If the IBGP session receiving the reflected route uses IPv6 transport and it is associated with an **advertise-ipv6-next-hops vpn-ipv6** command, the BGP next-hop is set to the value of the IPv6 local address used for set up of the session. Otherwise, the BGP next-hop is set to the IPv4 address of the system interface encoded as an IPv4-mapped IPv6 address (for example, with the IPv4 address in the least significant 32 bits of a ::FFFF/96 prefix).

When a VPN-IPv6 BGP route is reflected from one IBGP peer to another IBGP peer, the following command is configured, and the VPN-IPv6 route is matched and accepted by an export policy entry with a **next-hop** IP address action, the BGP next-hop of the matched routes changes to the **next-hop** IP address, if the IP address is specified as a 128-bit IPv6 address. If the IP address is specified as a 32-bit IPv4 address, the BGP next-hop changes to an IPv4-mapped IPv6 address encoding IP address.

- **MD-CLI**

```
configure router bgp rr-vpn-forwarding
```

- **classic CLI**

```
configure router bgp enable-rr-vpn-forwarding
```

5.5.3.5 Label-IPv4 routes

SR OS routers can always send and receive label-IPv4 routes with IPv4 next-hops. They can also be configured (using the **extended-nh-encoding** command) to receive label-IPv4 routes with IPv6 next-hop addresses from selected BGP peers by signaling the corresponding Extended NH Encoding BGP capability to those peers during session setup. If the capability is not advertised to a peer then such routes are not accepted from that peer. Also, if the SR OS router does not receive an Extended NH Encoding capability advertisement for [NLRI AFI=1, NLRI SAFI=4, next-hop AFI=2] from a peer then it is not advertise label-IPv4 routes with IPv6 next-hops to that peer.

When a label-IPv4 BGP route is advertised to an EBGP peer, next-hop-self applies unless the EBGP session has **next-hop-unchanged** enabled for the label-ipv4 address family. Next-hop-self results in one of the following outcomes.

- If the EBGP session uses IPv4 transport, then the BGP next-hop is taken from the value of the local-address used to setup the session.
- If the EBGP peer opened an IPv6-transport session by advertising an extended NH encoding capability with (NLRI AFI=1, NLRI SAFI=4, next-hop AFI=2) AND, in the configuration of the local router, the session is associated with an **advertise-ipv6-next-hops label-ipv4** command, then the BGP next-hop

is set to the value of the IPv6 local-address used for setup of the session. Otherwise, the BGP next-hop is set to the IPv4 address of the system interface.

When a label-IPv4 BGP route is received from an EBGP peer and re-advertised to an IBGP or confederation-EBGP peer, next-hop-self applies unless the IBGP or confederation EBGP session has **next-hop-unchanged** enabled for the label-ipv4 address family. Next-hop-self results in one of the following outcomes.

- If the IBGP or confederation EBGP session uses IPv4 transport, then the BGP next-hop is taken from the value of the local-address used to setup the session.
- If the IBGP or confederation EBGP peer opened an IPv6-transport session by advertising an extended NH encoding capability with (NLRI AFI=1, NLRI SAFI=4, next-hop AFI=2) AND, in the configuration of the local router, the session is associated with an **advertise-ipv6-next-hops label-ipv4** command then the BGP next-hop is set to the value of the IPv6 local-address used for setup of the session. Otherwise, the BGP next-hop is set to the IPv4 address of the system interface

When a label-IPv4 BGP route is reflected from one IBGP peer to another IBGP peer, the RR does not modify the next-hop by default. However, if the **next-hop-self** command is applied to the IBGP peer receiving the route, then this results in one of the following outcomes.

- If the IBGP session receiving the reflected route uses IPv4 transport, then the BGP next-hop is taken from the value of the local-address used to setup the session.
- If the IBGP session receiving the reflected route opened an IPv6-transport session by advertising an extended NH encoding capability with (NLRI AFI=1, NLRI SAFI=4, next-hop AFI=2) AND, in the configuration of the local router, the session is associated with an **advertise-ipv6-next-hops label-ipv4** command then the BGP next-hop is set to the value of the IPv6 local-address used for setup of the session. Otherwise, the BGP next-hop is set to the IPv4 address of the system interface.

When a label-IPv4 BGP route is reflected from one IBGP peer to another IBGP peer and the label-IPv4 route is matched and accepted by an export policy entry with a **next-hop** IP-address action, this changes the BGP next-hop of the matched routes to the IP address, except if the IP address is an IPv6 address and the receiving IBGP peer did not advertise an extended NH encoding capability with (NLRI AFI=1, NLRI SAFI=128, next-hop AFI=2) or, in the configuration of the local router, the session is not associated with an **advertise-ipv6-next-hops label-ipv4** command. In this case, the route is treated as though it was rejected by the policy entry.

5.5.3.6 Label-IPv6 routes

SR OS routers never send or receive label-IPv6 routes with 32-bit IPv4 next-hop addresses.

When a label-IPv6 BGP route is advertised to an EBGP peer, the **next-hop-self** command applies unless the EBGP session has the **next-hop-unchanged** command enabled for the label-ipv6 address family. Using the **next-hop-self** command results in one of the following outcomes:

- If the EBGP session uses IPv4 transport, then the BGP next-hop is taken from the value of the local address used to set up the session and encoded as an IPv4-mapped IPv6 address.
- If the EBGP peer opened an IPv6 transport session and it is associated with an **advertise-ipv6-next-hops label-ipv6** command then the BGP next-hop is set to the value of the IPv6 local address used for set up of the session. Otherwise, the BGP next-hop is set to the IPv4 address of the system interface encoded as an IPv4-mapped IPv6 address.

When a label-IPv6 BGP route is received from an EBGP peer and re-advertised to an IBGP or confederation EBGP peer, the **next-hop-self** command applies unless the IBGP or confederation EBGP

session has the **next-hop-unchanged** command enabled for the label-IPv6 address family. Using the **next-hop-self** command results in one of the following:

- If the IBGP or confederation EBGP session uses IPv4 transport, the BGP next-hop is taken from the value of the local address used to setup the session and encoded as an IPv4-mapped IPv6 address.
- If the IBGP or confederation EBGP peer opened an IPv6 transport session and it is associated with an **advertise-ipv6-next-hops label-ipv6** command, the BGP next-hop is set to the value of the IPv6 local address used for set up of the session. Otherwise, the BGP next-hop is set to the IPv4 address of the system interface encoded as an IPv4-mapped IPv6 address.

5.5.3.7 Next-hop resolution

To use a BGP route for forwarding, a BGP router must know how to reach the BGP next-hop of the route. The process of determining the local interface or tunnel used to reach the BGP next-hop is called next-hop resolution. This process depends on the type of route (the AFI/SAFI) and various configuration settings.

The following process applies to the next-hop resolution of IPv4 and IPv6 (unlabeled) unicast routes:

- When an IPv4 or IPv6 route is received with an IPv4-mapped IPv6 address as a next hop, BGP next-hop resolution is based on the embedded IPv4 address, which is matched with either IPv4 routes or IPv4 tunnels, depending on the configuration. The IPv4-mapped IPv6 address is never interpreted as an IPv6 address and is not matched with IPv6 routes or IPv6 tunnels.
- By default, BGP routes are ineligible to resolve a BGP next hop. Use the following command to enable the use of BGP routes to resolve BGP next hops.

  ```
  configure router bgp next-hop-resolution use-bgp-routes
  ```

  A maximum of four levels of recursion are supported.
- The LPM route matching the BGP next-hop address is the only route considered for resolving the next hop. If the LPM route is ineligible to resolve the next hop, the BGP route is unresolved.
- A VPN-leak route leaked from another router instance is eligible to resolve a BGP next hop, provided that it corresponds to a static route with direct next hops.
- If the LPM route is rejected by the next-hop resolution policy configured by the user, the BGP next hop is unresolved, and all routes with that next hop are considered invalid and not advertised to peers.
- BGP shortcuts are described in Next-hop resolution of BGP unlabeled IPv4 unicast routes to tunnels.

The following process applies to the next-hop resolution of VPN-IPv4 and VPN-IPv6 routes:

- When an VPN-IPv4 or VPN-IPv6 route is received with an IPv4-mapped IPv6 address as next-hop, BGP next-hop resolution is based on the embedded IPv4 address and this IPv4 address is matched to IPv4 routes and/or IPv4 tunnels, depending on configuration. The IPv4-mapped IPv6 address is never interpreted as an IPv6 address to be compared to IPv6 routes or IPv6 tunnels.
- If the VPN-IP route is imported into a VPRN, the next-hop is resolved to a tunnel based on the auto-bind-tunnel configuration of the importing VPRN. For more information, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 3 Services Guide: IES and VPRN*.
- If the next-hop entry in the tunnel-table that resolves the VPN-IP route is rejected by the user-configured **next-hop-resolution vpn-family-policy**, the BGP next hop is unresolved and all the VPN-IP routes with that next hop are considered invalid.
- If the VPN-IP route is received from an IBGP or EBGP peer, and the router is a next-hop-self RR or a model-B ASBR, the order of resolution is as follows:

  - local (interface) route
  - longest prefix-match static route, excluding default static routes, but only if allow-static is configured
  - tunnel, based on the transport-tunnel resolution-filter options for family VPN; for more information, see Next-hop resolution of BGP labeled routes to tunnels

The following process applies to the next-hop resolution of EVPN routes:

- If an EVPN ES route is imported, the next hop is resolved to any tunnel in the tunnel-table. If there is no entry matching the next hop in the tunnel-table, a route matching the next hop in the route-table also resolves the next hop.
  - If the resolving route in the tunnel-table or the route-table is rejected by a user-configured **next-hop-resolution vpn-family-policy**, the ES route's next hop is unresolved.
  - If the ES route next hop is unresolved, the PE that advertised the route is not considered as candidate for Designated Forwarder (DF) election.
- The imported AD per-ES and AD per-EVI routes are always shown as resolved and valid, irrespective of the next-hop resolution or the configuration of a **next-hop-resolution vpn-family-policy**.
  - With DF election, the router always considers the advertising PE a valid candidate, even if the AD routes next hops are unresolved.
  - However, if the AD per-EVI next hop is unresolved, EVPN traffic is not sent to the advertising PE. This is true for EVPN VPWS or multihoming aliasing or backup procedures.
  - A matching tunnel-table entry can resolve the next-hop of an AD per-EVI route in an EVPN-MPLS service. A route-table entry (other than a shortcut) can resolve the AD per-EVI route next-hop in an EVPN-VXLAN service.
- For any other imported EVPN service route, including IP prefix, IPv6 prefix, MAC/IP, Inclusive Multicast (IMET) and Selective Multicast (SMET) routes, the next hop is resolved as follows.
  - In EVPN-MPLS services, the next hop is resolved to a tunnel based on the **auto-bind-tunnel** configuration of the importing service.
  - In EVPN-VXLAN services, the next hop is resolved to a route in the route-table. That route cannot be a shortcut route.
  - If the next-hop entry in the tunnel-table or route-table that resolves the EVPN service route is rejected by the user-configured **next-hop-resolution vpn-family-policy**, the BGP next hop is unresolved and all the EVPN routes with that next hop are considered invalid.

The following process applies to the next-hop resolution of label-unicast IPv4 and label-unicast IPv6 routes:

- When a BGP-LU route is received with an IPv4-mapped IPv6 address as the next-hop, BGP next-hop resolution is based on the embedded IPv4 address and this IPv4 address is matched to IPv4 routes or IPv4 tunnels, depending on the configuration. The IPv4-mapped IPv6 address is never interpreted as an IPv6 address to be compared to IPv6 routes or IPv6 tunnels.
- If the BGP-LU route is received by a control-plane-only RR and the following commands are configured:
  - **MD-CLI**

```
configure router bgp route-table-install false
configure router bgp next-hop-resolution labeled-routes rr-use-route-table
```

– **classic CLI**

```
configure router bgp disable-route-table-install
configure router bgp next-hop-resolution labeled-routes rr-use-route-table
```

In this case the order of the resolution is as follows:

- the local route
- the longest prefix-match static route, excluding default static routes

  Use this route to resolve the BGP next-hop if it is a static route with blackhole next-hop. For other types of static routes, use this route to resolve the next-hop only if **allow-static** is configured. If it is another type of static route, use it to resolve the next-hop only if allow-static is configured.
- the longest prefix-match IGP route

- If a label-unicast IPv4 route or label-unicast IPv6 route with a label (other than explicit-null) is received from an IBGP or EBGP peer and the router is not an RR with the **rr-use-route-table** command configured, the order of resolution is as follows:
  – the local route
  – the longest prefix-match static route, excluding default static routes

    Use this route to resolve the BGP next-hop if it is a static route with a blackhole next-hop. If it is another type of static route, use it to resolve the next-hop only if **allow-static** is configured.
  – a tunnel, based on the **transport-tunnel resolution-filter** options for family label-ipv4 or label-ipv6 (depending on the situation); for more information, see Next-hop resolution of BGP labeled routes to tunnels
- If a label-unicast IPv6 route with an IPv6 explicit-null label is received from an IBGP or EBGP peer and the router is not an RR with the **rr-use-route-table** command configured, the order of resolution is as follows:
  – the local route
  – the longest prefix-match static route, excluding default static routes

    Use this route to resolve the BGP next-hop if it is a static route with a blackhole next-hop. If it is another type of static route, use it to resolve the next-hop only if **allow-static** is configured.
  – a tunnel, based on the **transport-tunnel resolution-filter** options for family label-ipv4; for more information, see Next-hop resolution of BGP labeled routes to tunnels
  – if the following command is enabled and the longest prefix length route that matches the next-hop is a BGP IPv4 unlabeled, BGP IPv6 unlabeled, or other 6PE route with an explicit-null label.

    ```
    configure router bgp next-hop-resolution labeled-routes use-bgp-routes label-ipv6-
    explicit-null
    ```

    Use that route, subject to the following conditions:
    - the resolving route cannot be a leaked route
    - an unlabeled IPv4 route or IPv6 route is ineligible to resolve the next-hop of a label-unicast IPv6 route if the unlabeled route has any of its own BGP next-hops resolved by an IGP route or a 6over4 route
    - the label-unicast IPv6 route can be recursively resolved by other label-unicast IPv6 routes with **explicit-null** so that the final route has up to four levels of recursion

5.5.3.7.1 Next-hop resolution of BGP unlabeled IPv4 unicast routes to tunnels

To enable the next-hop resolution of unlabeled IPv4 routes using tunnels in the tunnel table of the router, use the commands in the following context to configure the IPv4 family.

```
configure router bgp next-hop-resolution shortcut-tunnel family
```

The family context provides commands to specify the resolution mode (**any**, **disabled** or **filter**) and set tunnel types that are eligible for use when filter mode is selected.

If the resolution mode is **disabled**, the next hops of unlabeled IPv4 routes can only be resolved by route table lookup.

If there are multiple tunnels in the tunnel table that are allowed by the **any** or **filter** resolution modes and that can resolve the BGP next-hop, the selection of the resolving tunnel is determined by factors such as route color, admin-tag-policy, tunnel-table preference, and LDP FEC prefix length.

If the **disallow-igp** command is enabled and no resolving tunnel is found in the tunnel table, no attempt is made to resolve the IPv4 BGP route using route table lookup.

The available tunneling options for IPv4 shortcuts are as follows:

- **BGP**

  This refers to IPv4 and IPv6 tunnels created by receiving BGP label-unicast IPv4 routes for /32 IPv4 prefixes and BGP label-unicast IPv6 routes for /128 IPv6 prefixes. The installation of BGP-LU IPv6 tunnels in TTM requires configuration of the following command:

  – **MD-CLI**

  ```
  configure router bgp label-allocation label-ipv6 explicit-null false
  ```

  – **classic CLI**

  ```
  configure router bgp label-allocation label-ipv6 disable-explicit-null
  ```

- **LDP**

  This refers to LDP FEC prefixes imported into the tunnel table. For resolution purposes, BGP selects the LDP FEC that is the longest-prefix-match (LPM) of the BGP next-hop address.

- **RSVP**

  This refers to RSVP tunnels in tunnel-table. This option allows BGP to use the best metric RSVP LSP to the address of the BGP next-hop. This address can correspond to the system interface or to another loopback interface of the remote BGP router. In the case of multiple RSVP LSPs with the same lowest metric, BGP selects the LSP with the lowest tunnel ID.

- **SR ISIS**

  This refers to segment routing tunnels (shortest path) to destinations reachable by the IS-IS protocol. This option allows BGP to use the segment routing tunnel in tunnel-table submitted by the lowest preference IS-IS instance or, in case of a tie, the lowest numbered IS-IS instance.

- **SR OSPF**

  This refers to segment routing tunnels (shortest path) to destinations reachable by the OSPF protocol. This option allows BGP to use the segment routing tunnel in the tunnel table submitted by the lowest preference OSPF instance or (in case of a tie) the lowest numbered OSPF instance.

- **SR policy**

  This refers to segment routing policies that are statically configured in the local router or learned via BGP routes (AFI 1/SAFI 73). For BGP to resolve the next-hop of an IPv4 route using an SR policy, the highest numbered color-extended community attached to the IPv4 route must match the color of the SR policy. Also if the CO bits of this color-extended community have the value 00 the BGP next-hop of the route must exactly match the endpoint of the SR policy.

- **SR TE**

  This refers to traffic engineered (TE) segment routing tunnels. This option allows BGP to use the best metric SR-TE tunnel to the address of the BGP next-hop. In the case of multiple SR-TE tunnels with the same lowest metric, BGP selects the tunnel with the lowest tunnel ID.

5.5.3.7.2 Next-hop resolution of BGP unlabeled IPv6 unicast routes to tunnels

To enable the next-hop resolution of unlabeled IPv6 routes using tunnels in the tunnel-table of the router, use the commands in the following context to configure the IPv6 family.

```
configure router bgp next-hop-resolution shortcut-tunnel
```

If the next-hop of the IPv6 BGP route contains an IPv4-mapped IPv6 address, the shortcut-tunnel configuration applies to the use of IPv4 tunnels and IPv4 routes that match the embedded IPv4 address in the BGP next-hop. If the BGP next-hop is any other IPv6 address the shortcut-tunnel configuration applies to the use of IPv6 tunnels and IPv6 routes that match the full address of the BGP next-hop.

The **family ipv6** context provides commands to specify the resolution mode (**any**, **disabled** or **filter**) and the set of tunnel types that are eligible for use if the **filter** mode is selected.

If there are multiple tunnels in tunnel-table that are allowed by the **any** or **filter** resolution modes and that can resolve the BGP next-hop, the selection of the resolving tunnel is determined by factors such as route color, admin-tag-policy, tunnel-table preference, and LDP FEC prefix length.

If the resolution mode is set to disabled, the next-hops of unlabeled IPv6 routes can only be resolved by route table lookup.

If **disallow-igp** is enabled and no resolving tunnel is found in the tunnel table, no attempt is made to resolve the IPv6 BGP route using route table lookup.

The available tunneling options for IPv6 BGP routes with IPv4-mapped IPv6 next-hops are as follows:

- **BGP**

  This refers to IPv4 tunnels created by receiving BGP label-unicast IPv4 routes for /32 IPv4 prefixes.

- **LDP**

  This refers to /32 and shorter length LDP FEC prefixes imported into the tunnel table. For resolution purposes, BGP selects the LDP FEC that is the longest-prefix-match (LPM) of the BGP next-hop address.

- **RSVP**

  This refers to RSVP tunnels in tunnel-table. This option allows BGP to use the best metric RSVP LSP to the address of the BGP next-hop. This address can correspond to the system interface or to another loopback interface of the remote BGP router. In the case of multiple RSVP LSPs with the same lowest metric, BGP selects the LSP with the lowest tunnel ID.

- **SR ISIS**

This refers to segment routing tunnels (shortest path) to destinations reachable by the IS-IS protocol. This option allows BGP to use the segment routing tunnel in tunnel-table submitted by the lowest preference IS-IS instance or (in case of a tie) the lowest numbered IS-IS instance.

- **SR OSPF**

  This refers to segment routing tunnels (shortest path) to destinations reachable by the OSPF protocol. This option allows BGP to use the segment routing tunnel in tunnel-table submitted by the lowest preference OSPF instance or (in case of a tie) the lowest numbered OSPF instance.

- **SR policy**

  This refers to segment routing policies that are statically configured in the local router or learned via BGP routes (AFI 1/SAFI 73). For BGP to resolve the next-hop of an IPv4 route using an SR policy the highest numbered color-extended community attached to the IPv4 route must match the color of the SR policy and if the CO bits of this color -extended community have the value 00 the BGP next-hop of the route must exactly match the endpoint of the SR policy.

- **SR TE**

  This refers to TE segment routing tunnels. This option allows BGP to use the best metric SR-TE tunnel to the address of the BGP next-hop. In the case of multiple SR-TE tunnels with the same lowest metric, BGP selects the tunnel with the lowest tunnel ID.

5.5.3.7.3 Next-hop resolution of BGP labeled routes to tunnels

Use the commands in the following context to configure next-hop resolution of BGP labeled routes.

```
configure router bgp next-hop-res labeled-routes transport-tunnel
```

The **transport-tunnel** context provides separate control for the different types of BGP labeled routes: label-IPv4, label-IPv6, and VPN routes (which includes both VPN-IPv4 and VPN-IPv6 routes). By default, all labeled routes resolve to LDP (even if the preceding CLI commands are not configured in the system).

If the **resolution** command option is set to **disabled**, the default binding to LDP tunnels resumes. If the **resolution** command option is set to **any**, the supported tunnel type selection is based on TTM preference. The order of preference of TTM tunnels is: RSVP, SR-TE, LDP, segment routing OSPF, segment routing IS-IS, and UDP.

The **rsvp** command instructs BGP to search for the best metric RSVP LSP to the address of the BGP next-hop. The address can correspond to the system interface or to another loopback used by the BGP instance on the remote node. The LSP metric is provided by MPLS in the tunnel table. In the case of multiple RSVP LSPs with the same lowest metric, BGP selects the LSP with the lowest tunnel ID.

The **ldp** command instructs BGP to search for an LDP LSP with a FEC prefix corresponding to the address of the BGP next-hop.

The **bgp** command instructs BGP to search for a BGP tunnel in TTM with a prefix matching the address of the BGP next-hop. A label-unicast IPv4 route cannot be resolved by another label-unicast IPv4 or IPv6 route. A label-unicast IPv6 route cannot be resolved by another label-unicast IPv6 route, but it can be resolved by a label-unicast IPv4 route.

When the **sr-isis** or **sr-ospf** option is enabled, an SR tunnel to the BGP next-hop is selected in the TTM from the lowest preference IS-IS or OSPF instance. If many instances have the same lowest preference, the lowest numbered IS-IS or OSPF instance is chosen.

The **sr-te** launches a search for the best metric SR-TE LSP to the address of the BGP next-hop. The LSP metric is provided by MPLS in the tunnel table. In the case of multiple SR-TE LSPs with the same lowest metric, BGP selects the LSP with the lowest tunnel ID.

The **udp** command instructs BGP to look for an MPLSoUDP tunnel to the address of the BGP next-hop.

If one or more explicit tunnel types are specified using the **resolution-filter** command, only these tunnel types are selected again following the TTM preference. The **resolution** command must be set to **filter** to activate the list of tunnel types configured under the **resolution-filter** command context.

5.5.3.8 Next-hop tracking

In SR OS next-hop resolution is not a one-time event. If the IP route or tunnel that was used to resolve a BGP next-hop is withdrawn because of a failure or configuration change an attempt is made to re-resolve the BGP next-hop using the next-best route or tunnel. If there are no more eligible routes or tunnels to resolve the BGP next-hop then the BGP next-hop becomes unresolved. The continual process of monitoring and reacting to resolving route/tunnel changes is called next-hop tracking. In SR OS next-hop tracking is completely event driven as opposed to timer driven; this provides the best possible convergence performance.

5.5.3.9 Next-hop indirection

SR OS supports next-hop indirection for most types of BGP routes. Next-hop indirection means BGP next-hops are logically separated from resolved next-hops in the forwarding plane (IOMs). This separation allows routes that share the same BGP next-hops to be grouped so that when there is a change to the way a BGP next-hop is resolved only one forwarding plane update is needed, as opposed to one update for every route in the group. The convergence time after the next-hop resolution change is uniform and not linear with the number of prefixes; in other words, the next-hop indirection is a technology that supports Prefix Independent Convergence (PIC). SR OS uses next-hop indirection whenever possible; there is no option to disable the functionality.

5.5.3.10 Entropy label for RFC 8277 BGP labeled routes

The router supports the MPLS entropy label, as specified in RFC 6790, on RFC 8277 BGP labeled routes. LSR nodes in a network can load-balance labeled packets in a more granular way than by hashing on the standard label stack. For more information, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide*.

Entropy Label Capability (ELC) signaling is not supported for labeled routes representing BGP tunnels. Instead, ELC is configured at the head end LER using the following command.

```
configure router bgp override-tunnel-elc
```

This command causes the router to ignore any advertisements for ELC that may or may not be received from the network, and instead to assume that the whole domain supports entropy labels.

5.5.4 MED

The Multi-Exit Discriminator (MED) attribute is an optional attribute that can be added to routes advertised to an EBGP peer to influence the flow of inbound traffic to the AS. The MED attribute carries a 32-bit metric value. A lower metric is better than a higher metric when MED is compared by the BGP decision process. Unless the **always-compare-med** command is configured MED is compared only if the routes come from the same neighbor AS. By default, if a route is received without a MED attribute it is evaluated by the BGP decision process as though it had a MED containing the value 0, but this can be changed so that a missing MED attribute is handled the same as a MED with the maximum value. SR OS always removes the received MED attribute when advertising the route to an EBGP peer.

5.5.4.1 Deterministic MED

Deterministic MED is an optional enhancement to the BGP decision process that causes BGP to group paths that are equal up to the MED comparison step based on the neighbor AS. BGP compares the best path from each group to arrive at the overall best path. This change to the BGP decision process makes best path selection completely deterministic in all cases. Without **deterministic-med**, the overall best path selection is sometimes dependent on the order of route arrival because of the rule that MED cannot be compared in routes from different neighbor AS.

Note: When BGP routes are leaked into a target BGP RIB, they are not grouped (in a deterministic MED context) with routes learned by that target RIB, even if the neighbor AS happens to be the same.

5.5.5 Local preference

The LOCAL_PREF attribute is a well-known attribute that should be included in every route advertised to an IBGP or confederation-EBGP peer. It is used to influence the flow of outbound traffic from the AS. The local preference is a 32-bit value and higher values are more preferred by the BGP decision process. The LOCAL_PREF attribute is not included in routes advertised to EBGP peers (if the attribute is received from an EBGP peer it is ignored).

In SR OS the default local preference is 100 but this can be changed with the **local-preference** command or using route policies. When a LOCAL_PREF attribute needs to be added to a route because it does not have one (for example, because it was received from an EBGP peer) the value is the configured or default **local-preference** unless overridden by policy.

5.5.6 Route aggregation path attributes

An aggregate route is a configured IP route that is activated and installed in the routing table when it has at least one contributing route. A route (R) contributes to an aggregate route (S1) if all of the following conditions are true:

- the prefix length of (R) is greater than the prefix length of (S1)
- the prefix bits of (R) match the prefix bits of (S1) up to the prefix length of (S1)
- there is no other active aggregate route (S2) with a longer prefix length than (S1) that meets the previous two conditions
- (R) is actively used for forwarding and is not an aggregate route

- (R) is accepted by the route policy that is associated with (S1); if there is no configured route policy then (R1) is by default considered accepted

When an aggregate route is activated by a router, it is not installed in the forwarding table by default. In general though, it is advisable to specify a black-hole next-hop option for an aggregate route, so that when it is activated it is installed in the forwarding table with a black-hole next-hop; this avoids the possibility of creating a routing loop. SR OS also supports the option to program an aggregate route into the forwarding table with an indirect next-hop; in this case, packets matching the aggregate route but not a more-specific contributing route are forwarded toward the indirect next-hop instead of being discarded.

An active aggregate route can be advertised to a BGP peer (by exporting it into BGP) and this can avoid the need to advertise the more-specific contributing routes to the peer, reducing the number of routes in the peer AS and improving overall scalability. When a router advertises an aggregate route to a BGP peer the attributes in the route are set as follows.

- The ATOMIC_AGGREGATE attribute is included in the route if at least one contributing route has the ATOMIC_AGGREGATE attribute or the aggregate route was formed without the **as-set** command option and at least one contributing route has a non-empty AS_PATH. The ATOMIC_AGGREGATE attribute indicates that some of the AS numbers present in the AS paths of the contributing routes are missing from the advertised AS_PATH.
- The AGGREGATOR attribute is added to the route. This attribute encodes, by default, the global AS number (or confederation ID) and router ID (BGP identifier) of the router that formed the aggregate, but these values can be changed on a per aggregate route basis using the **aggregator** command option. The AS number in the AGGREGATOR attribute is either 2 bytes or 4 bytes (if the 4-octet ASN capability was announced by both peers). The router ID in the aggregate routes advertised to a particular set of peers can be set to 0.0.0.0 using the **aggregator-id-zero** command.
- The BGP next-hop is set to the local-address used with the peer receiving the route regardless of the BGP next-hops of the contributing routes.
- The ORIGIN attribute is based on the ORIGIN attributes of the contributing routes as described in RFC 4271.
- The information in the AS_PATH attribute depends on the **as-set** option of the aggregate route.
  - If the **as-set** option is not specified the AS_PATH of the aggregate route starts as an empty AS path and has elements added per the description in AS path.
  - If the **as-set** option is specified and all the contributing routes have the same AS_PATH then the AS_PATH of the aggregate route starts with that common AS_PATH and has elements added per the description in AS path.
  - If the **as-set** option is specified and some of the contributing routes have different AS paths the AS_PATH of the aggregate route starts with an AS_SET and/or an AS_CONFED_SET and then adds elements per the description in AS path.
- The COMMUNITY attribute contains all of the communities from all of the contributing routes unless the **discard-component-communities** command option is configured for the aggregate route. It also contains the communities associated directly with the aggregate route itself (up to 12 per aggregate route).
- No MED attribute is included by default.

Note: SR OS does not require all the contributing routes to have the same MED value.

5.5.7 Community attributes

A BGP route can be associated with one or more communities. There are three kinds of BGP communities:

- standard communities (each 4 bytes in length, all packed into a path attribute with type code 8)
- extended communities (each 8 bytes in length, potentially many possible subtypes, all packed into a path attribute with type code 16)
- large communities (each 12 bytes in length, all packet into a path attribute with type code 32)

5.5.7.1 Standard communities

In a standard community, the first two bytes usually encode the AS number of the administrative entity that assigned the value in the last two bytes. In SR OS, a standard community value is configured using the following format:

<asnum:comm-value>

The colon is a required separator character. In route policy applications, multiple standard community values can be matched with a regular expression in the following format:

<regex1>:<regex2>

where *regex1* and *regex2* are two regular expressions that are evaluated one numerical digit at a time.

The following well-known standard communities are understood and acted upon accordingly by SR OS routers.

- **NO_EXPORT**

  When a route carries this community, it must not be advertised outside a confederation boundary (for example, to EBGP peers).

- **NO_ADVERTISE**

  When a route carries this community, it must not be advertised to any other BGP peer.

- **NO_EXPORT_SUBCONFED**

  When a route carries this community, it must not be advertised outside a member AS boundary (for example, to confed-EBGP peers or EBGP peers.

- **LLGR_STALE**

  When a route carries this community, it indicates that the route was propagated by a router that is a long-lived graceful restart helper and normally (in the absence of LLGR) the route would have been withdrawn.

- **NO_LLGR**

  When a route carries this community, it indicates that the route should not be retained and used past the normal graceful restart window of time.

- **BLACKHOLE**

  When a route carries this community, it indicates that the route should be installed into the FIB with a blackhole next-hop.

Standard communities can be added to or removed from BGP routes using BGP import and export policies. When a BGP route is locally originated by exporting a static or aggregate route into BGP, and the static or aggregate route has one or more standard communities, these community values are automatically added to the BGP route. This may affect the advertisement of the locally originated route if one of the well-known communities is associated with the static or aggregate route.

To remove all the standard communities from all routes advertised to a BGP peer, use the following command:

- **MD-CLI**

```
configure router bgp send-communities standard false
```

- **classic CLI**

```
configure router bgp disable-communities standard
```

5.5.7.2 Extended communities

Extended communities serve specialized roles. Each extended community is eight bytes. The first one or two bytes identifies the type or sub-type and the remaining six or seven bytes identify a value. Some of the more common extended communities supported by SR OS include:

- Transitive 2-octet AS-specific
  - Route target (type 0x0002)
  - Route origin (type 0x0003)
  - OSPF domain ID (type 0x0005)
  - Source AS (type 0x0009)
  - L2VPN identifier (type 0x000A)
- Non-transitive 2-octet AS-specific

  Link bandwidth (0x4004)
- Transitive 4-octet AS-specific
  - Route target (type 0x0202)
  - Route origin (type 0x0203)
  - OSPF domain ID (type 0x0205)
  - Source AS (type 0x0209)
- Transitive IPv4-address-specific
  - Route target (type 0x0102)
  - Route origin (type 0x0103)
  - OSPF domain ID (type 0x0105)
  - L2VPN identifier (type 0x010A)
  - VRF route import (type 0x010B)
- Transitive opaque
  - OSPF route type (type 0x0306)

  - Color extended community (type 0x030B)
- Non-transitive opaque

  BGP origin validation state (type 0x4300)
- Transitive experimental
  - FlowSpec traffic rate (type 0x8006)
  - FlowSpec traffic action (type 0x8007)
  - FlowSpec redirect (type 0x8008)
  - FlowSpec traffic-remarking (0x8009)
  - Layer 2 info (type 0x800A)
- Transitive FlowSpec

  FlowSpec interface-set (type 0x0702)
- Non-transitive FlowSpec

  FlowSpec interface-set (type 0x4702)

Extended communities can be added to or removed from BGP routes using BGP import and export policies. When a BGP route is locally originated by exporting a static or aggregate route into BGP, and the static or aggregate route has one or more extended communities, these community values are automatically added to the BGP route.

Note: While it may not make sense to add specific types of extended communities to routes of certain address families, SR OS allows such actions.

To remove all the extended communities from all routes advertised to a BGP peer, use the following command:

- **MD-CLI**

```
configure router bgp send-communities extended false
```

- **classic CLI**

```
configure router bgp disable-communities extended
```

5.5.7.3 Large communities

Each large community is a 12-byte value, formed from the logical concatenation of three 4-octet values: a Global Administrator part, a Local Data part 1, and Local Data part 2. The Global Administrator is a four-octet namespace identifier, which should be an Autonomous System Number assigned by IANA. The Global Administrator field is intended to allow different Autonomous Systems to define large communities without collision. Local Data Part 1 is a four-octet operator-defined value and Local Data Part 2 is another four-octet operator-defined value.

In SR OS, a large community value is configured using the format <*ext-asnum*>:<*ext-comm-val*>:<*ext-comm-val*>; the colon is a required separator character between each of the 4-byte values. In route policy applications, it is possible to match multiple large community values with a regular expression in the format <*regex1*>&<*regex2*>&<*regex3*>, where *regex1*, *regex2* and *regex3* are three regular expressions, each evaluated one numerical (decimal) digit at a time.

Large communities can be added to or removed from BGP routes using BGP import and export policies. When a BGP route is locally originated by exporting a static or aggregate route into BGP, and the static or aggregate route has one or more large communities, these community values are automatically added to the BGP route.

To remove all the large communities from all routes advertised to a BGP peer, use the following command:

- **MD-CLI**

```
configure router bgp send-communities large false
```

- **classic CLI**

```
configure router bgp disable-communities large
```

5.5.8 Route reflection attributes

The ORIGINATOR_ID and CLUSTER_LIST are optional non-transitive attributes that play a role in route reflection, as described in the section titled Route reflection.

5.5.9 Multiprotocol BGP attributes

As discussed in the BGP chapter overview the uses of BGP have increased well beyond Internet IPv4 routing because of its support for multiprotocol extensions, or more simply MP-BGP. MP-BGP allows BGP peers to exchange routes for NLRI other than IPv4 prefixes - for example IPv6 prefixes, Layer 3 VPN routes, Layer 2 VPN routes, FlowSpec rules, and so on. A BGP router that supports MP-BGP indicates the types of routes it wants to exchange with a peer by including the corresponding AFI (Address Family Identifier) and SAFI (Subsequent Address Family Identifier) values in the MP-BGP capability of its OPEN message. The two peers forming a session do not need to indicate support for the same address families. As long as there is one AFI/SAFI in common the session establishes and routes associated with all the common AFI/SAFI can be exchanged between the peers.

The list of AFI/SAFI advertised in the MP-BGP capability is controlled entirely by the **family** commands. The AFI/SAFI supported by the SR OS and the method of configuring the AFI/SAFI support is summarized in Table 4: Multiprotocol BGP support in SR OS.

Table 4: Multiprotocol BGP support in SR OS

Name	AFI	SAFI	Configuration commands
IPv4 unicast	1	1	**family ipv4**
IPv4 multicast	1	2	**family mcast-ipv4**
IPv4 labeled unicast	1	4	**family label-ipv4**
NG-MVPN IPv4	1	5	**family mvpn-ipv4**
MDT-SAFI	1	66	**family mdt-safi**
VPN-IPv4	1	128	**family vpn-ipv4**

Name	AFI	SAFI	Configuration commands
VPN-IPv4 multicast	1	129	**family mcast-vpn-ipv4**
RT constrain	1	132	**family route-target**
IPv4 flow-spec	1	133	**family flow-ipv4**
IPv6 unicast	2	1	**family ipv6**
IPv6 multicast	2	2	**family mcast-ipv6**
IPv6 labeled unicast	2	4	**family label-ipv6**
NG-MVPN IPv6	2	5	**family mvpn-ipv6**
VPN-IPv6	2	128	**family vpn-ipv6**
IPv6 flow-spec	2	133	**family flow-ipv6**
Multisegment PW	25	6	**family ms-pw**
L2 VPN	25	65	**family l2-vpn**
EVPN	25	70	**family evpn**

To advertise reachable routes of a particular AFI/SAFI a BGP router includes a single MP_REACH_NLRI attribute in the UPDATE message. The MP_REACH_NLRI attribute encodes the AFI, the SAFI, the BGP next-hop and all the reachable NLRI. To withdraw routes of a particular AFI/SAFI a BGP router includes a single MP_UNREACH_NLRI attribute in the UPDATE message. The MP_UNREACH_NLRI attribute encodes the AFI, the SAFI and all the withdrawn NLRI. While it is valid to advertise and withdraw IPv4 unicast routes using the MP_REACH_NLRI and MP_UNREACH_NLRI attributes, SR OS always uses the IPv4 fields of the UPDATE message to convey reachable and unreachable IPv4 unicast routes.

5.5.10 4-octet AS attributes

The AS4_PATH and AS4_AGGREGATOR path attributes are optional transitive attributes that support the gradual migration of routers that can understand and parse 4-octet ASN numbers. The use of these attributes is discussed in the section titled 4-octet autonomous system numbers.

5.5.11 AIGP metric

The accumulated IGP (AIGP) metric is an optional non-transitive attribute that can be attached to selected routes (using route policies) to influence the BGP decision process to prefer BGP paths with a lower end-to-end IGP cost, even when the compared paths span more than one AS or IGP instance. AIGP is different from MED in several important ways:

- AIGP is not intended to be transitive between completely distinct autonomous systems (only across internal AS boundaries)
- AIGP is always compared in paths that have the attribute, regardless of whether they come from different neighbor AS or not

- AIGP is more important than MED in the BGP decision process (see the section titled BGP decision process)
- AIGP is automatically incremented every time there is a BGP next-hop change so that it can track the end-to-end IGP cost. All arithmetic operations on MED attributes must be done manually (for example, using route policies)

In the SR OS implementation, AIGP is supported only in the base router BGP instance and only for the following types of routes: IPv4, label-IPv4, IPv6 and label-IPv6. The AIGP attribute is only sent to peers configured with the **aigp** command. If the attribute is received from a peer that is not configured for **aigp** or if the attribute is received in a non-supported route type the attribute is discarded and not propagated to other peers (but it is still displayed in BGP **show** commands).

When a 7450, 7750, or 7950 router receives a route with an AIGP attribute and it re-advertises the route to an AIGP-enabled peer without any change to the BGP next-hop the AIGP metric value is unchanged by the advertisement (RIB-OUT) process. But if the route is re-advertised with a new BGP next-hop the AIGP metric value is automatically incremented by the route table (or tunnel table) cost to reach the received BGP next-hop and/or by a statically configured value (using route policies).

5.6 BGP routing information base

The entire set of BGP routes learned and advertised by a BGP router make up its BGP Routing Information Base (RIB). Conceptually the BGP RIB can be divided into 3 parts:

- RIB-IN
- LOC-RIB
- RIB-OUT

The RIB-IN (or Adj-RIBs-In as defined in RFC 4271) holds the BGP routes that were received from peers and that the router decided to keep (store in its memory).

The LOC-RIB contains modified versions of the BGP routes in the RIB-IN. The path attributes of a RIB-IN route can be modified using BGP import policies. All of the LOC-RIB routes for the same NLRI are compared in a procedure called the BGP decision process that results in the selection of the best path for each NLRI. The best paths in the LOC-RIB are the ones that are actually 'usable' by the local router for forwarding, filtering, auto-discovery, and so on.

The RIB-OUT (or Adj-RIBs-Out as defined in RFC 4271) holds the BGP routes that were advertised to peers. Normally a BGP route is not advertised to a peer (in the RIB-OUT) unless it is 'used' locally but there are exceptions. BGP export policies modify the path attributes of a LOC-RIB route to create the path attributes of the RIB-OUT route. A particular LOC-RIB route can be advertised with different path attribute values to different peers so there can exist a 1:N relationship between LOC-RIB and RIB-OUT routes.

The following sections describe many important BGP features in the context of the RIB architecture described above.

5.6.1 RIB-IN features

SR OS implements the following features related to RIB-IN processing:

- UPDATE message fault tolerance; this is described in the section titled Update message error handling
- BGP import policies

5.6.1.1 BGP import policies

The **import** command is used to apply one or more policies (up to 15) to a neighbor, group or to the entire BGP context. The **import** command that is most-specific to a peer is the one that is applied. An **import** policy command applied at the **neighbor** level takes precedence over the same command applied at the **group** or global level. An **import** policy command applied at the **group** level takes precedence over the same command specified on the global level. The **import** policies applied at different levels are not cumulative. The policies listed in an **import** command are evaluated in the order in which they are specified.

Note: The **import** command can reference a policy before it has been created (as a **policy-statement**).

When an IP route is rejected by an import policy it is still maintained in the RIB-IN so that a policy change can be made later on without requiring the peer to re-send all its RIB-OUT routes. This is sometimes called soft reconfiguration inbound and requires no special configuration in SR OS.

When a VPN route is rejected by an import policy or not imported by any services it is deleted from the RIB-IN. For VPN-IPv4 and VPN-IPv6 routes this behavior can be changed by configuring the **mp-bgp-keep** command; this option maintains rejected VPN-IP routes in the RIB-IN so that a Route Refresh message does not have to be issued when there is an import policy change.

5.6.2 LOC-RIB features

SR OS implements the following features related to LOC-RIB processing:

- BGP decision process
- BGP route installation in the route table
- BGP route installation in the tunnel table
- BGP fast reroute
- QoS Policy Propagation via BGP (QPPB)
- policy accounting
- route flap damping (RFD)

These features are discussed in the following sections.

5.6.2.1 BGP decision process

When a BGP router has multiple paths in its RIB for the same NLRI, its BGP decision process is responsible for deciding which path is the best. The best path can be used by the local router and advertised to other BGP peers.

On 7450 ESS, 7750 SR, and 7950 XRS routers, the BGP decision process orders received paths based on the following sequence of comparisons. If there is a tie between paths at any step, BGP proceeds to the next step.

1. Select a valid route over an invalid route. If a BGP route is invalid because its next-hop it is not resolved then it may still be advertisable if there are no valid routes. For example, an unresolved route can be reflected by a route reflector if it is not trying to set **next-hop-self**.

2. Prefer a route for which **disable-route-table-install** does not apply over a route for which **disable-route-table-install** has been specified.
3. Prefer a non-stale route over a stale route (in the context of long-lived graceful restart).
4. A default route generated by the **send-default** command is less preferred than a default route programmed by other means.
5. Select the route with the lowest origin validation state, where Valid<Not-Found<Invalid.
6. Select the route with the numerically lowest route-table preference. For VPN-IP routes this also consider the number of VPRNs that imported the route.
7. Select the route with the highest local preference.
8. Select the route with an AIGP metric. If they both have an AIGP metric, select the route with the lowest sum of:
   a. AIGP metric value stored with the LOC-RIB copy of the route
   b. route-table or tunnel-table cost between the calculating router and the BGP next-hop in the received route
9. Select the route with the shortest AS path. AS numbers in AS_CONFED_SEQ and AS_CONFED_SET elements do not count toward the AS path length. This step is skipped if **as-path-ignore** is configured for the address family.
10. Select the route with the lowest Origin (IGP<EGP<Incomplete).
11. Select the route with the lowest MED. The following considerations apply:
    - By default, MED is only compared between routes from the same neighbor Autonomous System (AS) (ignoring confederation sub ASNs). This default can be changed by configuring the **strict-as** option of the relevant **always-compare-med** command to **false**.
    - By default, a missing MED is considered equivalent to a MED of 0. This default can be changed by configuring the **missing-med-infinity** option of the relevant **always-compare-med** command.
    - The relevant **always-compare-med** configuration for a route comparison depends on the types of routes that are involved.
      – When comparing BGP VPN routes (with the same Route Distinguisher [RD] or different RDs), the **always-compare-med** configuration in the base router BGP configuration applies.
      – When comparing a BGP PE-CE route with a BGP VPN route that has been imported into a VPRN, the **always-compare-med** configuration in the VPRN BGP configuration applies.
12. Select the route with the lowest owner type (BGP < BGP-label < BGP-VPN).
13. Prefer routes learned from EBGP peers over routes learned from IBGP and confed-EBGP peers.
14. Select the route with the lowest route-table or tunnel-table cost to the NEXT_HOP. This step is skipped if **ignore-nh-metric** is configured, or if the routes are associated with different RIBs. For VPN-IP routes received by a router without any configured VPRN services, next-hop cost is determined from the route-table cost.
15. Select the route with the lowest next-hop type. Resolutions made in the route table are preferred to resolutions made in the tunnel-table. This step is skipped if **ignore-nh-metric** is configured, or if the routes are associated with different RIBs.
16. Select the route received from the peer with the lowest router ID; this comes from the ORIGINATOR_ID attribute (if present) or the BGP identifier of the peer (received in its OPEN message). If **ignore-router-id** is configured, keep the current best path and skip the remaining steps.

17. Select the route with the shortest CLUSTER_LIST length.
18. Select the route received from the peer with the lowest IP address.
19. For VPN-IP routes imported into a VPRN, select the route with the lowest route-distinguisher value.

5.6.2.2 BGP route installation in the route table

Each BGP RIB with IP routes (unlabeled IPv4, labeled-unicast IPv4, unlabeled IPv6, and labeled-unicast IPv6) submits its best path for each prefix to the common IP route table, unless the **disable-route-table-install** command is configured or the **selective-label-ipv4-install** command has prevented the installation. The best path is selected by the BGP decision process. The default preference for BGP routes submitted by the label-IPv4 and label-IPv6 RIBs (these appear in the route table and FIB as having a BGP-LABEL protocol type) can be modified by using the **label-preference** command. The default preference for BGP routes submitted by the unlabeled IPv4 and IPv6 RIBs can be modified by using the **preference** command.

Note:
The BGP instance level **disable-route-table-install** command can be configured on control-plane route reflectors that are not involved in packet forwarding (that is, those that do not modify the BGP next hop). This command improves performance and scalability. The **disable-route-table-install** policy action can be applied to BGP routes matching a peer import policy to conserve FIB space on a router that is in the datapath, for example, a router that should advertise BGP routes with itself as next hop even though it has not installed those routes into its own forwarding table.

The **configure router bgp selective-label-ip no-install** command behaves similarly to the **disable-route-table-install** policy action for BGP-LU routes on a **next-hop-self** router. For information about route table entry installation on BGP-LU routes, see Selective download of labeled unicast routes on next-hop-self routers.

If a BGP RIB has multiple BGP paths for the same IPv4 or IPv6 prefix that qualify as the best path up to a specific point in the comparison process, then a specified number of these multipaths can be submitted to the common IP route table. This is called BGP multipath and must be explicitly enabled using one or more commands in the **multi-path** context. These commands specify the maximum number of BGP paths, including the overall best path, that each BGP RIB can submit to the route table for any particular IPv4 or IPv6 prefix. If ECMP, with a limit of n, is enabled in the base router instance, then up to n paths are selected for installation in the IP FIB. In the data-path, traffic matching the IP route is load-shared across the ECMP next hops based on a per-packet hash calculation.

By default, the hashing is not sticky, meaning that when one or more of the ECMP BGP next hops fail, all traffic flows matching the route are potentially moved to new BGP next hops. If required, a BGP route can be marked (using the **sticky-ecmp** action in route policies) for sticky ECMP behavior so that BGP next hop failures are handled by moving only the affected traffic flows to the remaining next hops as evenly as possible. If new ECMP BGP next hops become available for a marked BGP, then route flows are moved as evenly as possible onto the resultant set of next hops. For more information about sticky ECMP, see BGP support for sticky ECMP.

A BGP route to an IPv4 or IPv6 prefix is a candidate for installation as an ECMP next hop only if it meets all of the following criteria:

- The multipath route must be the same type of route as the best path (same AFI/SAFI and, in some cases, same next-hop resolution method).
- The multipath route must tie with the best path for all criteria of greater significance than next-hop cost, except for criteria that are configured to be ignored.

- If the best path selection reaches the next-hop cost comparison, the multipath route must have the same next-hop cost as the best route, unless **unequal-cost** is configured.
- The multipath route must not have the same BGP next hop as the best path or any other multipath route.
- The multipath route must not cause the ECMP limit of the routing instance to be exceeded. Use the **ecmp** command to configure the ECMP limit of the routing instance.
- The multipath route must not cause the applicable maximum paths limit to be exceeded.
- The multipath route must have the same neighbor AS in its AS path as the best path if **restrict** is set to **same-neighbor-as**.

  By default, any path with the same AS path length as the best path, regardless of the neighbor AS, is eligible to be considered a multipath.
- The route must have the same AS path as the best path if **restrict** is set to **exact-as-path**.

  By default, any path with the same AS path length as the best path, regardless of the AS numbers, is eligible to be considered a multipath.

SR OS also supports IBGP multipath. In some topologies, a BGP next hop is resolved by an IP route that has multiple ECMP next hops. When **ibgp-multipath** is not configured, only one of the ECMP next hops is programmed as the next hop of the BGP route in the IOM. When **ibgp-multipath** is configured, the IOM attempts to use all the ECMP next hops of the resolving route in the forwarding state. Although the name of the **ibgp-multipath** command implies that it is specific to IBGP-learned routes, this is not the case. It also applies to routes learned from any multihop BGP session including routes learned from multihop EBGP peers.

The **multi-path** and **ibgp-multipath** commands are not mutually exclusive and work together. The first context enables ECMP load-sharing across different BGP next hops (corresponding to different BGP routes) while the **ibgp-multipath** enables ECMP load-sharing across the next hops of IP routes that resolve the BGP next hops.

Finally, **ibgp-multipath** does not control traffic load sharing toward a BGP next hop that is resolved by a tunnel, as when dealing with BGP shortcuts or labeled routes (VPN-IP, label-IPv4, or label-IPv6). When a BGP next hop is resolved by a tunnel that supports ECMP, the load-sharing of traffic across the ECMP next hops of the tunnel is automatic.

SR OS supports direct resolution of a BGP next hop to multiple RSVP-TE or SR-TE tunnels. In addition, a BGP next hop can be resolved by multiple LDP ECMP next hops that each correspond to a separate LDP-over-RSVP or LDP-over-SRTE tunnel. It is also possible for a BGP next hop to be resolved by an IGP shortcut route that has multiple RSVP-TE or SR-TE tunnels as its ECMP next hops.

5.6.2.3 BGP support for sticky ECMP

Each sticky ECMP route uses 64 distribution buckets to apportion flows onto the available next hops. Figure 22: Sticky ECMP flow distribution as next hops are removed part 1, Figure 23: Sticky ECMP flow distribution as next hops are removed part 2, and Figure 24: Sticky ECMP flow distribution as next hops are removed part 3 provide an example of the distribution of flows over multiple BGP next hops as next hops are removed.

Figure 22: Sticky ECMP flow distribution as next hops are removed part 1

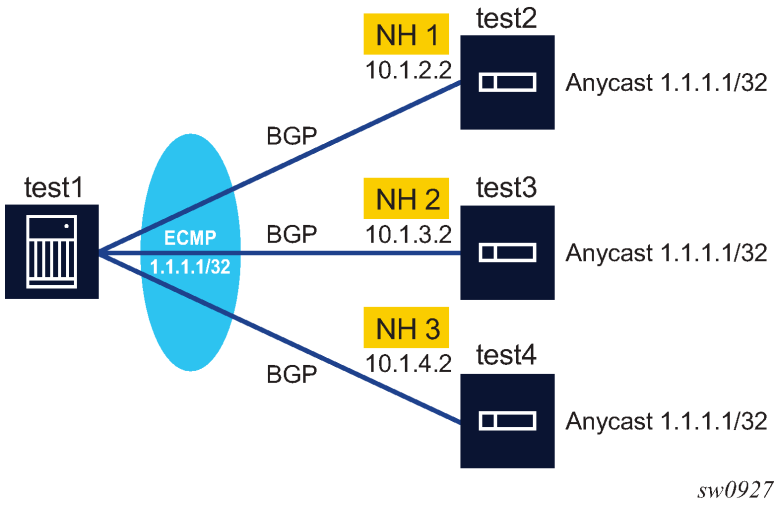


Figure 23: Sticky ECMP flow distribution as next hops are removed part 2

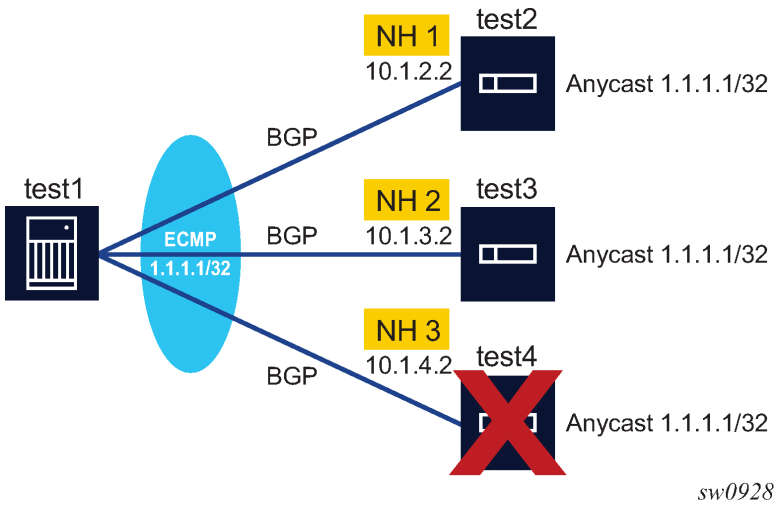


Figure 24: Sticky ECMP flow distribution as next hops are removed part 3

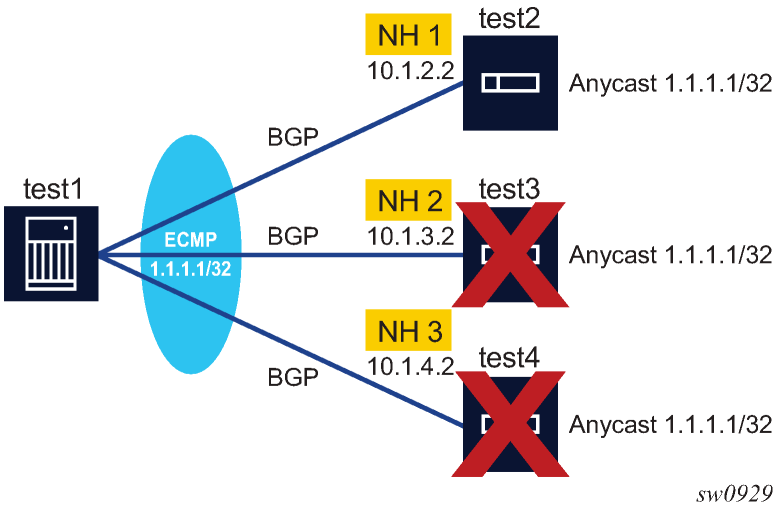


Table 5: Sticky ECMP flow distribution as next hops are removed for 1.1.1.1/32 lists the sticky ECMP flow distribution as next hops are removed for 1.1.1.1/32.

Table 5: Sticky ECMP flow distribution as next hops are removed for 1.1.1.1/32

Initial sticky ECMP distribution for 1.1.1.1/32 in Figure 22: Sticky ECMP flow distribution as next hops are removed part 1		**ECMP distribution for 1.1.1.1/32 if next hop 3 fails in Figure 23: Sticky ECMP flow distribution as next hops are removed part 2**		**ECMP distribution for 1.1.1.1/32 if next hop 2 subsequently fails in Figure 24: Sticky ECMP flow distribution as next hops are removed part 3**	
Bucket	**NH**	**Bucket**	**NH**	**Bucket**	**NH**
00	1	00	1	00	1
01	2	01	2	01	1
02	3	02	1	02	1
03	1	03	1	03	1
04	2	04	2	04	1
05	3	05	2	05	1
06	1	06	1	06	1
07	2	07	2	07	1
08	3	08	1	08	1
09	1	09	1	09	1

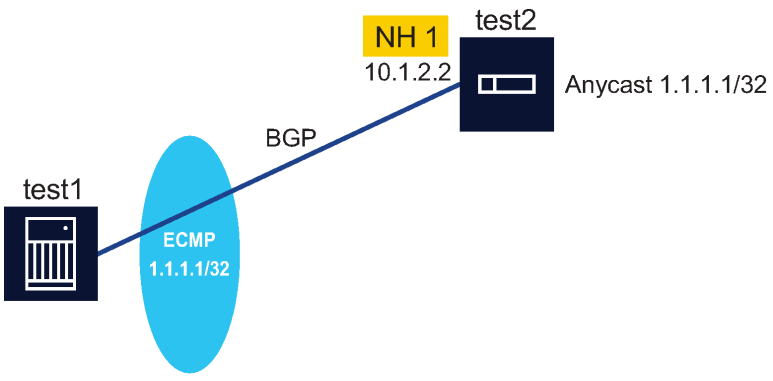
Initial sticky ECMP distribution for 1.1.1.1/32 in Figure 22: Sticky ECMP flow distribution as next hops are removed part 1		ECMP distribution for 1.1.1.1/32 if next hop 3 fails in Figure 23: Sticky ECMP flow distribution as next hops are removed part 2		ECMP distribution for 1.1.1.1/32 if next hop 2 subsequently fails in Figure 24: Sticky ECMP flow distribution as next hops are removed part 3	
Bucket	**NH**	**Bucket**	**NH**	**Bucket**	**NH**
10	2	10	2	10	1
11	3	11	2	11	1
12	1	12	1	12	1
13	2	13	2	13	1
14	3	14	1	14	1
15	1	15	1	15	1
16	2	16	2	16	1
17	3	17	2	17	1
18	1	18	1	18	1
19	2	19	2	19	1
20	3	20	1	20	1
21	1	21	1	21	1
22	2	22	2	22	1
23	3	23	2	23	1
24	1	24	1	24	1
25	2	25	2	25	1
26	3	26	1	26	1
27	1	27	1	27	1
28	2	28	2	28	1
29	3	29	2	29	1
30	1	30	1	30	1
31	2	31	2	31	1
32	3	32	1	32	1

Initial sticky ECMP distribution for 1.1.1.1/32 in Figure 22: Sticky ECMP flow distribution as next hops are removed part 1		ECMP distribution for 1.1.1.1/32 if next hop 3 fails in Figure 23: Sticky ECMP flow distribution as next hops are removed part 2		ECMP distribution for 1.1.1.1/32 if next hop 2 subsequently fails in Figure 24: Sticky ECMP flow distribution as next hops are removed part 3	
Bucket	**NH**	**Bucket**	**NH**	**Bucket**	**NH**
33	1	33	1	33	1
34	2	34	2	34	1
35	3	35	2	35	1
36	1	36	1	36	1
37	2	37	2	37	1
38	3	38	1	38	1
39	1	39	1	39	1
40	2	40	2	40	1
41	3	41	2	41	1
42	1	42	1	42	1
43	2	43	2	43	1
44	3	44	1	44	1
45	1	45	1	45	1
46	2	46	2	46	1
47	3	47	2	47	1
48	1	48	1	48	1
49	2	49	2	49	1
50	3	50	1	50	1
51	1	51	1	51	1
52	2	52	2	52	1
53	3	53	2	53	1
54	1	54	1	54	1
55	2	55	2	55	1

Initial sticky ECMP distribution for 1.1.1.1/32 in Figure 22: Sticky ECMP flow distribution as next hops are removed part 1		ECMP distribution for 1.1.1.1/32 if next hop 3 fails in Figure 23: Sticky ECMP flow distribution as next hops are removed part 2		ECMP distribution for 1.1.1.1/32 if next hop 2 subsequently fails in Figure 24: Sticky ECMP flow distribution as next hops are removed part 3	
Bucket	**NH**	**Bucket**	**NH**	**Bucket**	**NH**
56	3	56	1	56	1
57	1	57	1	57	1
58	2	58	2	58	1
59	3	59	2	59	1
60	1	60	1	60	1
61	2	61	2	61	1
62	3	62	1	62	1
63	1	63	1	63	1

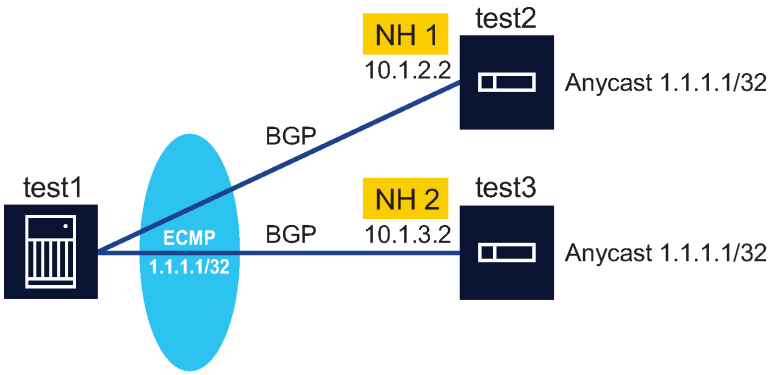
Figure 25: Sticky ECMP flow distribution as next hops are added part 1, Figure 26: Sticky ECMP flow distribution as next hops are added part 2, and Figure 27: Sticky ECMP flow distribution as next hops are added part 3 provide an example of the distribution of flows over multiple BGP next hops as next hops are added.

Figure 25: Sticky ECMP flow distribution as next hops are added part 1



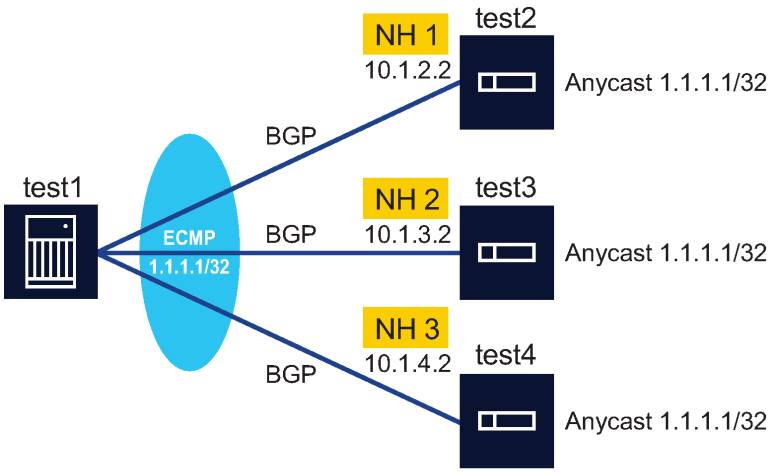
sw0930

Figure 26: Sticky ECMP flow distribution as next hops are added part 2



sw0931

Figure 27: Sticky ECMP flow distribution as next hops are added part 3



sw0932

The following table lists the sticky ECMP flow distribution as next hops are added for 1.1.1.1/32:

Table 6: Sticky ECMP flow distribution as next hops are added for 1.1.1.1/32

Initial sticky ECMP distribution for 1.1.1.1/32 in Figure 25: Sticky ECMP flow distribution as next hops are added part 1		ECMP distribution for 1.1.1.1/32 if next hop 2 becomes available in Figure 26: Sticky ECMP flow distribution as next hops are added part 2		ECMP distribution for 1.1.1.1/32 if next hop 3 additionally becomes available in Figure 27: Sticky ECMP flow distribution as next hops are added part 3	
Bucket	**NH**	**Bucket**	**NH**	**Bucket**	**NH**
00	1	00	1	00	1
01	1	01	2	01	2
02	1	02	1	02	3
03	1	03	2	03	2
04	1	04	1	04	1
05	1	05	2	05	3
06	1	06	1	06	1
07	1	07	2	07	2
08	1	08	1	08	3
09	1	09	2	09	2
10	1	10	1	10	1
11	1	11	2	11	3
12	1	12	1	12	1
13	1	13	2	13	2
14	1	14	1	14	3
15	1	15	2	15	2
16	1	16	1	16	1
17	1	17	2	17	3
18	1	18	1	18	1
19	1	19	2	19	2
20	1	20	1	20	3
21	1	21	2	21	2

Initial sticky ECMP distribution for 1.1.1.1/32 in Figure 25: Sticky ECMP flow distribution as next hops are added part 1		ECMP distribution for 1.1.1.1/32 if next hop 2 becomes available in Figure 26: Sticky ECMP flow distribution as next hops are added part 2		ECMP distribution for 1.1.1.1/32 if next hop 3 additionally becomes available in Figure 27: Sticky ECMP flow distribution as next hops are added part 3	
Bucket	**NH**	**Bucket**	**NH**	**Bucket**	**NH**
22	1	22	1	22	1
23	1	23	2	23	3
24	1	24	1	24	1
25	1	25	2	25	2
26	1	26	1	26	3
27	1	27	2	27	2
28	1	28	1	28	1
29	1	29	2	29	3
30	1	30	1	30	1
31	1	31	2	31	2
32	1	32	1	32	3
33	1	33	2	33	2
34	1	34	1	34	1
35	1	35	2	35	3
36	1	36	1	36	1
37	1	37	2	37	2
38	1	38	1	38	3
39	1	39	2	39	2
40	1	40	1	40	1
41	1	41	2	41	3
42	1	42	1	42	1
43	1	43	2	43	2
44	1	44	1	44	3

Initial sticky ECMP distribution for 1.1.1.1/32 in Figure 25: Sticky ECMP flow distribution as next hops are added part 1		ECMP distribution for 1.1.1.1/32 if next hop 2 becomes available in Figure 26: Sticky ECMP flow distribution as next hops are added part 2		ECMP distribution for 1.1.1.1/32 if next hop 3 additionally becomes available in Figure 27: Sticky ECMP flow distribution as next hops are added part 3	
Bucket	**NH**	**Bucket**	**NH**	**Bucket**	**NH**
45	1	45	2	45	2
46	1	46	1	46	1
47	1	47	2	47	3
48	1	48	1	48	1
49	1	49	2	49	2
50	1	50	1	50	3
51	1	51	2	51	2
52	1	52	1	52	1
53	1	53	2	53	3
54	1	54	1	54	1
55	1	55	2	55	2
56	1	56	1	56	3
57	1	57	2	57	2
58	1	58	1	58	1
59	1	59	2	59	3
60	1	60	1	60	1
61	1	61	2	61	2
62	1	62	1	62	3
63	1	63	2	63	2

5.6.2.4 Weighted ECMP for BGP routes

In some cases, the ECMP BGP next-hops of an IP route correspond to paths with very different bandwidths and it makes sense for the ECMP load-balancing algorithm to distribute traffic across the BGP next-hops in proportion to their relative bandwidths. The bandwidth associated with a path can be signaled to other BGP routers by including a link-bandwidth extended community in the BGP route. The

link-bandwidth extended community is optional and non-transitive and encodes an autonomous system (AS) number and a bandwidth.

The SR OS implementation supports the link-bandwidth extended community in routes associated with the following address families: IPv4, IPv6, label-IPv4, label-IPv6, VPN-IPv4, and VPN-IPv6. The router automatically performs weighted ECMP for an IP BGP route when all of the ECMP BGP next-hops of the route include a link-bandwidth extended community. The relative weight of traffic sent to each BGP next-hop is visible in the output of the **show router route-table extensive** and **show router fib extensive** commands.

A route with a link-bandwidth extended community can be received from any IBGP peer. If such a route is received from an EBGP peer, the link-bandwidth extended community is stripped from the route unless an **accept-from-ebgp** command applies to that EBGP peer. However, a link-bandwidth extended community can be added to routes received from a directly connected (single hop) EBGP peer, potentially replacing the received Extended Community. This is accomplished using the **add-to-received-ebgp** command, which is available in group and neighbor configuration contexts.

When a route with a link-bandwidth extended community is advertised to an EBGP peer, the link-bandwidth extended community is removed by default. However, transitivity across an AS boundary can be allowed by configuring the **send-to-ebgp** command.

When a route with a link-bandwidth extended community is advertised to a peer using next-hop-self, the Extended Community is usually removed if it was not added locally (that is, by policy or **add-to-received-ebgp** command). However, in the special case that a route is readvertised (with next-hop-self) toward a peer covered by the scope of an **aggregate-used-paths** command, and the re-advertising router has installed multiple ECMP paths toward the destination each associated with a link-bandwidth extended community, the route is readvertised with a link-bandwidth extended community encoding the total bandwidth of all the used multipaths.

The link-bandwidth extended community associated with a BGP route can be displayed using the **show router bgp routes** command. For the bandwidth value, the system automatically converts the binary value in the extended community to a decimal number in units of Mb/s (1 000 000 b/s).

Weighted ECMP across the BGP next-hops of an IP BGP route is supported in combination with ECMP at the level of the route or tunnel that resolves one or more of the ECMP BGP next-hops. This ECMP at the resolving level can also be weighted ECMP when the following conditions all apply:

- the BGP next-hop is resolved by an IP route (OSPF, IS-IS, or static) with MPLS LSP ECMP next-hops
- **ibgp-multipath** is configured under BGP
- **config**>**router**>**weighted-ecmp** is configured

5.6.2.5 BGP route installation in the tunnel table

Received label-unicast routes can be installed by BGP as tunnels in the tunnel table. In SR OS, the tunnel table is used to resolve a BGP next-hop to a tunnel when required by the configuration or the type of route (see Next-hop resolution). BGP tunnels play a key role in the following solutions:

- inter-AS model C
- carrier supporting carrier (CSC)
- seamless MPLS

BGP tunnels have a preference of 10 in the tunnel table, compared to 9 for LDP tunnels and 7 for RSVP tunnels. If the router configuration allows all types of tunnels to resolve a BGP next-hop, an RSVP LSP is preferred over an LDP tunnel, and an LDP tunnel is preferred over a BGP tunnel.

Further details about BGP-LU tunnels depending on the address family, are described below.

5.6.2.5.1 Label-IPv4 tunnels

A label-IPv4 is automatically added as a BGP tunnel entry to the tunnel table if all of the following conditions are met.

- The label-IPv4 route is the best BGP path for the /32 IPv4 prefix.
- The label-IPv4 route has the numerically lowest preference value among all routes (regardless of the protocol) for the /32 IPv4 prefix.
- The **disable-route-table-install** command does not apply to the route.
- The **selective-label-ipv4-install** command does not prevent the installation of the route.

If multipath and ECMP are configured so that they apply to label IPv4 routes, then a BGP tunnel can be installed in the tunnel table with multiple ECMP next-hops, each one corresponding to a path through a different BGP next-hop. The multipath selection process described in BGP route installation in the route table also applies to this case.

5.6.2.5.2 Label-IPv6 tunnels

A label-IPv6 is automatically added as a BGP tunnel entry to the tunnel table if all of the following conditions are met.

- The label-IPv6 route is the best BGP path for a /128 IPv6 prefix is a label-IPv6 route (AFI 2, SAFI 4).
- The label-IPv6 route has the numerically lowest preference value among all routes (regardless of protocol) for the /128 IPv6 prefix.
- The **disable-route-table-install** command does not apply to the route.
- The **disable-explicit-null** command is configured.

If multipath and ECMP are configured so that they apply to label IPv6 routes, a BGP tunnel can be installed in the tunnel table with multiple ECMP next-hops, each one corresponding to a path through a different BGP next-hop. However, when **disable-explicit-null** is configured, the label-IPv6 routes used for ECMP toward an IPv6 destination cannot be a mix of routes with regular label values and routes with special (IPv6 explicit null) label values.

5.6.2.6 Selective download of labeled unicast routes on next-hop-self routers

When a router is configured to perform **next-hop-self** for labeled unicast routes, the router must swap and advertise learned labeled unicast routes to downstream routers because there is no way for next-hop-self routers to know which BGP-LU routes are in use by downstream routers at a specific time. Additionally, a new service may resolve to a new labeled unicast route on a downstream router at any time. A **next-hop-self** router is required to readvertise all learned labeled routes downstream after best route selection.

A **next-hop-self** router may have local services that resolve to a subset of learned labeled unicast routes. In this case, it is possible to enable selective download of labeled unicast routes to datapath tables. Only the labeled unicast routes used by local services can be downloaded to the datapath.

Use the following command to install labeled unicast routes in local datapath tables that are used for NHLFE resolution by local services.

```
configure router bgp selective-label-ip no-install
```

This process is dynamic. When additional labeled unicast routes are called by services, they are automatically downloaded to the datapath tables. After a labeled unicast route is downloaded to a datapath, the route can also be used by IP shortcuts for next-hop resolution.

Use the following command to install labeled unicast routes to the RTM for NHLFE resolution by IP shortcuts and continue with selective install for local services. This option is used to extend the **no-install** mode via downloading all the labeled unicast routes to the RTM for IP next-hop resolution. The RTM is fully populated by labeled unicast routes, and services continue to trigger the download of only the required labeled unicast routes to the LTN.

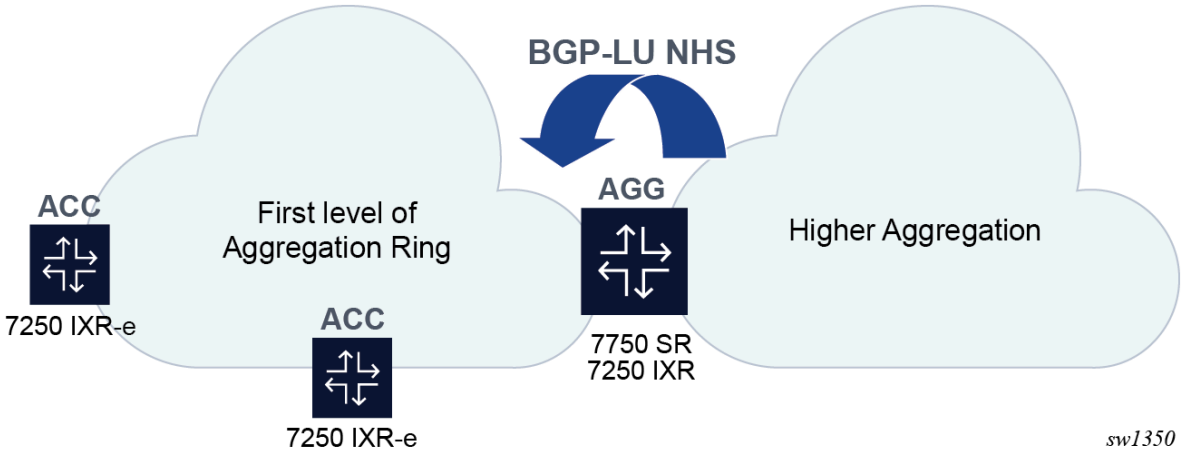
```
configure router bgp selective-label-ip route-table-install-only
```

Labeled unicast routes required for NHLFE resolution are handled selectively. That is, local services that require resolution to a labeled unicast route trigger the download.

The **selective-label-ip** command does not impact how labeled unicast routes are readvertised downstream.

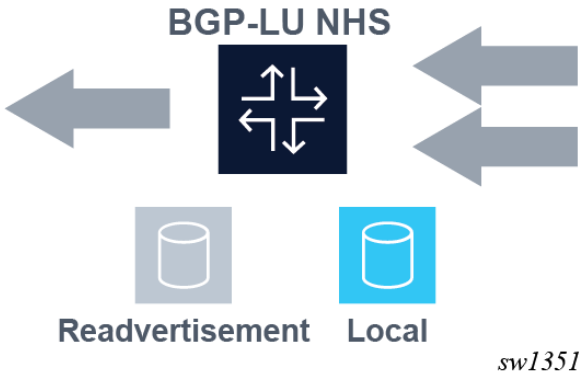
In the following figure, the 7750 SR performs **next-hop-self** for labeled unicast routes and readvertises all learned labeled unicast routes to access the network (such as ACC routers, including the 7250 IXR-e).

Figure 28: Next-hop-self for BGP-LU routes



The following figure shows the logical view of the BGP-LU NHS component from the example in the preceding figure.

Figure 29: Next-hop-self for BGP-LU routes (detailed view)



Enabling **selective-label-ip** install saves datapath table space that can be used by other applications.

Note: This command is supported for labeled unicast for both IPv4 and IPv6 routes.

5.6.2.7 BGP fast reroute

BGP fast reroute is a feature that brings together indirection techniques in the forwarding plane and pre-computation of BGP backup paths in the control plane to support fast reroute of BGP traffic around unreachable/failed BGP next-hops. BGP fast reroute is supported with IPv4, label-IPv4, IPv6, label-IPv6, VPN-IPv4 and VPN-IPv6 routes. The scenarios supported by the base router BGP context are described in Table 7: BGP fast reroute scenarios (base context).

See the VPRN section of the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 3 Services Guide: IES and VPRN* for more information about BGP fast reroute information specific to IP VPNs.

Table 7: BGP fast reroute scenarios (base context)

Ingress packet	Primary route	Backup route	Prefix independent convergence
IPv4	IPv4 route with next-hop A resolved by an IPv4 route or any shortcut tunnel	IPv4 route with next-hop B resolved by an IPv4 route or any shortcut tunnel	Yes
IPv4	Label-IPv4 route with next-hop A resolved by any transport tunnel	Label-IPv4 route with next-hop B resolved by any transport tunnel	Yes
IPv4	Label-IPv4 route with next-hop A resolved by a local route	Label-IPv4 route with next-hop B resolved by a local route	Yes
IPv4	Label-IPv4 route with next-hop A resolved by a static route	Label-IPv4 route with next-hop B resolved by a static route	Yes
IPv6	IPv6 route with next-hop A resolved by an IPv6 route	IPv6 route with next-hop B resolved by an IPv6 route	Yes

Ingress packet	Primary route	Backup route	Prefix independent convergence
IPv6	Label-IPv6 route with next-hop A resolved by any transport tunnel	Label-IPv6 route with next-hop B resolved by any transport tunnel	Yes
IPv6	Label-IPv6 route with next-hop A resolved by a local route	Label-IPv6 route with next-hop B resolved by a local route	Yes
IPv6	Label-IPv6 route with next-hop A resolved by a static route	Label-IPv6 route with next-hop B resolved by a static route	Yes

5.6.2.7.1 Calculating backup paths

In SR OS, fast reroute is optional and must be enabled by using either the BGP **backup-path** command or the route-policy **install-backup-path** command. Typically, only one approach is used.

The **backup-path** command in the base router context is used to control fast reroute on a per-RIB basis (IPv4, label-IPv4, IPv6, and label-IPv6). When the command specifies a particular family, BGP attempts to find a backup path for every prefix learned by the associated BGP RIB.

The **install-backup-path** command, available in route-policy-action contexts, marks a BGP route as requesting a backup path. It only takes effect in BGP import and VRF import policies. If only some prefixes should have backup paths, then the **backup-path** command should not be used, and instead the **install-backup-path** command should be used to mark only those prefixes that require extra protection.

In general, a prefix supports ECMP paths or a backup path, but not both. The backup path is the best path after the primary path and any paths with the same BGP next-hop as the primary path have been removed.

5.6.2.7.2 Failure detection and switchover to the backup path

When BGP fast reroute is enabled the IOM reroutes traffic onto a backup path based on input from BGP. When BGP decides that a primary path is no longer usable it notifies the IOM and affected traffic is immediately switched to the backup path.

The following events trigger failure notifications to the IOM and reroute of traffic to backup paths.

- The peer IP address is unreachable and the peer-tracking is enabled.
- The BFD session associated with BGP peer goes down.
- The BGP session terminated with peer (for example, send/receive NOTIFICATION).
- There is no longer any route (allowed by the next-hop resolution policy, if configured) that can resolve the BGP next-hop address.
- The LDP tunnel that resolves the next-hop goes down. This could happen because there is no longer any IP route that can resolve the FEC, or the LDP session goes down, or the LDP peer withdraws its label mapping.
- The RSVP tunnel that resolves the next-hop goes down. This could happen because a ResvTear message is received, or the RESV state times out, or the outgoing interface fails and is not protected by FRR or a secondary path.
- The BGP tunnel that resolves the next-hop goes down. This could happen because the BGP label-IPv4 route is withdrawn by the peer or else becomes invalid because of an unresolved next-hop.

5.6.2.8 QoS policy propagation through BGP

The QoS Policy Propagation through BGP (QPPB) is a feature that allows a QoS treatment (forwarding class and optionally priority) to be associated with a BGP IPv4, label-IPv4, IPv6, or label-IPv6 route that is installed in the routing table. This is done so that when traffic arrives on a QPPB-enabled IP interface and its source or destination IP address matches a BGP route with QoS information, the packet is handled according to the QoS of the matching route. SR OS supports QPPB on the following types of interfaces:

- base router network interfaces
- IES and VPRN SAP interfaces
- IES and VPRN spoke SDP interfaces
- IES and VPRN subscriber interfaces

QPPB is enabled on an interface using the **qos-route-lookup** command. There are separate commands for IPv4 and IPv6 so QPPB can be enabled in one mode, source, destination, or none, for IPv4 packets arriving on the interface, and a different mode source, destination, or none, for IPv6 packets arriving on the interface.

Note: The source-based QPPB is not supported on subscriber interfaces.

Different BGP routes for the same IP prefix can be associated with different QPPB information. If these BGP routes are combined in support of ECMP or BGP fast reroute then the QPPB information becomes next-hop specific. If these LOC-RIB routes are combined in support of ECMP or BGP fast reroute then the QPPB information becomes next-hop specific. This means that in destination QPPB mode the QoS assigned to a packet depends on the BGP next-hop that is selected for that particular packet by the ECMP hash or fast reroute algorithm. In source QPPB mode the QoS assigned to a packet comes from the first BGP next-hop of the IP route matching the source address.

5.6.2.9 BGP policy accounting and policing

Policy accounting is a feature that allows "classes" to be associated with specific IPv4 or IPv6 routes, static or BGP learned, when they are installed in the routing table. This is done for the following reasons.

- To collect "per-interface, per-class" traffic statistics on policy accounting-enabled interfaces of the router. This is supported by all FP2 and later generation cards and systems.
- To implement "per-interface, per-FP, per-class" traffic policing on policy accounting-enabled interfaces of the router. This is only supported for destination classes and only by FP4 cards and systems. The rate limit is applied per-interface, per-FP, per-class, when the IP interface is a distributed interface such as R-VPLS, LAG, or spoke SDP, that spans multiple complexes. Otherwise, for a simple interface, the rate limit is applied per-interface per-class.

For both applications the following IP interface types are supported:

- base router network interfaces
- IES and VPRN SAP interfaces
- IES and VPRN spoke SDP interfaces
- IES and VPRN subscriber interfaces (with some limitations)
- IES and VPRN R-VPLE interfaces

Policy accounting, and policing (if needed and supported), is enabled on an interface using the **policy-accounting** command. The name of a policy accounting template must be specified as an argument of this command. SR OS supports up to 1024 different templates. Each policy accounting template can have a list of source classes (up to 255), a list of destination classes (up to 255), and a list of policers (up to 63). Each source class, destination class, and policer, in their respective list, has an index number. Source class indexes and destination class indexes have a global meaning. In other words, **destination-class** index 5 in one template refers to the same set of routes as **destination-class** index 5 in another policy accounting template. Policer indexes have a local scope to the enclosing template. In one template, **destination-class** index 5 could use policer index 2 and in another template **destination-class** index 5 could use policer index 62. If a destination class has an associated policer then incoming traffic on each IP interface on which the template is applied is rate-limited based on that policer if the destination IP address matches a route with that destination class.

Policy accounting templates containing one or more source class identifiers cannot be applied to subscriber interfaces.

The policy accounting template tells the IOM the number of statistics and policer resources to use for each interface. These resources are derived from two pools that are sized per-FP. The first pool consists of policer statistics indexes. Every policy-accounting interface on a card or FP uses one of these resources for every source and destination class index listed in the template referenced by the interface. These are basic resources needed for statistics collection. The total reservation at the FP level is set using the **configure card** *slot-number* **fp** *fp-number* **policy-accounting** command.

The second pool (FP4 cards only) consists of policer index resources. Every policy-accounting interface on a complex uses one of these resources for every destination class associated with a policer in the template referenced by the interface. The total reservation of this second resource at the FP level is set using the **configure card** *slot-number* **fp** *fp-number* **ingress** policy-accounting policers command.

The total number of the above two resources, per FP, must be less than or equal to 128000. In addition, the second resource pool size must be less than or equal to the size of the first resource pool.

It is possible to increase or decrease the size of either resource sub-pool at any time. A decrease can cause some interfaces (randomly selected) to immediately lose their resources and stop counting or policing some traffic that was previously being counted or policed.

If the policy accounting is enabled on a spoke SDP or R-VPLS interface, all FPs in the system should have a reservation for each of the above resources, otherwise the **show router interface policy-accounting** command output reports that the statistics are possibly incomplete.

Through route policy or configuration mechanisms, a BGP or static route for an IP prefix can have a source class index (1 to 255), a destination class index (1 to 255) or both. When an ingress packet on a policy accounting-enabled interface [I1] is forwarded by the IOM and its destination address matches a BGP or static route with a destination class index [D], and [D] is listed in the relevant policy accounting template, then the packets-forwarded and IP-bytes-forwarded counters for [D] on interface [I1] are incremented accordingly. If [D] is also associated with a policer (FP4 only) the packet is also subjected to rate limiting as discussed above. The policer statistics displayed by the **show router interface policy-accounting** command include Layer 2 encapsulation and is different from the destination-class byte-level statistics.

When an ingress packet on a policy accounting-enabled interface [I2] is forwarded by the IOM and its source address matches a BGP or static route with a source class index [S], and [S] is listed in the relevant policy accounting template, the packets-forwarded and IP-bytes-forwarded counters for [S] on interface [I2] are incremented accordingly. Policing based on the source class is unsupported.

It is possible that different BGP or static routes for the same IP prefix (through different next hops) are associated with different class information. If these routes are combined in support of ECMP or fast reroute then the destination class of a packet depends on the next hop that is selected for that particular packet by

the ECMP hash or fast reroute algorithm. If the source address of a packet matches a route with multiple next hops its source class is derived from the first next hop of the matching route.

5.6.2.10 Route flap damping

Route Flap Damping (RFD) is a mechanism supported by 7450, 7750, and 7950 routers, as well as other BGP routers, that was designed to help improve the stability of Internet routing by mitigating the impact of route flaps. Route flaps describe a situation where a router alternately advertises a route as reachable and then unreachable or as reachable through one path and then another path in rapid succession. Route flaps can result from hardware errors, software errors, configuration errors, unreliable links, and so on. However not all perceived route flaps represent a true problem; when a best path is withdrawn the next-best path may not be immediately known and may trigger a number of intermediate best path selections (and corresponding advertisements) before it is found. These intermediate best path selections may travel at different speeds through different routers because of the effect of the min-route-advertisement interval (MRAI) and other factors. RFD does not handle this type of situation particularly well and for this and other reasons many Internet service providers do not use RFD.

In SR OS route flap damping is configurable; by default, it is disabled. It can be enabled on EBGP and confed-EBGP sessions by including the **damping** command in their group or neighbor configuration. The **damping** command has no effect on IBGP sessions. When a route of any type (any AFI/SAFI) is received on a non-IBGP session that has **damping** enabled.

- If the route changes from reachable to unreachable because of a withdrawal by the peer then damping history is created for the route (if it does not already exist) and in that history the Figure of Merit (FOM), an accumulated penalty value, is incremented by 1024.
- If a reachable route is updated by the peer with new path attribute values then the FOM is incremented by 1024.
- In SR OS the FOM has a hard upper limit of 21540 (not configurable).
- The FOM value is decayed exponentially as described in RFC 2439. The **half-life** of the decay is 15 minutes by default, however a BGP import policy can be used to apply a non-default damping profile to the route, and the **half-life** in the non-default damping profile can have any value between 1 and 45 minutes.
- The FOM value at the last time of update can be displayed using the **show router bgp damping detail** command. The time of last update can be up to 640 seconds ago; SR OS does not calculate the current FOM every time the show command is entered.
- When the FOM reaches the suppress limit, which is 3000 by default, but can be changed to any value between 1 and 20000 in a non-default damping profile, the route is suppressed, meaning it is not used locally and not advertised to peers. The route remains suppressed until either the FOM exponentially decays to a value less than or equal to the **reuse** threshold or the **max-suppress** time is reached. By default, the **reuse** threshold is 750 and the **max-suppress** time is 60 minutes, but these can be changed in a non-default damping profile: **reuse** can have a value between 1 and 20000 and **max-suppress** can have a value between 1 and 720 minutes.

5.6.3 RIB-OUT features

SR OS implements the following features related to RIB-OUT processing:

- BGP export policies
- outbound route filtering (ORF)

- RT constrained route distribution
- configurable min-route-advertisement (MRAI)
- advertise-inactive
- best-external
- add-path
- split-horizon

These features are discussed in the following sections.

5.6.3.1 BGP export policies

The **export** command is used to apply one or more policies (up to 15) to a neighbor, group or to the entire BGP context. The **export** command that is most-specific to a peer is the one that is applied. An **export** policy command applied at the **neighbor** level takes precedence over the same command applied at the **group** or global level. An **export** policy command applied at the **group** level takes precedence over the same command specified on the global level. The **export** policies applied at different levels are not cumulative. The policies listed in an **export** command are evaluated in the order in which they are specified.

Note: The **export** command can reference a policy before it has been created (as a **policy-statement**).

The most common uses for BGP export policies are as follows.

- To locally originate a BGP route by exporting (or redistributing) a non-BGP route that is installed in the route table and actively used for forwarding. The non-BGP route is most frequently a direct, static or aggregate route (exporting IGP routes into BGP is generally not recommended).
- To block the advertisement of specific BGP routes toward specific BGP peers. The routes may be blocked on the basis of IP prefix, communities, and so on.
- To modify the attributes of BGP routes advertised to specific BGP peers. The following path attribute modifications are possible using BGP export policies.
  - Change the ORIGIN value.
  - Add a sequence of AS numbers to the start of the AS_PATH. When a route is advertised to an EBGP peer the addition of the local-AS/global-AS numbers to the AS_PATH is always the final step (done after export policy).
  - Replace the AS_PATH with a new AS_PATH. When a route is advertised to an EBGP peer the addition of the local-AS/global-AS numbers to the AS_PATH is always the final step (done after export policy).
  - Prepend an AS number multiple times to the start of the AS_PATH. When a route is advertised to an EBGP peer the addition of the local-AS/global-AS numbers to the AS_PATH is always the final step (done after export policy). The add/replace action on the AS_PATH supersedes the prepend action if both are specified in the same policy entry.
  - Change the NEXT_HOP to a specific IP address. When a route is advertised to an EBGP peer the next-hop cannot be changed from the local-address.
  - Change the NEXT_HOP to the local-address used with the peer (next-hop-self).
  - Add a value to the MED. If the MED attribute does not exist it is added.

- Subtract a value from the MED. If the MED attribute does not exist it is added with a value of 0. If the result of the subtraction is a negative number the MED metric is set to 0.
- Set the MED to a particular value.
- Set the MED to the cost of the IP route (or tunnel) used to resolve the BGP next-hop.
- Set LOCAL_PREF to a particular value when advertising to an IBGP peer.
- Add, remove or replace standard communities.
- Add, remove or replace extended communities.
- Add a static value to the AIGP metric when advertising the route to an AIGP-enabled peer with a modified BGP next-hop. The static value is incremental to the automatic adjustment of the LOC-RIB AIGP metric to reflect the distance between the local router and the received BGP next-hop.
- Increment the AIGP metric by a fixed amount when advertising the route to an AIGP-enabled peer with a modified BGP next-hop. The static value is a substitute for the dynamic value of the distance between the local router and the received BGP next-hop.

The **vpn-apply-export** command allows BGP export policies to match VPN-IP routes. When this command is enabled, a VPN-IP route can match both a VRF export policy entry and a BGP export policy entry. VRF export policies can add VPN-IP routes to the Base BGP process, but cannot remove VPN-IP routes from the LOC-Rib or RIB-OUT. A VPN-IP route received from another BGP peer can be filtered by BGP export policies only while **vpn-apply-export** is enabled.

5.6.3.2 Outbound route filtering

Outbound Route Filtering (ORF) is a mechanism that allows one router, the ORF-sending router to signal to a peer, the ORF-receiving router, a set of route filtering rules (ORF entries) that the ORF-receiving router should apply to its route advertisements toward the ORF-sending router. The ORF entries are encoded in Route Refresh messages.

The use of ORF on a session must be negotiated — that is, both routers must advertise the ORF capability in their Open messages. The ORF capability describes the address families that support ORF, and for each address family, the ORF types that are supported and the ability to send/receive each type. 7450, 7750, and 7950 routers support ORF type 3, which is ORF based on Extended Communities. It is supported for only the following address families:

- VPN-IPv4
- VPN-IPv6
- MVPN-IPv4
- MVPN-IPv6

In SR OS the send/receive capability for ORF type 3 is configurable (with the **send-orf** and **accept-orf** commands) but the setting applies to all supported address families.

SR OS support for ORF type 3 allows a PE router that imports VPN routes with a particular set of Route Target Extended Communities to indicate to a peer (for example a route reflector) that it only wants to receive VPN routes that contain one or more of these Extended Communities. When the PE router wants to inform its peer about a new RT Extended Community it sends a Route Refresh message to the peer containing an ORF type 3 entry instructing the peer to *add* a *permit* entry for the 8-byte extended community value. When the PE router wants to inform its peer about a RT Extended Community that is no longer needed it sends a Route Refresh message to the peer containing an ORF type 3 entry instructing the peer to *remove* the *permit* entry for the 8-byte extended community value.

In SR OS the type-3 ORF entries that are sent to a peer can be generated dynamically (if no Route Target Extended Communities are specified with the **send-orf** command) or else specified statically. Dynamically generated ORF entries are based on the route targets that are imported by all locally-configured VPRNs.

A router that has installed ORF entries received from a peer can still apply BGP export policies to the session. If the evaluation of a BGP export policy results in a reject action for a VPN route that matches a permit ORF entry the route is not advertised (that is, the export policy has the final word).

Note: The SR OS implementation of ORF filtering is very efficient. It takes less time to filter a large number of VPN routes with ORF than it does to reject non-matching VPN routes using a conventional BGP export policy.

Despite the advantages of ORF compared to manually configured BGP export policies a better technology, when it comes to dynamic filtering based on Route Target Extended Communities, is RT Constraint. RT Constraint is discussed further in the next section.

5.6.3.3 RT constrained route distribution

RT constrained route distribution, or RT-constrain for short, is a mechanism that allows a router to advertise to specific peers a special type of MP-BGP route called an RTC route; the associated AFI is 1 and the SAFI is 132. The NLRI of an RTC route encodes an Origin AS and a Route Target Extended Community with prefix-type encoding (for instance, if there is a prefix-length and "host" bits after the prefix-length are set to zero). A peer receiving RTC routes does not advertise VPN routes to the RTC-sending router unless they contain a Route Target Extended Community that matches one of the received RTC routes. As with any other type of BGP route RTC routes are propagated loop-free throughout and between Autonomous Systems. If there are multiple RTC routes for the same NLRI the BGP decision process selects one as the best path. The propagation of the best path installs RIB-OUT filter rules as it travels from one router to the next and this process creates an optimal VPN route distribution tree rooted at the source of the RTC route.

Note: RT-constrain and Extended Community-based ORF are similar to the extent that they both allow a router to signal to a peer the Route Target Extended Communities they want to receive in VPN routes from that peer. But RT-constrain has distinct advantages over Extended Community-based ORF: it is more widely supported, it is simpler to configure, and its distribution scope is not limited to a direct peer.

In SR OS the capability to exchange RTC routes is advertised when the **route-target** keyword is added to the relevant **family** command. RT-constrain is supported on EBGP and IBGP sessions of the base router instance. On any particular session either ORF or RT-constrain may be used but not both; if RT-constrain is configured the ORF capability is not announced to the peer.

When RT-constrain has been negotiated with one or more peers SR OS automatically originates and advertises to these peers one /96 RTC route (the origin AS and Route Target Extended Community are fully specified) for every route target imported by a locally-configured VPRN or BGP-based L2 VPN; this includes MVPN-specific route targets.

SR OS also supports a group/neighbor level **default-route-target** command that causes routers to generate and send a 0:0:0/0 default RTC route to one or more peers. Sending the default RTC route to a peer conveys a request to receive all VPN routes from that peer. The **default-route-target** command is typically configured on sessions that a route reflector has with its PE clients. A received default RTC route is never propagated to other routers.

The advertisement of RTC routes by a route reflector follows special rules that are described in RFC 4684. These rules are needed to ensure that RTC routes for the same NLRI that are originated by different PE routers in the same Autonomous System are properly distributed within the AS.

When a BGP session comes up, and RT-constrain is enabled on the session (both peers advertised the MP-BGP capability), routers delay sending any VPN-IPv4 and VPN-IPv6 routes until either the session has been up for 60 seconds or the End-of-RIB marker is received for the RT-constrain address family. When the VPN-IPv4 and VPN-IPv6 routes are sent they are filtered to include only those with a Route Target Extended Community that matches an RTC route from the peer. VPN-IP routes matching an RTC route originated in the local AS are advertised to any IBGP peer that advertises a valid path for the RTC NLRI. In other words, route distribution is not constrained to only the IBGP peer advertising the best path. On the other hand, VPN-IP routes matching an RTC route originated outside the local AS are only advertised to the EBGP or IBGP peer that advertises the best path.

Note: SR OS does not support an equivalent of *BGP-Multipath* for RT-Constrain routes. There is no way to distribute VPN routes across more than one 'almost' equal set of inter-AS paths.

5.6.3.4 Min route advertisement interval

According to the BGP standard (RFC 4271), a BGP router should not send updated reachability information for an NLRI to a BGP peer until a specific period of time, Min Route Advertisement Interval (MRAI), has elapsed from the last update. The RFC suggests the MRAI should be configurable per peer but does not propose a specific algorithm, and therefore, MRAI implementation details vary from one router operating system to another.

In SR OS, the MRAI is configurable, on a per-session basis, using the **min-route-advertisement** command. The **min-route-advertisement** command can be configured with any value between 1 and 255 seconds and the setting applies to all address families. The default value is 30 seconds, regardless of the session type (EBGP or IBGP). The MRAI timer is started at the configured value when the session is established and counts down continuously, resetting to the configured value whenever it reaches zero. Every time it reaches zero, all pending RIB-OUT routes are sent to the peer.

To send UPDATE messages that advertise new NLRI reachability information more frequently for some address families than others, SR OS offers a **rapid-update** command that overrides the remaining time on a peer's MRAI timer and immediately sends routes belonging to specified address families (and all other pending updates) to the peers receiving these routes. The address families that can be configured with **rapid-update** support are:

- EVPN
- L2-VPN
- label-IPv4
- label-IPv6
- MCAST-VPN-IPv4
- MCAST-VPN-IPv6
- MDT-SAFI
- MVPN-IPv4
- MVPN-IPv6
- VPN-IPv4
- VPN-IPv6

In many cases, the default MRAI is appropriate for all address families (or at least those not included in the preceding list) when it applies to UPDATE messages that advertise reachable NLRI, but it is not the best option for UPDATE messages that advertise unreachable NLRI (route withdrawals). Fast re-convergence after some types of failures requires route withdrawals to propagate to other routers as quickly as possible so that they can calculate and start using new best paths, which would be impeded by the effect of the MRAI timer at each router hop. This is facilitated by the **rapid-withdrawal** configuration command.

When **rapid-withdrawal** is configured, UPDATE messages containing withdrawn NLRI are sent immediately to a peer without waiting for the MRAI timer to expire. UPDATE messages containing reachable NLRI continue to wait for the MRAI timer to expire, or for a **rapid-update** trigger, if it applies. When **rapid-withdrawal** is enabled, it applies to all address families.

When there is a change to a labeled-unicast route that requires reprogramming of the label operations in the data plane, these IOM updates are not made until the changed route is advertised to a peer, which depends on MRAI. Lowering the MRAI value or using **rapid-update** improves the speed of this operation.

5.6.3.5 Advertise-inactive

BGP does not allow a route to be advertised unless it is the best path in the RIB and an export policy allows the advertisement.

In some cases, it may be useful to advertise the best BGP path to peers despite the fact that is inactive. For example, because there are one or more preferred non-BGP routes to the same destination and one of these other routes is the active route. One way SR OS supports this flexibility is using the **advertise-inactive** command; other methods include **best-external** and **add-paths**.

When the BGP **advertise-inactive** command is configured so that it applies to a BGP session it has the following effect on the IPv4, IPv6, mcast-ipv4, mcast-ipv6, label-IPv4 and label-IPv6 routes advertised to that peer.

- If the active route for the IP prefix is a BGP route then that route is advertised. If the active route for the IP prefix is a non-BGP route and there is at least one valid but inactive BGP route for the same destination then the best of the inactive and valid BGP routes is advertised unless the non-BGP active route is matched and accepted by an export policy applied to the session.
- If the active route for the IP prefix is a non-BGP route and there are no (valid) BGP routes for the same destination then no route is advertised for the prefix unless the non-BGP active route is matched and accepted by an export policy applied to the session.

5.6.3.6 Best-external

Best-external is a BGP enhancement that allows a BGP speaker to advertise to its IBGP peers its best "external" route for a prefix/NLRI when its best overall route for the prefix/NLRI is an "internal" route. This is not possible in a normal BGP configuration because the base BGP specification prevents a BGP speaker from advertising a non-best route for a destination.

In specific topologies **best-external** can improve convergence times, reduce route oscillation and allow better loadsharing. This is achieved because routers internal to the AS have knowledge of more exit paths from the AS. Enabling **add-paths** on border routers of the AS can achieve a similar result but **add-paths** introduces NLRI format changes that must be supported by BGP peers of the border router and therefore has more interoperability constraints than **best-external** (which requires no messaging changes).

Best-external is supported in the base router BGP context. (A related feature is also supported in VPRNs; consult the Services Guide for more details.) It is configured using the **advertise-external** command, which provides IPv4, label-IPv4, IPv6, and label-IPv6 as options.

The advertisement rules when **advertise-external** is enabled can be summarized as follows.

- If a router has **advertise-external** enabled and its best overall route is a route from an IBGP peer then this best route is advertised to EBGP and confed-EBGP peers, and the "best external" route is advertised to IBGP peers. The "best external" route is the one found by running the BGP path selection algorithm on all LOC-RIB paths except for those learned from the IBGP peers.

  Note: A route reflector with **advertise-external** enabled does not include IBGP routes learned from other clusters in its definition of 'external'.

- If a router has **advertise-external** enabled and its best overall route is a route from an EBGP peer then this best route is advertised to EBGP, confed-EBGP, and IBGP peers.
- If a router has **advertise-external** enabled and its best overall route is a route from a confed-EBGP peer in member AS X then this best route is advertised to EBGP, IBGP peers and confed-EBGP peers in all member AS except X and the "best external" route is advertised to confed-EBGP peers in member AS X. In this case the "best external" route is the one found by running the BGP path selection algorithm on all RIB-IN paths except for those learned from member AS X.

  Note: If the best-external route is not the best overall route it is not installed in the forwarding table and in some cases this can lead to a short-duration traffic loop after failure of the overall best path.

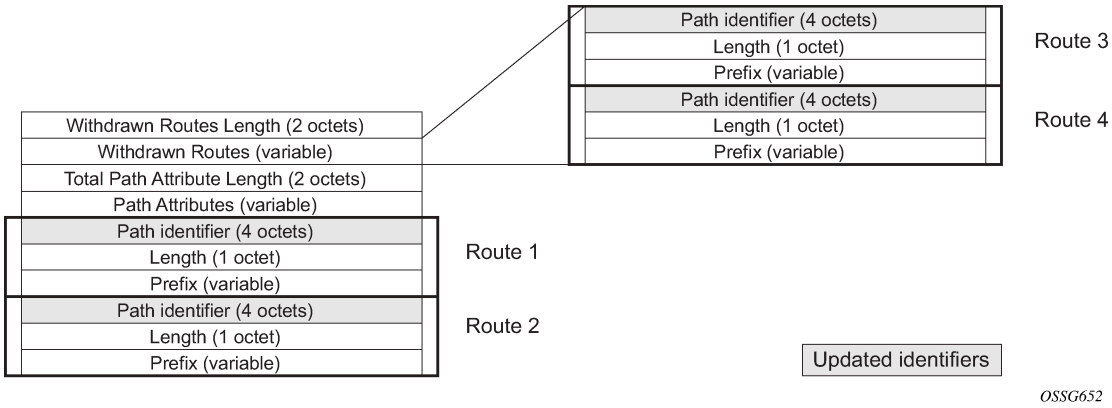
5.6.3.7 Add-paths

Add-paths is a BGP enhancement that allows a BGP router to advertise multiple distinct paths for the same prefix/NLRI. Add-Paths provides a number of potential benefits, including reduced routing churn, faster convergence, and better loadsharing.

For a router to receive multiple paths per NLRI from a peer, for a particular address family, the peer must announce the BGP capability to send multiple paths for the address family and the local router must announce the BGP capability to receive multiple paths for the address family. When the Add-Path capability has been negotiated this way, all advertisements and withdrawals of NLRI by the peer must include a path identifier. The path identifier has no significance to the receiving router. If the combination of NLRI and path identifier in an advertisement from a peer is unique (does not match an existing route in the RIB-IN from that peer) then the route is added to the RIB-IN. If the combination of NLRI and path identifier in a received advertisement is the same as an existing route in the RIB-IN from the peer then the new route replaces the existing one. If the combination of NLRI and path identifier in a received withdrawal matches an existing route in the RIB-IN from the peer, then that route is removed from the RIB-IN.

An UPDATE message carrying an IPv4 NLRI with a path identifier is shown in Figure 30: BGP update message with path identifier for IPv4 NLRI.

Figure 30: BGP update message with path identifier for IPv4 NLRI



Add-paths is only supported by the base router BGP instance and the EBGP and IBGP sessions it forms with other peers capable of add-paths. The ability to send and receive multiple paths per prefix is configurable per family, and supported for the following options:

- IPv4
- label-IPv4
- VPN-IPv4
- IPv6
- label-IPv6
- VPN-IPv6
- MCAST-VPN-IPv4
- MCAST-VPN-IPv6
- MVPN-IPv4
- MVPN-IPv6
- EVPN

5.6.3.7.1 Path selection with add-paths

The local RIB may have multiple paths for a prefix. The path selection mode refers to the algorithm used to decide which paths to advertise to an add-paths peer. SR OS supports a send *N* path selection algorithm (See *draft-ietf-idr-add-paths-guidelines*) and a send multipaths selection algorithm.

The send N algorithm selects the N best advertisable paths that meet the following constraints.

- The BGP next-hop of the route is unique.
- The BGP route is not rejected by an export policy.
- The BGP route is not blocked by a split-horizon rule.
- The number of advertised paths does not exceed *N*. *N* is derived from the send-limit of the best BGP RIB-IN as applied by BGP import policy action or the configuration of the **add-paths** command that applies to the neighbor and the address family.

The send multipaths algorithm selects the *N* best advertisable paths that meet the following constraints.

- The BGP route is a multipath (in other words, it is tied with the best path up to and including the NH-cost comparison step of the decision process, skipping steps that do not apply).
- The BGP next-hop of the route is unique.
- The BGP route is not rejected by an export policy.
- The BGP route is not blocked by a split-horizon rule.

5.6.3.8 Split-horizon

Split-horizon refers to the action taken by a router to avoid advertising a route back to the peer from which it was received. By default, SR OS applies split-horizon behavior only to routes received from IBGP non-client peers, and split-horizon only works for routes to non-imported routes within a RIB. This split-horizon functionality, which can never be disabled, prevents a route learned from a non-client IBGP peer to be advertised to the sending peer or any other non-client peer.

To apply split-horizon behavior to routes learned from RR clients, confed-EBGP peers or (non-confed) EBGP peers the **split-horizon** command must be configured in the appropriate contexts; it is supported at the global BGP, **group** and **neighbor** levels. When **split-horizon** is enabled on these types of sessions, it only prevents the advertisement of a route back to its originating peer; for example, SR OS does not prevent the advertisement of a route learned from one EBGP peer back to a different EBGP peer in the same neighbor AS.

5.7 BGP monitoring protocol

The BGP Monitoring Protocol (BMP) provides a monitoring station that obtains route updates and statistics from a BGP router. The BMP protocol is described in detail in RFC 7854, *BGP Monitoring Protocol (BMP)*. A router communicates information about one or more BGP sessions to a BMP station. Specifically, BMP allows a BGP router to advertise the pre-policy or post-policy BGP RIB-In from specific BGP peers to a monitoring station. This allows the monitoring station to monitor the routing table size, identify issues, and monitor trends in the table size and update or withdraw the frequency. The BMP station is also sometimes called a BMP collector. A router sends information in BMP messages to a BMP station.

BMP is a unidirectional protocol. A BMP station never sends back any messages to a router.

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 the established mode (for example, when the peer goes up or down). These notifications are called BMP peer-up and peer-down messages.
- A router can periodically send statistical information about one or more neighbors. This information consists of a number of counters, for example, the number of routes received from a particular neighbor or the number of rejected or accepted routes because of ingress policy parameters.

  Other counters report the number of errors that were encountered, for example, AS-path loops, duplicate prefixes, or withdrawals received.
- A router can also report the exact routes received from a particular neighbor. This action is called route monitoring. A router encapsulates a BGP route into the original BGP update message, then encapsulates that BGP update message in a BMP route monitoring message.

BMP on an SR OS router reports information about routes that were received from a neighbor. The SR OS cannot report routes that were sent to a neighbor.

When periodic statistics are enabled, the router sends all the statistics as described in RFC 7854, section 4.8, with the exception of statistic number 13, "Number of duplicate update messages received". The supported statistics are listed in Table 8: Supported statistics.

Table 8: Supported statistics

Statistic	Type
Number of Prefixes rejected by inbound policy	0
Number of duplicate prefix advertisements received	1
Number of duplicate withdraws received	2
Number of invalidated prefixes because of Cluster_List loop detection	3
Number of invalidated prefixes because of AS_PATH loop detection	4
Number of invalidated prefixes because of Originator ID validation	5
Number of invalidated prefixes because AS-Confed loop detection	6
Total number of routes in adj-rib-in (all families)	7
Total number of routes in Local-RIB (all families)`	8
Number of routes per address-family in adj-rib-in	9
Number of routes per address-family in loc-rib	10
Number of updates subjected to treat-as-withdraw	11
Number of prefixes subjected to treat-as-withdraw	12

Note: Statistics 9 and 10 are per address family. The address family is specified as an AFI/SAFI pair. Regardless of which families are configured for route monitoring, a router reports the statistics of all address families that were negotiated with the neighbor.
The values in these counters are the same values that can be seen in the output of the following commands.

```
show router bgp neighbor
show router bgp neighbor detail
```

5.8 BGP applications

SR OS implements the following BGP application:

BGP FlowSpec

5.8.1 BGP FlowSpec

FlowSpec is a standardized method for using BGP to distribute traffic flow specifications (flow routes) throughout a network. A flow route carries a description of a flow in terms of packet header fields such as source IP address, destination IP address, or TCP/UDP port number and indicates (through a community attribute) an action to take on packets matching the flow. The primary application for FlowSpec is DDoS mitigation.

FlowSpec is supported for both IPv4 and IPv6. To exchange non-VPN-aware IPv4 FlowSpec routes with a BGP peer, **flow-ipv4** must be enabled in the **family** configuration that applies to the session. To exchange non-VPN-aware IPv6 FlowSpec routes with a BGP peer, **flow-ipv6** must be enabled in the relevant **family** configuration. The IP filter entries created from non-VPN-aware FlowSpec routes can only be used on the IP interfaces of the base or VPRN router instance that received the FlowSpec routes.

SR OS BGP also supports VPN-aware FlowSpec routes. These SAFI 134 routes carry Route Distinguisher (RD) and RT Extended Communities. To exchange VPN-aware IPv4 FlowSpec routes with an IBGP peer of the base router, **flow-vpn-ipv4** must be enabled in the **family** configuration that applies to the session. To exchange VPN-aware IPv6 FlowSpec routes with an IBGP peer of the base router, **flow-vpn-ipv6** must be enabled in the relevant **family** configuration. IP filter entries are created from a VPN-aware FlowSpec route when the route is imported into a locally configured VPRN. This is done by appropriately configuring a VRF import or VRF target policy for the VPRN. There must also be an IP filter policy applied to the IP interfaces of the VPRN that embeds FlowSpec routes.

The NLRI of a FlowSpec IPv4 or FlowSpec-VPN IPv4 route can contain one or more of the subcomponents shown in the following table.

Table 9: Subcomponents of FlowSpec IPv4 and FlowSpec-VPN IPv4 NLRI

Subcomponent name [type]	Value encoding	SR OS support
Destination IPv4 prefix [1]	Prefix length, prefix	Yes
Source IPv4 prefix [2]	Prefix length, prefix	Yes
IP protocol [3]	One or more (operator, value) pairs	Partial. No support for multiple values other than "TCP or UDP".
Port [4][2]	One or more (operator, value) pairs	Yes. Support single port, port-range.
Destination port [5]	One or more (operator, value) pairs	Yes. Support single port, port-range.
Source port [6]	One or more (operator, value) pairs	Yes. Support single port, port-range.
ICMP type [7]	One or more (operator, value) pairs	Partial. Only a single value is supported.
ICMP code [8]	One or more (operator, value) pairs	Partial. Only a single value is supported.
TCP flags [9]	The following restrictions apply: • FP4- and FP5-based platforms support multiple (operator, bitmask)	Yes

[2] The Port [4] subcomponent specifies both source and destination ports.

Subcomponent name [type]	Value encoding	SR OS support
	pairs, provided a single TCP flag bit is matched in each bitmask pair and the match bit is set to 0, resulting in an AND operation between the TCP flags. • Multiple TCP flags can be set in the same (operator, bitmask) pair, provided there is a single pair in the NLRI component with match bit is set to 1 and not bit set to 0. • FP3-based platforms support SYN and ACK only.	
Packet length [10]	One or more (operator, value) pairs	Yes
DSCP [11]	One or more (operator, value) pairs	Partial. Only a single value is supported.
Fragment [12]	One or more (operator, bitmask) pairs	Partial. No support for matching Last Fragment.

The NLRI of a FlowSpec IPv6 or FlowSpec-VPN IPv6 route can contain one or more of the subcomponents shown in the following table.

Table 10: Subcomponents of FlowSpec IPv6 and FlowSpec-VPN IPv6 NLRI

Subcomponent name [type]	Value encoding	SR OS support
Destination IPv6 prefix [1]	Prefix length, prefix offset, prefix	Partial. No support for prefix offset.
Source IPv6 prefix [2]	Prefix length, prefix offset, prefix	Partial. No support for prefix offset.
Next header [3]	One or more (operator, value) pairs	Partial. Only a single value supported.
Port [4][2]	One or more (operator, value) pairs	Yes. Support single port, port-range.
Destination port [5]	One or more (operator, value) pairs	Yes. Support single port, port-range.
Source port [6]	One or more (operator, value) pairs	Yes. Support single port, port-range.
ICMP type [7]	One or more (operator, value) pairs	Partial. Only a single value is supported.
ICMP code [8]	One or more (operator, value) pairs	Partial. Only a single value is supported.
TCP flags [9]	The following restrictions apply: • FP4- and FP5-based platforms support multiple (operator, bitmask) pairs, provided a single TCP flag	Yes

Subcomponent name [type]	Value encoding	SR OS support
	bit is matched in each bitmask pair and the match bit is set to 0, resulting in an AND operation between the TCP flags. • Multiple TCP flags can be set in the same (operator, bitmask) pair, provided there is a single pair in the NLRI component with match bit is set to 1 and not bit set to 0. • FP3-based platforms support SYN and ACK only.	
Packet length [10]	One or more (operator, value) pairs	Yes
Traffic class [11]	One or more (operator, value) pairs	Partial. Only a single value is supported.
Fragment [11]	One or more (operator, bitmask) pairs	Partial. No support for matching Last Fragment.
Flow label [13]	One or more (operator, value) pairs	Partial. Only a single value is supported.

Table 11: IPv4 FlowSpec actions summarizes the actions that may be associated with FlowSpec IPv4 and FlowSpec-VPN IPv4 routes. Table 12: IPv6 FlowSpec actions summarizes the actions that may be associated with FlowSpec IPv6 and FlowSpec-VPN IPv6 routes.

Table 11: IPv4 FlowSpec actions

Action	Encoding	SR OS support
Rate limit	Extended community type 0x8006	Yes
Sample/log	Extended community type 0x8007 S-bit	Yes
Next entry	Extended community type 0x8007 T-bit	—
Redirect to VRF	Extended community type 0x8008	Yes
Mark traffic class	Extended community type 0x8009	Yes
Redirect to IPv4	Extended community type 0x010c	Yes
Redirect to IPv6	Extended community type 0x000c	—
Redirect to LSP	Extended community type 0x0900	Partial, only support for ID-type 0x00 (localized ID)

Table 12: IPv6 FlowSpec actions

Action	Encoding	SR OS support
rate limit	Extended community type 0x8006	Yes
sample/log	Extended community type 0x8007 S-bit	Yes
next entry	Extended community type 0x8007 T-bit	—
Redirect to VRF	Extended community type 0x8008	Yes
Mark traffic class	Extended community type 0x8009	Yes
Redirect to IPv4	Extended community type 0x010c	—
Redirect to IPv6	Extended community type 0x000c	Yes
Redirect to LSP	Extended community type 0x0900	Partial, only support for ID-type 0x00 (localized ID)

5.8.1.1 Validating received FlowSpec routes

Received non-VPN-aware FlowSpec routes are validated following the procedures described in RFC 5575 and *draft-ietf-idr-bgp-flowspec-oid-03*, *Revised Validation Procedure for BGP Flow Specifications*. Configure the **validate-dest-prefix** command in a routing instance to enable validation checks based on the destination prefix. By default, this check is not performed. When the command is enabled, BGP determines whether a non-VPN-aware FlowSpec route is valid or invalid based on the following logic:

1. If the FlowSpec route was originated in the same Autonomous System (AS) as the receiving BGP router, it is automatically valid.
2. If rule 1 does not apply, the FlowSpec route was originated in an external AS, and it does not contain a destination prefix subcomponent, it is considered valid.
3. If rule 1 does not apply, the FlowSpec route was originated in an external AS, and it does contain a destination prefix subcomponent, it is considered valid if all of the following are true:
   - The neighbor AS (last non-confed AS in the AS_PATH) of the FlowSpec route matches the neighbor AS of the unicast IP route that is the best match of the destination prefix. The best match unicast IP route must be a BGP route (that is, not static, IGP, or other routes).
   - The neighbor AS of the FlowSpec route matches the neighbor AS of all unicast IP routes that are longer matches of the destination prefix. All longer match unicast IP routes must be BGP routes (that is, not static, IGP, or other routes).

Non-VPN-aware FlowSpec routes that are received with a redirect-to-IPv4 extended community action are also subject to an additional set of validation checks. If the **validate-redirect-ip** command is enabled in the receiving BGP instance, a non-VPN-aware FlowSpec route is considered invalid, if it is deemed to have originated in a different AS than the IP route that resolves the redirection IPv4 address. The originating AS of a FlowSpec route is determined from its AS paths.

A FlowSpec route that is determined to be invalid by any of the validation rules described earlier is retained in the BGP RIB, but not used for traffic filtering and not propagated to other BGP speakers.

5.8.1.2 Using flow routes to create dynamic filter entries

When the base router BGP instance receives a non-VPN-aware flow IPv4 or IPv6 route that is considered valid and best, the system attempts to construct an IPv4 or IPv6 filter entry from the NLRI contents and actions encoded in the UPDATE message. If successful, the filter entry is added to the system-created "fSpec-0" IPv4 embedded filter or to the "fSpec-0" IPv6 embedded filter. These embedded filters can be inserted into configured IPv4 and IPv6 filter policies that are applied to ingress traffic on a selected set of the base router IP interfaces. These interfaces can include network interfaces, IES SAP interfaces, and IES spoke SDP interfaces.

When the VPRN BGP instance receives a non-VPN-aware flow IPv4 or IPv6 route from a BGP peer of the VPRN or imports a VPN-aware FlowSpec-VPN IPv4 or IPv6 route that was received in the base router BGP instance and considered to be valid and best, the system attempts to construct an IPv4 or IPv6 filter entry from the NLRI contents and actions encoded in the UPDATE message. If successful, the filter entry is added to the system-created "fSpec-*$vprnid*" IPv4 embedded filter or to the "fSpec-*$vprnid*" IPv6 embedded filter. *$vprnid* represents a parameter value that is unique to each VPRN. These embedded filters can be inserted into configured IPv4 and IPv6 filter policies that are applied to ingress traffic on one or more of the IP interfaces on the VPRN.

When FlowSpec rules are embedded into a user-defined filter policy, configure the insertion point of the rules using the following commands:

- **MD-CLI**

```
configure filter ip-filter embed flowspec offset
configure filter ipv6-filter embed flowspec offset
```

- **classic CLI**

```
configure filter ip-filter embed-filter flowspec offset
configure filter ipv6-filter embed-filter flowspec offset
```

The sum of the configured maximum number of FlowSpec routes and offset must not exceed the maximum filter entry ID range.

5.8.1.2.1 BGP FlowSpec steering toward the SRv6 policy

BGP FlowSpec is enhanced to facilitate IPv6 traffic steering toward a remote IPv6 address in an SRv6 policy. This enhancement adds supplementary information, such as the redirect IPv6 next-hop, color, and SRv6 service SID. The system uses this information to guide specific traffic that matches the FlowSpec filter Network Layer Reachability Information (NLRI) toward the designated SRv6 policy.

FlowSpec refers to the use of BGP to distribute traffic flow specifications (flow routes) throughout a network. A flow route carries a description of a flow in the packet header fields such as source IP address, destination IP address, and TCP/UDP port number, and indicates (using BGP communities) an action to take on packets matching the flow.

The *draft-ietf-idr-ts-flowspec-srv6-policy-03* document defines the use of BGP FlowSpec to direct traffic toward an SRv6 policy. This process involves leveraging existing FlowSpec and localized Policy-Based Routing (PBR) capabilities, with no novel BGP attributes.

The *ietf-idr-FlowSpec-redirect-ip-02* document defines the framework for redirecting traffic to both IPv4 and IPv6 next-hop addresses. The FlowSpec steering to the SRv6 policy uses the IPv6 extended community as a destination point for the SRv6 policy.

After reaching the SRv6 policy tail-end device, specific identifiers determine the processing steps for the flows. The BGP prefix-SID, as defined in RFC 8669, plays a role in enabling SRv6 VPN services (defined in RFC 9252).The SRv6 service TLVs of the BGP prefix-SID attribute indicate endpoint functions that serve as service SIDs to guide the processing of incoming packets at the policy endpoint. The BGP-encoded SRv6 TLV must be an SRv6 L3 Service TLV to extract a valid SRv6 Service SID.

The *draft-ietf-idr-ts-flowspec-srv6-policy-03* document introduces the color extended community within a FlowSpec NLRI context, as defined in RFC 8955 and RFC 8956. This feature serves to select the appropriate SRv6 SR policy.

Conditions

The following FlowSpec validation conditions apply for steering to an SRv6 policy:

- When the FlowSpec IPv6 extended community exists but either the color extended community or BGP prefix-SID is missing, the FlowSpec steering falls back to the redirect-IP action.
- If the FlowSpec NLRI is missing the IPv6 extended community but has the color extended community, or the BGP prefix-SID, then no steering to redirect-IP or SR-policy happens and no other FlowSpec action is considered.

Restrictions

The following restrictions apply:

- The feature only supports combining one color extended community with one BGP prefix-SID attribute.
- The SRv6 Service TLV in the SRv6 policy must be the L3 Service TLV.
- The feature does not support combining redirect-to-VRF and redirect-to-IP extended communities for a single FlowSpec NLRI.
- The feature does not support a received redirect-to-IP extended community with C=1 set in the local administrator field of the redirect-to-IP extended community.
- An unsupported FlowSpec entry is handled as an exception and the system generates a log for unsupported actions and creates no filter entry. However, the entry is propagated as if the BGP route was used.

5.8.2 Configuration of TTL propagation for BGP labeled routes

This feature allows the separate configuration of TTL propagation for in transit and CPM generated IP packets at the ingress LER within a BGP labeled route context.

5.8.2.1 TTL propagation for RFC 8277 labeled route at ingress LER

For IPv4 and IPv6 packets forwarded using a RFC 8277 labeled route in the global routing instance, including label-IPv6, the following command specified with the **all** value enables TTL propagation from the IP header into all labels in the transport label stack:

- **config router ttl-propagate label-route-local [none | all]**
- **config router ttl-propagate label-route-transit [none | all]**

The **none** value reverts to the default mode which disables TTL propagation from the IP header to the labels in the transport label stack.

These commands do not have a **no** version.

Note:

- The TTL of the IP packet is always propagated into the RFC 8277 label itself. The commands only control the propagation into the transport labels, for example, the labels of the RSVP or LDP LSP which the BGP labeled route resolves to and which are pushed on top of the BGP label.
- If the BGP peer advertised the **implicit-null** label value for the BGP labeled route, the TTL propagation does not follow the configuration described, but follows the configuration to which the BGP labeled route resolves:

- RSVP LSP shortcut:
  - **configure router mpls shortcut-transit-ttl-propagate**
  - **configure router mpls shortcut-local-ttl-propagate**
- LDP LSP shortcut:
  - **configure router ldp shortcut-transit-ttl-propagate**
  - **configure router ldp shortcut-local-ttl-propagate**

This feature does not impact packets forwarded over BGP shortcuts. The ingress LER operates in uniform mode by default and can be changed into pipe mode using the configuration of TTL propagation for RSVP or LDP LSP shortcut.

5.8.2.2 TTL propagation for RFC 8277 labeled routes at LSR

This feature configures the TTL propagation for transit packets at a router acting as an LSR for a BGP labeled route.

When an LSR swaps the BGP label for a IPv4 prefix packet, therefore acting as a ABR, ASBR, or data-path Route-Reflector (RR) in the base routing instance, or swaps the BGP label for a vpn-IPv4 or vpn-IPv6 prefix packet, therefore acting as an inter-AS Option B VPRN ASBR or VPRN dataHpath Route-Reflector (RR), the all value of the following command enables TTL propagation of the decremented TTL of the swapped BGP label into all LDP or RSVP transport labels.

configure router ttl-propagate lsr-label-route [none | all]

When an LSR swaps a label or stitches a label, it always writes the decremented TTL value into the outgoing swapped or stitched label. What the above CLI controls is whether this decremented TTL value is also propagated to the transport label stack pushed on top of the swapped or stitched label.

The **none** value reverts to the default mode which disables TTL propagation. This changes the existing default behavior which propagates the TTL to the transport label stack. When a customer upgrades, the new default becomes in effect. The above commands do not have a no version.

The following describes the behavior of LSR TTL propagation in a number of other use cases and indicates if the above CLI command applies or not.

- When an LSR stitches an LDP label to a BGP label, the decremented TTL of the stitched label is propagated or not to the LDP or RSVP transport labels as per the above configuration.

- When an LSR stitches a BGP label to an LDP label, the decremented TTL of the stitched label is automatically propagated into the RSVP label if the outgoing LDP LSP is tunneled over RSVP. This behavior is not controlled by the above CLI.
- When an LSR pops a BGP label and forwards the packet using an IGP route (IGP route to destination of prefix wins over the BGP labeled route), it pushes an LDP label on the packet and the TTL behavior is as described in the previous bullet when stitching from a BGP label to an LDP label.
- When an ingress Carrier Supporting Carrier (CsC) PE swaps the incoming EBGP label into a VPN-IPv4 label. The reverse operation is performed by the egress CsC PE. In both cases, the decremented TTL of the swapped label is or is not passed on to the LDP or RSVP transport labels as per the above configuration.

5.8.3 BGP prefix origin validation

BGP prefix origin validation is a solution developed by the IETF SIDR working group for reducing the vulnerability of BGP networks to prefix mis-announcements and specific man-in-the-middle attacks. BGP has traditionally relied on a trust model where it is assumed that when an AS originates a route it has the right to announce the associated prefix. BGP prefix origin validation takes extra steps to ensure that the origin AS of a route is valid for the advertised prefix.

7450, 7750, and 7950 routers support BGP prefix origin validation for IPv4 and IPv6 routes received from selected peers. When prefix origin validation is enabled on a base router BGP or VPRN BGP session using the **enable-origin-validation** command, every received IPv4, IPv6, or both route received from the peer is checked to determine whether the origin AS is valid for the received prefix. The origin AS is the first AS that was added to the AS_PATH attribute and indicates the autonomous system that originated the route.

For purposes of determining the origin validation state of received BGP routes, the router maintains an Origin Validation database consisting of static and dynamic entries. Each entry is called a VRP (Validated ROA Payload) and associates a prefix (range) with an origin AS.

Static VRP entries are configured using the **static-entry** command in the **configure router origin-validation** context of the base router. In SR OS, a static entry can express that a specific prefix and origin AS combination is either valid or invalid.

Dynamic VRP entries are learned from RPKI local cache servers and express valid origin AS and prefix combinations. The router communicates with RPKI local cache servers using the RPKI-RTR protocol. SR OS supports the RPKI-RTR protocol over TCP/IPv4 or TCP/IPv6 transport; TCP-MD5 and other forms of session security are not supported. 7450, 7750, and 7950 routers can set up an RPKI-RTR session using the base routing table (in-band) or the management router (out-of-band). Dynamic VRP entries are configured and displayed using the RPKI commands in the **configure router origin-validation** and **show router origin-validation** contexts. For command information, see the following:

- **MD-CLI**

  7450 ESS, 7750 SR, 7950 XRS, and VSR MD-CLI Command Reference Guide

  7450 ESS, 7750 SR, 7950 XRS, and VSR Clear, Monitor, Show, and Tools CLI Command Reference Guide

- **classic CLI**

  7450 ESS, 7750 SR, 7950 XRS, and VSR Classic CLI Command Reference Guide

  7450 ESS, 7750 SR, 7950 XRS, and VSR Clear, Monitor, Show, and Tools CLI Command Reference Guide

An RPKI local cache server is one element of the larger RPKI system. The RPKI is a distributed database containing cryptographic objects relating to Internet Number resources. Local cache servers are deployed in the service provider network and retrieve digitally signed Route Origin Authorization (ROA) objects from Global RPKI servers. The local cache servers cryptographically validate the ROAs before passing the information along to the routers.

The algorithm used to determine the origin validation states of routes received over a session with **enable-origin-validation** configured uses the following definitions:

- A route is matched by a VRP entry if all of the following occurs:
  - the prefix bits in the route match the prefix bits in the VRP entry (up to its min prefix length
  - the route prefix length is greater than or equal to the VRP entry min prefix length
  - the route prefix length is less than or equal to the VRP entry max prefix length
  - the origin AS of the route matches the origin AS of the VRP entry
- A route is covered by a VRP entry if all of the follow occurs:
  - the prefix bits in the route match the prefix bits in the VRP entry (up to its min prefix length)
  - the route prefix length is greater than or equal to the VRP entry min prefix length
  - the VRP entry type is static-valid or dynamic

Using the above definitions, the origin validation state of a route is based on the following rules.

1. If a route is matched by at least one VRP entry, and the most specific of these matching entries includes a static-invalid entry then the origin validation state is Invalid (2).
2. If a route is matched by at least one VRP entry, and the most specific of these matching entries does not include a static-invalid entry then the origin validation state is Valid (0).
3. If a route is not matched by any VRP entry, but it is covered by at least one VRP entry then the origin validation state is Invalid (2).
4. If a route is not covered by any VRP entry then the origin validation state is Not-Found (1).

Consider the following example. Suppose the Origin Validation database has the following entries:

```
10.1.0.0/16-32, origin AS=5, dynamic
10.1.1.0/24-32, origin AS=4, dynamic
10.0.0.0/8-32, origin AS=5, static invalid
10.1.1.0/24-32, origin AS=4, static invalid
```

In this case, the origin validation state of the following routes are as indicated:

10.1.0.0/16 with AS_PATH {…5}: Valid

10.1.1.0/24 with AS_PATH {…4}: Invalid

10.2.0.0/16 with AS_PATH {…5}: Invalid

10.2.0.0/16 with AS_PATH {…6}: Not-Found

The origin validation state of a route can affect its ranking in the BGP decision process. When **origin-invalid-unusable** is configured, all routes that have an origin validation state of ‘Invalid’ are considered unusable by the best path selection algorithm, that is, they cannot be used for forwarding and cannot be advertised to peers.

If **origin-invalid-unusable** is not configured then routes with an origin validation state of ‘Invalid’ are compared to other "usable” routes for the same prefix according to the BGP decision process.

When **compare-origin-validation-state** is configured a new step is added to the BGP decision process after removal of invalid routes and before the comparison of local preference. The new step compares the origin validation state, so that a route with a 'Valid' state is preferred over a route with a 'Not-Found' state, and a route with a 'Not-Found' state is preferred over a route with an 'Invalid' state assuming that these routes are considered 'usable'. The new step is skipped if the **compare-origin-validation-state** command is not configured.

Route policies can be used to attach an Origin Validation State extended community to a route received from an EBGP peer to convey its origin validation state to IBGP peers and save them the effort of repeating the Origin Validation database lookup. To add an Origin Validation State extended community encoding the 'Valid' result, the route policy should add a community list that contains a member in the format **ext:4300:0**. To add an Origin Validation State extended community encoding the 'Not-Found' result, the route policy should add a community list that contains a member in the format **ext:4300:1**. To add an Origin Validation State extended community encoding the 'Invalid' result, the route policy should add a community list that contains a member in the format **ext:4300:2**.

5.8.4 BGP route leaking

It is possible to leak a copy of a BGP route (including all its path attributes) from one routing instance RIB to another routing instance RIB of the same type (labeled or unlabeled) in the same router. Leaking is supported from the GRT to a VPRN, from one VPRN to another VPRN, and from a VPRN to the GRT. Any valid BGP route for an IPv4, IPv6, label-IPv4, or label-IPv6 prefix can be leaked. A BGP route does not have to be the best path or used for forwarding in the source instance to be leaked.

An IPv4, IPv6, label-IPv4, or label-IPv6 BGP route becomes a candidate for leaking to another instance when it is specially marked by a BGP import policy. The operator can achieve this special marking 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 received in a specific source instance into the BGP RIB of a specific target instance, the operator must configure the target instance with a **leak-import** policy that matches and accepts the leakable route. The operator can specify different leak-import policies for each of the following RIBs: IPv4, label-IPv4, IPv6, and label-IPv6. Up to 15 **leak-import** policies can be chained together for more complex use cases. The **leak-import** policies are configured in various **config**>**router**>**bgp**>**rib-management** contexts.

Note: Using a **leak-import** policy to change the BGP attributes of leaked routes (compared to the original source copy) is not supported. The only attribute that can be changed is the RTM preference.

In the target instance, leaked BGP routes are compared to other (leaked and non-leaked) BGP routes for the same prefix based on the complete BGP decision process. Leaked routes do not have information about the router ID and peer IP address of the original peer and use all-zero values for these properties.

BGP always tries to resolve the BGP next hop of a leaked route using the route and tunnel table of the original (source) routing instance and this resolution information is carried with the leaked route, avoiding the need to leak the resolving routes as well. If BGP cannot resolve the route or tunnel in the source instance, the unresolved route cannot be leaked unless **allow-unresolved-leaking** is configured and the source routing instance is the GRT. In this case, the importing VPRN tries to resolve the BGP next hop of the leaked route by using its own route table (and according to its own BGP **next-hop-resolution** configuration options).

If a target instance has BGP multipath and ECMP enabled and some of the equal-cost best paths for a prefix are leaked routes, they can be used along with non-leaked best paths as ECMP next hops of the route.

When BGP fast reroute is enabled in a target instance (for a particular IP prefix), BGP attempts to find a qualifying backup path by considering both leaked and non-leaked BGP routes. The backup path criteria are unchanged by this feature, that is, the backup path is the best remaining path after the primary paths and all paths with the same BGP next hops as the primary paths have been removed.

A leaked BGP route can be advertised to direct BGP neighbors of the target routing instance.

Note: VPRN BGP instances do not support label-IPv6 route advertisements.

The BGP next hop of a leaked route is automatically reset to itself whenever it is advertised to a peer of the target instance. Normal route advertisement rules apply, meaning that by default, the leaked route is advertised only if (in the target instance) it is the overall best path and is used as the active route to the destination and is not blocked by the IBGP-to-IBGP split-horizon rule.

A BGP route resolved in the source routing instance and leaked into a VPRN can be exported from the VPRN as a VPN-IPv4 or VPN-IPv6 route if it matches the VRF export policy. In this case, normal VPN export rules apply, meaning that by default, the leaked route is exported only if (in the VPRN) it is the overall best path and is used as the active route to the destination.

A BGP route that is unresolved in the GRT, leaked into a VPRN, and resolved by a BGP-VPN route in the VPRN cannot be exported from the VPRN as a VPN-IPv4 or VPN-IPv6 route unless it matches the VRF export policy and the VPRN is configured with the **allow-bgp-vpn-export** command.

Note: A leaked route cannot be exported as a VPN-IP route and then reimported into another local VPRN.

5.8.5 BGP optimal route reflection

BGP Route Reflectors (RRs) are used in networks to improve network scalability by eliminating or reducing the need for a full-mesh of IBGP sessions. When a BGP RR receives multiple paths for the same IP prefix, it typically selects a single best path to send to all clients. If the RR has multiple nearly-equal best paths and the tie-break is determined by the next-hop cost, the RR advertises the path based on its view of next-hop costs. The advertised route may differ from the path that a client would select if it had visibility of the same set of candidate paths and used its own view of next-hop costs.

Non-optimal advertisements by the RR can be a problem in hot-potato routing designs. Hot-potato routing aims to hand off traffic to the next AS using the closest possible exit point from the local AS. In this context, the closest exit point implies minimum IGP cost to reach the BGP next-hop. SR OS implements the hot-potato routing solution described in *draft-ietf-idr-bgp-optimal-route-reflection*.

Optimal Route Reflection (ORR) is supported in the base router BGP instance only. It applies to routes in the following address families: IPv4 unicast, label-IPv4, label-IPv6 (6PE), VPN-IPv4, and VPN-IPv6.

Note: For the RR to compare two VPN routes (and therefore for ORR to apply), the routes must contain the same RD and IP prefix information.

ORR locations are created when **config**>**router**>**bgp**>**orr**>**location** is configured. The RR can maintain information for a maximum of 255 ORR locations. A primary IPv4 or primary IPv6 address is required for each location; optionally, specify a secondary and tertiary IPv4 and IPv6 addresses for the location. The IP addresses are used to find a node in the network topology that can serve as the root for SPF calculations.

The IP addresses must correspond to loopback or system IP addresses of routers that participate in IGP protocols. The secondary and tertiary IP address parameters provide redundancy in case the node selected to be root for the SPF calculations disappears.

The route reflector's TE database, populated with information from local IGP instances or BGP-LS NLRI, is used to compute the SPF cost from each ORR location to IPv4 and IPv6 BGP next-hops in the candidate set of best paths. The use of BGP-LS allows the route reflector to learn IGP topology information for OSPF areas, IS-IS levels, and others in which the route reflector is not a direct participant.

To configure an ORR client, configure the **cluster** command for the BGP session to reference one of the defined ORR locations. The association of a client with an ORR location is not automatic. Choose an ORR location as close as possible to the client that is being configured. The **allow-local-fallback** option of the **cluster** command affects RR behavior when no BGP routes are reachable from the ORR location of the client. When **allow-local-fallback** is configured, the RR is allowed, in this circumstance only, to advertise the best reachable BGP path from its own topology location. If **allow-local-fallback** is not configured and this situation applies, then no route is advertised to the client.

Note: ORR is supported with add-paths; add-paths advertised to an ORR client are based on ORR location.

5.8.6 LSP tagging for BGP next-hops or prefixes and BGP-LU

The tunnels used by the system for resolution of BGP next-hops or prefixes and BGP labeled unicast routes can be constrained using LSP administrative tags. For more information, see the 7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide, 'LSP Tagging and Auto-Bind Using Tag Information'.

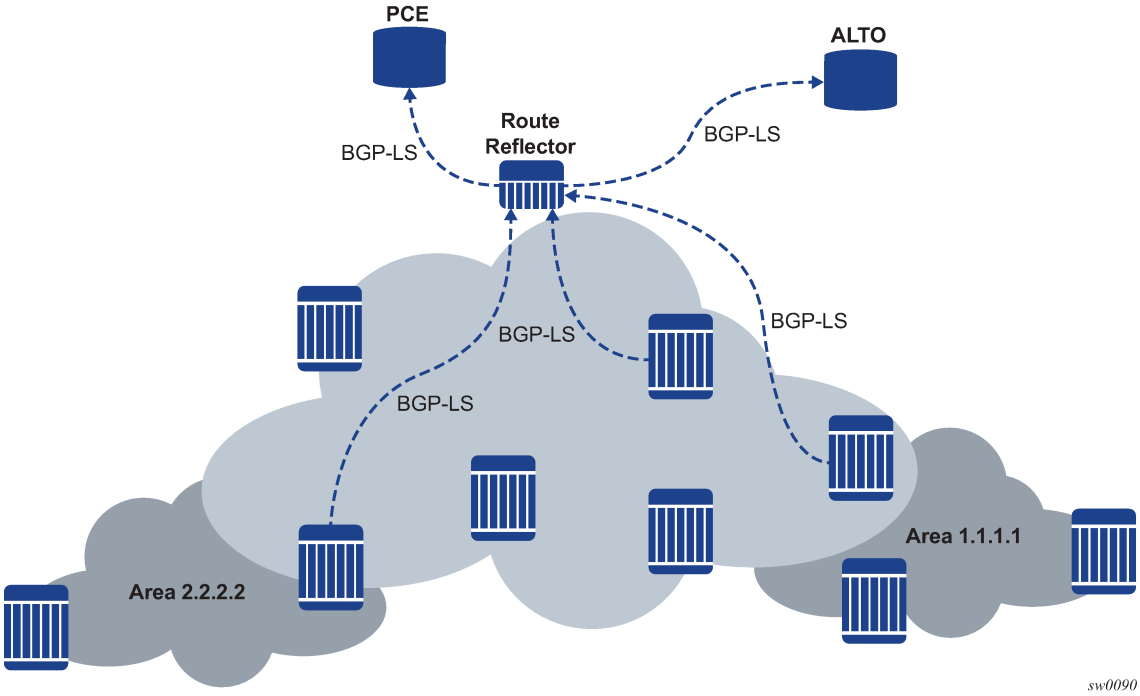
5.8.7 BGP-LS

BGP-LS is a new BGP address family that is intended to distribute IGP topology information to external servers such as Application Later Traffic Optimization (ALTO) or Path Computation Engines (PCE) servers. These external traffic engineering databases can then use this information when calculating optimal paths.

BGP-LS provides external ALTO and PCE servers with topology information for a multi-area or multilevel network. Through the use of one or two BGP-LS speakers per area or level, the external ALTO or PCE servers can receive full topology information for the entire network. The BGP-LS information can also be distributed through route reflectors supporting the BGP-LS to minimize the peering requirements.

Figure 31: Example BGP-LS network shows an example BGP-LS network.

Figure 31: Example BGP-LS network



5.8.7.1 Supported BGP-LS components

The following BGP-LS components are currently supported.

Protocol-ID

- IS-IS level 1
- IS-IS level 2
- OSPFv2
- BGP

NLRI Types

- Node NLRI
- Link NLRI
- IPv4 Topology Prefix NLRI
- IPv6 Topology Prefix NLRI
- SRv6 SID

Node Descriptor TLVs

- 512 — Autonomous System
- 513 — BGP-LS Identifier
- 514 — OSPF Area-ID
- 515 — IGP Router-ID

Node Attribute TLVs

- 263 — Multitopology ID (IS-IS only)
- 266 — Node MSD
- 1024 — Node Flag Bits (O and B bits supported)
- 1026 — Node Name
- 1027 — IS-IS Area Identifier
- 1028 — IPv4 Router ID of Local Node (Only supported for IS-IS; Only when received from remote router. Not supported for local links.)
- 1029 — IPv6 Router-ID of Local Node (only supported for IS-IS)
- 1032 — S-BFD Discriminators
- **Segment Routing**
  - 1034 — SR Capabilities
  - 1035 — SR Algorithm
  - 1038 — SRv6 Capabilities
  - 1039 — Flexible Algorithm Definition
  - 1040 — Flexible Algorithm Exclude-Any Affinity
  - 1041 — Flexible Algorithm Include-Any Affinity
  - 1042 — Flexible Algorithm Include-All Affinity
  - 1043 — Flexible Algorithm Definition Flags
  - 1045 — Flexible Algorithm Exclude SRLG

Link Descriptors TLVs

- 258 — Link Local/Remote Identifiers
- 259 — IPv4 interface address
- 260 — IPv4 neighbor address
- 261 — IPv6 interface address (only supported for IS-IS)
- 262 — IPv6 neighbor address (only supported for IS-IS)
- 263 — Multitopology ID (only supported for IS-IS)

Link Attributes TLVs

- 267 — Link MSD (Only when received from remote router. Not supported for local links.)
- 1088 — Administrative group (color)
- 1089 — Maximum link bandwidth
- 1090 — Max. reservable link bandwidth
- 1091 — Unreserved bandwidth
- 1092 — TE Default Metric
- 1095 — IGP Metric
- 1096 — Shared Risk Link Group

- 1114 — Unidirectional Link Delay (Only when received from remote router. Not supported for local links.)
- 1115 — Min/Max Unidirectional Link Delay
- 1116 — Unidirectional Delay Variation (Only when received from remote router. Not supported for local links.)
- 1117 — Unidirectional Link Loss (Only when received from remote router. Not supported for local links.)
- 1118 — Unidirectional Residual Bandwidth (Only when received from remote router. Not supported for local links.)
- 1119 — Unidirectional Available Bandwidth (Only when received from remote router. Not supported for local links)
- 1120 — Unidirectional Utilized Bandwidth (Only when received from remote router. Not supported for local links)
- 1122 — Application Specific Link Attributes
- 1173 — Extended Administrative Group
- **Segment Routing**
  - 1099 — Adjacency Segment Identifier
  - 1100 — LAN Adjacency Segment Identifier
  - 1106 — SRv6 End.X SID (only supported for IS-IS)
  - 1107 — IS-IS SRv6 LAN End.X SID (only supported for IS-IS)

Prefix Descriptors TLVs

- 263 — Multitopology ID (only supported for IS-IS)
- 264 — OSPF Route Type (only Intra-Area and Inter-Area)
- 265 — IP Reachability Information

Prefix Attributes TLVs

- 1044 — Flexible Algorithm Prefix Metric
- 1152 — IGP Flags (only D flag supported [only supported for IS-IS])
- 1155 — Prefix Metric
- **Segment Routing**
  - 1158 — Prefix SID
  - 1159 — Range (prefix-SID and sub-TLV only)
  - 1162 — SRv6 Locator (only supported for IS-IS)
  - 1170 — IGP Prefix Attributes

SRv6 SID

- **SRv6 SID Descriptor TLVs**
  - 263 — Multitopology ID (only supported for IS-IS)
  - 518 — SRv6 SID Information (only supported for IS-IS)
- **SRv6 SID Attribute TLVs**
  - 1250 — SRv6 Endpoint Behavior (only supported for IS-IS)

– 1152 — SRv6 SID Structure (only supported for IS-IS)

5.8.8 BGP-LU traffic statistics

SR OS can collect BGP-LU traffic statistics.

Traffic statistics can be collected on egress datapaths. This requires the use of **egress-statistics** keyword when creating an import policy and that the BGP tunnel exists for the corresponding prefix. If multiple paths exist (for example, ECMP), a single statistical index is allocated and reflects the traffic sent over all paths.

Traffic statistics can also be collected on ingress datapaths if the label is assigned and effectively advertised per prefix. This typically requires the use of **advertise-label per-prefix** when creating the import policy and applies whether the **sr-label-index** keyword is in use or not. However, there are cases where this may not result in a per-prefix label advertisement. When a non-BGP route (for example, static route) is requested to be advertised (**advertise-inactive**) with a Label Per Prefix (LPP) policy but it exists as an active RTM route and as inactive BGP route, the system does not use the LPP but instead uses the LPNH policy. Statistics are not counted for this prefix. An imported (local loopback) SR labeled route can also be configured to use the **ingress-statistics** keyword by using a route table import policy under **rib-management** (either label-ipv4 or label-ipv6).

Overall, BGP-LU statistics apply at the:

- PE and forwarding RR on egress
- ASBR on both ingress and egress

Note:

- Only host prefixes (/32 and /128) are supported on egress statistics.
- Host and non-host prefixes are supported on ingress statistics.

Control messages sent over the BGP-LU tunnel are accounted for in traffic statistics.

BGP-LU statistics are not supported for imported LDP routes (ldp-bgp stitching) or for VPN labels (for example, inter-AS B or C).

5.8.9 BGP Egress Peer Engineering using BGP Link State

BGP Egress Peer Engineering (BGP EPE) extends segment routing (SR) capabilities beyond the AS boundary toward directly attached EBGP peers. Operators can use a central controller to enforce more programmatic control of traffic distribution across these BGP peering links.

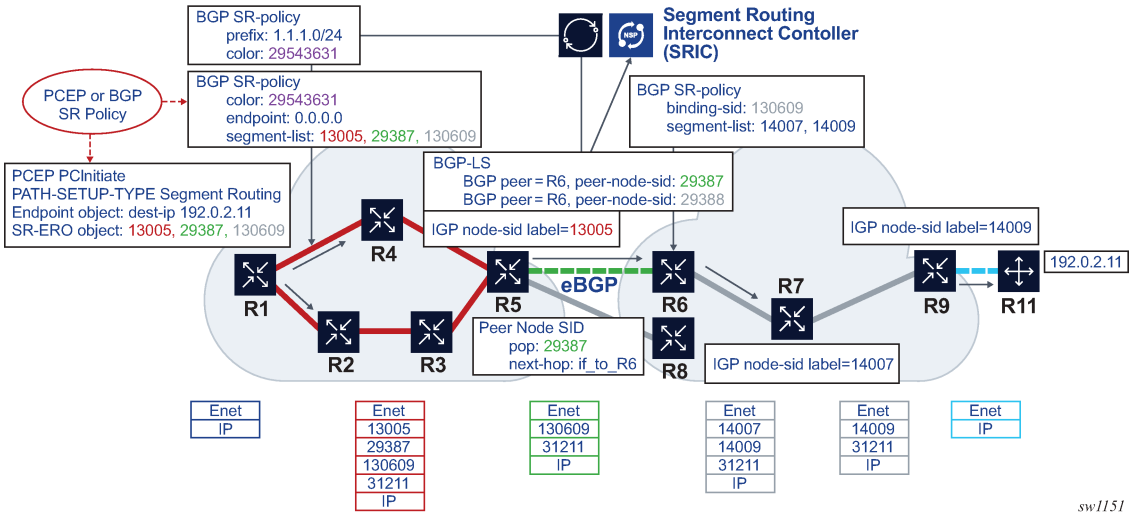
An SR SID can be allocated to a BGP peering segment and advertised in BGP Link State (BGP-LS) toward a controller, such as the Nokia NSP. The instantiation of the following BGP peering segments is supported:

- an eBGP or iBGP peer (peer node SIDs as defined in RFC 8402, *Segment Routing Architecture*)
- a link to such a peer (peer adjacency SIDs as defined in RFC 8402, *Segment Routing Architecture*)

The controller includes the specific SID in the path for an SR-TE LSP or SR policy, which it programs at the head-end LER. EPE enables a head-end router to steer traffic across a downstream peering link to a node, for traffic optimization, resiliency, or load-balancing purposes.

Figure 32: EPE example use case shows an example use case for BGP EPE.

Figure 32: EPE example use case



In this example, there are two IGP domains with eBGP running between R5 and R6, and between R5 and R8. These adjacencies are not visible to R1, which is external to the IGP domain. At R5, separate peer node SIDs are allocated for R6 and R8. The peer node SIDs are advertised to the Nokia NSP in BGP-LS. This allows the NSP to compute a path across either the R5-R6 adjacency or the R5-R8 adjacency by including the appropriate peer node SID in the path. In the preceding use case, the R5-R6 adjacency is preferable. This peer node SID can be included in either the SR-ERO of an SR-TE LSP that is computed by PCEP or the segment list of BGP SR policy that is programmed at R1. Traffic on this LSP or SR policy is, therefore, steered across the required peering.

EPE is supported for BGP neighbors with either eBGP or iBGP sessions. SR peer node SIDs and peer adjacency SIDs are supported. The SID labels are dynamically allocated from the local label space on the node and advertised in BGP-LS using the encoding specified in section 4 of *draft-ietf-idr-bgpls-segment-routing-epe-19*.

The peering node can behave as both an LSR and an LER for steering traffic toward the peering segment. Both ILM and LTN entries are programmed for peer node SIDs and peer adjacency SIDs, with a label swap to or push of an implicit null label.

Note: BGP EPE only supports neighbor nodes that are directly connected to the egress router (for example, not indirectly through tunnels).

ECMP is supported by default if there are multiple peer adjacency SIDs. BGP only allocates peer adjacency SIDs to the ECMP set of next hops toward the peer node. For non-ECMP next hops, only a peer adjacency SID is allocated and it is advertised if all ECMP sets go down.

LSP ping and LSP trace echo requests are supported by including a label representing a peer node SID or peer adjacency SID in a NIL FEC of the target FEC stack. An EPE router can validate and respond to an LSP ping or trace echo request containing this FEC.

5.8.9.1 Configuring BGP EPE

In addition to enabling the BGP-LS route family for a BGP neighbor, the following CLI is required to send the Egress Peering Segments described in BGP Egress Peer Engineering using BGP Link State using the NLRI Type 2 with protocol ID set to BGP-EPE.

CLI syntax

```
configure>router>bgp
   egress-peer-engineering {
     admin-state {enable | disable}
   }
```

When **egress-peer-engineering** is administratively enabled, BGP registers with SR and the router starts advertising any peer node and peer adjacency SIDs in BGP-LS.

To allocate peer node and peer adjacency SIDs, use the following syntax to configure the **egress-engineering** command and enable BGP-EPE for a BGP neighbor or group.

CLI syntax

```
configure>router>bgp
  group
    neighbor <a.b.c.d> {
      egress-engineering {
        admin-state {enable | disable}
      }

configure>router>bgp
  group
    egress-engineering {
      admin-state {enable | disable}
    }
```

The BGP **egress-engineering** at the neighbor level overrides the group level configuration. When a neighbor does not have an **egress-engineering** configuration context, the group configuration is inherited in the following cases.

- If the group does not have an **egress-engineering** configuration, **egress-engineering** is disabled for the neighbor.
- If the group has an **egress-engineering** configuration in the default disabled state, **egress-engineering** is disabled for the neighbor.
- If the group has an enabled **egress-engineering** configuration, **egress-engineering** is enabled for the neighbor.

When a neighbor has **egress-engineering** configured and in the default disabled state, **egress-engineering** is disabled for the neighbor, irrespective of the disabled, enabled, or no-context configuration at the group level. When a neighbor has **egress-engineering** configured and enabled, **egress-engineering** is enabled for the neighbor, irrespective of the disabled, enabled, or no-context configuration at the group level.

By default, enabling **egress-engineering** at the peer or group level causes SID values (MPLS labels) to be dynamically allocated for the peer node segment and the peer adjacency segments. Although the labels are assigned when the neighbor or group is configured, they are not programmed until the adjacency comes up. Peer node segments are derived from the BGP next hops used to reach a specific peer. If the node reboots, these dynamically allocated label values may change and are re-announced in BGP-LS.

If a BGP neighbor goes down, the router advertises a delete for all SIDs associated with the neighbor and deprograms them from the IOM. However, the label values for the SIDs are not released and the router re-advertises the same values when the BGP neighbor comes back up.

If a BGP neighbor is deleted from the configuration or is shut down, or **egress-engineering** is disabled, the router advertises a delete for all SIDs associated with the neighbor and deprograms them from the IOM. The router also releases the label values for the SIDs.

5.8.10 BGP Egress Peer Engineering using Labeled Unicast

Egress Peer Engineering (EPE) allows an ingress PE or source host in an Autonomous System (AS) to use a specific egress peering router and a specific external interface or neighbor of that peering router to reach IP destinations external to the AS.

The following solutions implement EPE:

- Use segment routing and BGP-LS to signal peer node SIDs and peer adjacency SIDs to a PCE or controller that computes and programs the steering instructions that make use of these SIDs. For more information, see BGP Egress Peer Engineering using BGP Link State.
- Use a distributed EPE solution relying on BGP Labeled Unicast (LU) routes and recursive BGP route resolution. BGP EPE using LU does not depend on segment routing, BGP-LS, or a controller. It only depends on the appropriate configuration in the EPE border routers and ingress PE routers. This is the simpler solution and is described in what follows.

To enable an egress border router (with EPE peers) to support BGP-LU based EPE, use the **egress-peer-engineering-label-unicast** command for the base router BGP groups and neighbors in the **configure router bgp group** and **configure router bgp group neighbor** contexts so that all potential EPE peers are covered. When this command is applied, BGP generates a labeled unicast route for the /32 or /128 prefix that corresponds to each EPE peer. These routes can be advertised to other routers to recursively resolve unlabeled BGP routes for AS external destinations. The BGP-LU EPE routes can resolve unlabeled BGP routes only when the unlabeled BGP routes are advertised in the local AS with the next-hop unchanged. In general, the unlabeled routes should be advertised in the local AS using add-paths or best-external so that multiple exit paths are available to the route selection process in ingress PE routers.

The system generates an EPE route for a peer address when all the following conditions are met:

- The peer address is not an IPv6 link-local address.
- The peer session is up.
- The detected peer type is EBGP (and not IBGP or confederation-EBGP).
- The peer address is resolved by a direct interface route (that is, the peer is a single-hop connected peer).

The system withdraws an EPE route if any of these conditions is no longer met; for example, the peer session goes down. After the withdrawal reaches the ingress PE routers, the BGP route selection process must select a different EPE exit point.

In line cards supporting FP3 or later FP technology, the label that is advertised with an EPE route is programmed with an action that depends on the state of the interface toward the EPE peer:

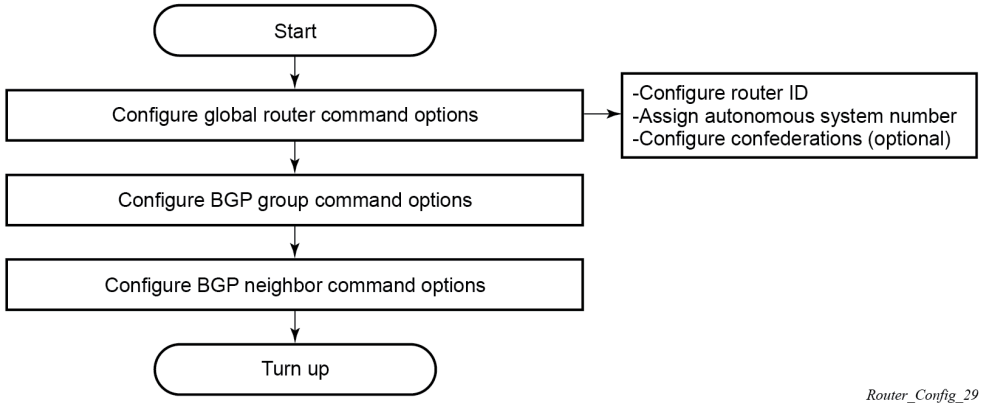
- If the interface is up, the system forwards a packet received with the EPE label as a bottom-of-stack label without IP header lookup (and with the EPE label removed) to the EPE peer.
- If the interface is down, the system forwards a packet received with the EPE label as a bottom-of-stack label as an IP packet (with the EPE label removed, based on the IP FIB lookup).

These datapath optimizations enable fast reroute behavior (BGP FRR) when the interface toward an EPE peer goes down. The BGP FRR behavior is only available with FP3 or later FP technology.

5.9 BGP configuration process overview

Figure 33: BGP configuration and implementation flow displays the process to provision basic BGP parameters.

Figure 33: BGP configuration and implementation flow



5.10 Configuration notes

This section describes BGP configuration restrictions.

5.10.1 General

- Before BGP can be configured, the router ID and autonomous system should be configured.
- BGP must be added to the router configuration. There are no default BGP instances on a router.

5.10.1.1 BGP defaults

The following list summarizes the BGP configuration defaults.

- By default, the router is not assigned to an AS.
- A BGP instance is created in the administratively enabled state.
- A BGP group is created in the administratively enabled state.
- A BGP neighbor is created in the administratively enabled state.
- No BGP router ID is specified. If no BGP router ID is specified, BGP uses the router system interface address.

- The router BGP timer defaults are generally the values recommended in IETF drafts and RFCs (see BGP MIB notes)
- If no *import* route policy statements are specified, then all BGP routes are rejected.
- If no *export* route policy statements are specified, then no routes are advertised.

5.10.1.2 BGP MIB notes

The router implementation of the RFC 1657 MIB variables listed in Table 13: SR OS and IETF MIB variations differs from the IETF MIB specification.

Table 13: SR OS and IETF MIB variations

MIB variable	Description	RFC 1657 allowed values	SR OS allowed Values
bgpPeerMinRoute AdvertisementInterval	Time interval in seconds for the MinRouteAdvertisementInterval timer. The suggested value for this timer is 30.	1 to 65535	1 to 255 A value of 0 is supported when the rapid-update command is applied to an address family that supports it.

If SNMP is used to set a value of X to the MIB variable in Table 13: SR OS and IETF MIB variations , there are three possible results:

Table 14: MIB variable with SNMP

Condition	Result
X is within IETF MIB values and X is within SR OS values	SNMP set operation does not return an error MIB variable set to X
X is within IETF MIB values and X is outside SR OS values	SNMP set operation does not return an error MIB variable set to "nearest" SR OS supported value (for example, SR OS range is 2 - 255 and X = 65535, MIB variable is set to 255) Log message generated
X is outside IETF MIB values and X is outside SR OS values	SNMP set operation returns an error

When the value set using SNMP is within the IETF allowed values and outside the SR OS values as specified in Table 13: SR OS and IETF MIB variations and Table 14: MIB variable with SNMP , a log message is generated.

The log messages that display are similar to the following log messages:

Sample Log Message for setting bgpPeerMinRouteAdvertisementInterval to 256

```
535 2006/11/12 19:40:53 [Snmpd] BGP-4-bgpVariableRangeViolation: Trying to set
bgpPeerMinRouteAdvInt to 256 - valid range is [2-255] - setting to 255
```

Sample Log Message for setting bgpPeerMinRouteAdvertisementInterval to 1

```
566 2006/11/12 19:44:41 [Snmpd] BGP-4-bgpVariableRangeViolation: Trying to set
bgpPeerMinRouteAdvInt to 1 - valid range is [2-255] - setting to 2
```

5.11 Configuring BGP with CLI

This section provides information to configure BGP using the command line interface.

5.11.1 Configuration overview

5.11.1.1 Preconfiguration requirements

Before BGP can be implemented, the following entities must be configured:

- **The autonomous system (AS) number for the router**

  An AS number is a globally unique value which associates a router to a specific autonomous system. This number is used to exchange exterior routing information with neighboring ASs and as an identifier of the AS itself. Each router participating in BGP must have an AS number specified.

  To implement BGP, the AS number must be specified in the **config**>**router** context.

- **Router ID**

  The router ID is the IP address of the local router. The router ID identifies a packet's origin. The router ID must be a valid host address.

5.11.1.2 BGP hierarchy

BGP is configured in the **config**>**router**>**bgp** context. Three hierarchical levels are included in BGP configurations:

- global level
- group level
- neighbor level

Commands and parameters configured on the global level are inherited to the group and neighbor levels although parameters configured on the group and neighbor levels take precedence over global configurations.

5.11.1.3 Internal and external BGP configuration

A BGP system comprises of ASs which share network reachability information. Network reachability information is shared with adjacent BGP peers. BGP supports two types of routing information exchanges.

- **External BGP (EBGP) is used between ASs**

EBGP speakers peer to different ASs and typically share a subnet. In an external group, the next hop is dependent upon the interface shared between the external peer and the specific neighbor. The multihop command must be specified if an EBGP peer is more than one hop away from the local router.

- **Internal BGP (IBGP) is used within an AS**

  IBGP peers belong to the same AS and typically does not share a subnet. Neighbors do not have to be directly connected to each other. Because IBGP peers are not required to be directly connected, IBGP uses the IGP path (the IP next-hop learned from the IGP) to reach an IBGP peer for its peering connection.

5.11.1.4 Default external BGP route propagation behavior without policies

A newly created or existing BGP instance, group, or EBGP neighbor in a classic interface (the classic CLI and SNMP) maintains backwards compatibility with the insecure default to advertise and receive all routes. It is not compliant with RFC 8212. The secure default behavior must be enabled using the **ebgp-default-reject-policy** command in these cases.

A newly created BGP instance, group, or EBGP neighbor in a model-driven interface (the MD-CLI, NETCONF, or gRPC) applies the secure default behavior to reject all routes. It is compliant with RFC 8212. The secure behavior can be disabled using the **ebgp-default-reject-policy** command. However, Nokia recommends configuring import and export policies that express the intended routing instead of using the insecure default behavior. Defining an empty policy does not match any routes, an accept must match the route through an **action accept** or **default-action accept** statement.

The default behavior is inherited from the BGP instance to the group to an EBGP neighbor.

The import and export policies that are applied can be displayed using **info detail** or the **show router bgp neighbor** commands.

Table 15: Default EBGP route propagation behavior shows the default EBGP route propagation behavior according to how the neighbor was configured.

Table 15: Default EBGP route propagation behavior

Management-interface configuration-mode	**Classic**	**Mixed**		**Model-driven**
BGP instance, group, or EBGP neighbor	**Configured in a classic interface**	**Configured in a classic interface**	**Configured in a model-driven interface**	**Configured in a model-driven interface**
Configured before Release 19.5.R1	Default accept	Default accept	Default accept	Default accept
Configured in Release 19.5.R1 or higher	Default accept	Default accept	Default reject[3]	Default reject[3]
ISSU to Release 19.5.R1 or higher	Default accept	Default accept	According to rows 1 and 2[3]	According to rows 1 and 2[3]

[3] Indicates a default behavior change

Management-interface configuration-mode	Classic	Mixed		Model-driven
BGP instance, group, or EBGP neighbor	**Configured in a classic interface**	**Configured in a classic interface**	**Configured in a model-driven interface**	**Configured in a model-driven interface**
Reboot with Release 19.5.R1 or higher	Default accept	Default accept	According to rows 1 and 2[3]	Default reject[3]

Caution: Configuration in model-driven management-interface configuration-mode made before Release 19.5.R1 changes from default accept to default reject if the router is rebooted with Release 19.5.R1 or higher. Configuration in classic or mixed mode maintains the existing default accept behavior.

5.11.2 Basic BGP configuration

This section provides information to configure BGP and configuration examples of common configuration tasks. The minimal BGP parameters that need to be configured are shown below.

- An autonomous system number for the router.
- A router ID. If a new or different router ID value is entered in the BGP context, then the new value takes precedence and overwrites the router-level router ID.
- A BGP peer group.
- A BGP neighbor with which to peer.
- A BGP peer-AS that is associated with the above peer.

The BGP configuration commands have three primary configuration levels: **bgp** for global configurations, group **name** for BGP group configuration, and neighbor **ip-address** for BGP neighbor configuration. Within the different levels, many of the configuration commands are repeated. For the repeated commands, the command that is most specific to the neighboring router is in effect, that is, neighbor settings have precedence over group settings which have precedence over BGP global settings.

Following is an example configuration that includes the above parameters. The other parameters shown below are optional:

```
info
#--------------------------------------------------
echo "IP Configuration"
#--------------------------------------------------
...
        autonomous-system 200
        confederation 300 members 200 400 500 600
        router-id 10.10.10.103
#--------------------------------------------------
...
#--------------------------------------------------
echo "BGP Configuration"
#--------------------------------------------------
        bgp
            graceful-restart
            exit
            cluster 0.0.0.100
            export "direct2bgp"
```

```
            router-id 10.0.0.12
            group "To_AS_10000"
                connect-retry 20
                hold-time 90
                keepalive 30
                local-preference 100
                remove-private
                peer-as 10000
                neighbor 10.0.0.8
                    description "To_Router B - EBGP Peer"
                    connect-retry 20
                    hold-time 90
                    keepalive 30
                    local-address 10.0.0.12
                    passive
                    preference 99
                    peer-as 10000
                exit
            exit
            group "To_AS_30000"
                connect-retry 20
                hold-time 90
                keepalive 30
                local-preference 100
                remove-private
                peer-as 30000
                neighbor 10.0.3.10
                    description "To_Router C - EBGP Peer"
                    connect-retry 20
                    hold-time 90
                    keepalive 30
                    peer-as 30000
                exit
            exit
            group "To_AS_40000"
                connect-retry 20
                hold-time 30
                keepalive 30
                local-preference 100
                peer-as 65206
                neighbor 10.0.0.15
                    description "To_Router E - Sub Confederation AS 65205"
                    connect-retry 20
                    hold-time 90
                    keepalive 30
                    local-address 10.0.0.12
                    peer-as 65205
                exit
            exit
        exit
#--------------------------------------------------
....
```

5.11.3 Common configuration tasks

About this task

This task provides a brief overview of the tasks that must be performed to configure BGP and provides the CLI commands. To enable BGP, one AS must be configured and at least one group must be configured which includes neighbor (system or IP address) and peering information (AS number).

All BGP instances must be explicitly created on each router. After the instances are created, BGP is administratively enabled.

Configuration planning is essential to organize ASs and the SRs within the ASs, and determine the internal and external BGP peering.

To configure a basic autonomous system, perform the following steps.

Procedure

Step 1. Prepare a plan detailing the autonomous systems, the router belonging to each group, group names, and peering connections.

Step 2. Associate each router with an autonomous system number.

Step 3. Configure each router with a router ID.

Step 4. Associate each router with a peer group name.

Step 5. Specify the local IP address that is used by the group or neighbor when communicating with BGP peers.

Step 6. Specify neighbors.

Step 7. Specify the autonomous system number associated with each neighbor.

5.11.3.1 Creating an autonomous system

Before BGP can be configured, the autonomous system must be configured first. In BGP, routing reachability information is exchanged between autonomous systems (ASs). An AS is a group of networks that share routing information. The **autonomous-system** command associates an autonomous system number to the router being configured. The autonomous-system command is configured in the **config**>**router** context.

Use the following CLI syntax to associate a router to an autonomous system.

CLI syntax

```
config>router# autonomous-system autonomous-system
```

The router series supports 4 bytes AS numbers by default. This means autonomous-system can have any value from 1 to 4294967295. The following example displays autonomous system configuration command usage.

Example

```
config>router# autonomous-system 100
```

The following example displays the autonomous system configuration:

```
ALA-B>config>router# info
#------------------------------------------
# IP Configuration
#------------------------------------------
        interface "system"
            address 10.10.10.104/32
        exit
        interface "to-103"
            address 10.0.0.104/24
```

```
            port 1/1/1
        exit
        autonomous-system 100

#------------------------------------------
ALA-B>config>router#
```

5.11.3.2 Configuring a router ID

In BGP, routing information is exchanged between autonomous systems. The BGP router ID, expressed like an IPv4 address, uniquely identifies the router. It can be set to be the same as the system interface address.

It is possible to configure an SR OS to operate with an IPv6 only BOF and no IPv4 system interface address. When configured in this manner, the operator must explicitly define IPv4 router IDs for protocols such as OSPF and BGP as there is no mechanism to derive the router ID from an IPv6 system interface address.

If a new or different router ID value is entered in the BGP context, then the new router ID value is used instead of the router ID configured on the router level, system interface level, or inherited from the MAC address. The router-level router ID value remains intact. The router ID used by BGP is selected in the following order:

- the router-id configured under **config**>**router**>**bgp**
- the router-id configured under **config**>**router**
- the system interface IPv4 address
- the last 4 bytes of the system MAC address

When configuring a new router ID outside of the **config**>**router**>**bgp** context, BGP is not automatically restarted with the new router ID; the next time BGP is (re) initialized the new router ID is used. An interim period of time can occur when different protocols use different router IDs. To force the new router ID, issue the shutdown and no shutdown commands for BGP or restart the entire router. Use the following CLI syntax to configure the router ID for multiple protocols.

CLI syntax

```
config>router# router-id router-id
```

The following example displays router ID configuration command usage.

Example

```
config>router# router-id 10.10.10.104
```

The following example displays the router ID configuration:

```
ALA-B>config>router# info
----------------------------------------------
# IP Configuration
#------------------------------------------
        interface "system"
            address 10.10.10.104/32
        exit
        interface "to-103"
            address 10.0.0.104/24
            port 1/1/1
```

```
        exit
        autonomous-system 100
        router-id 10.10.10.104
#------------------------------------------
...
ALA-B>config>router#
```

5.11.3.3 BGP confederations

About this task

Perform the following steps to configure a BGP confederation.

Procedure

Step 1. Configure the autonomous system number of the confederation using the confederation command in the **config>router** context.

Step 2. Configure the BGP confederation members using the confederation command in the **config>router** context.

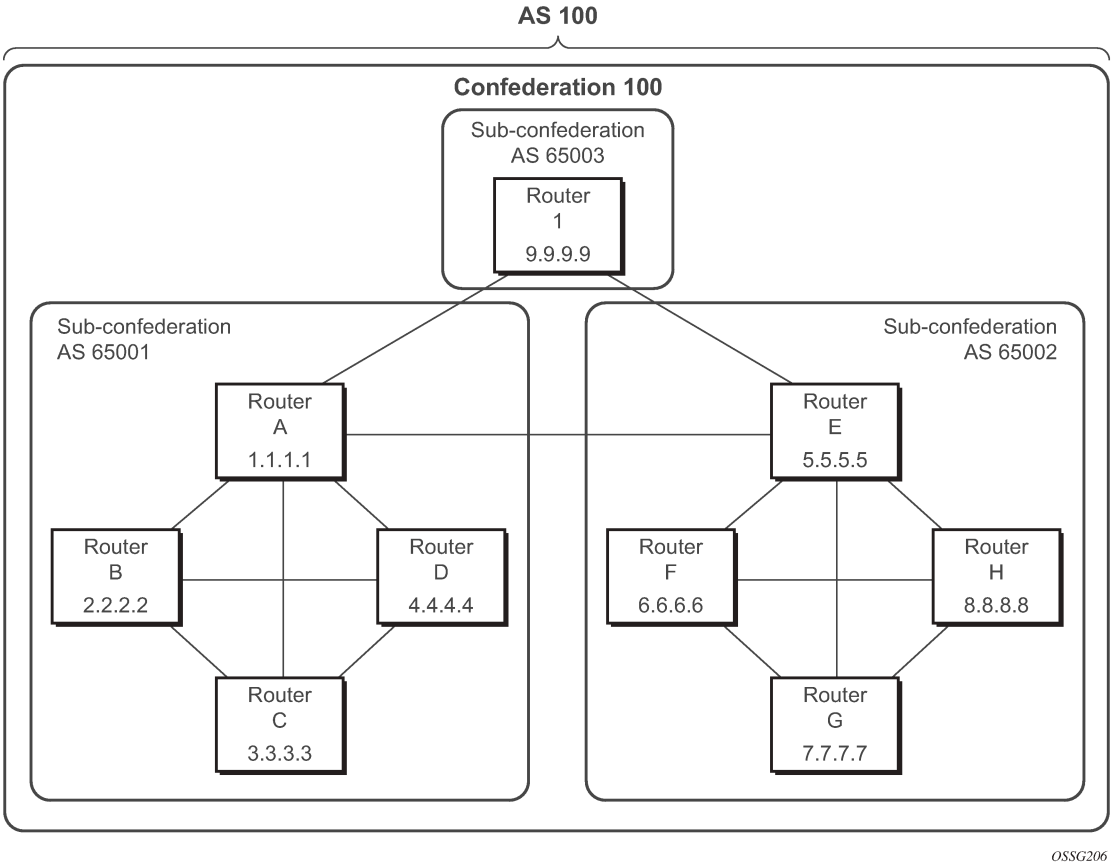
Step 3. Configure IBGP peering within the (local) sub-confederation.

Step 4. Configure one or more confed-EBGP peerings to peers in other neighboring sub-confederations.

Example

Figure 34: Confederation network diagram example shows a confederation network diagram example.

Figure 34: Confederation network diagram example



The following configuration displays the minimum BGP configuration for routers in sub-confederation AS 65001 described in Figure 34: Confederation network diagram example.

```
ALA-A
    config router
        autonomous-system 65001
        confederation 100 members 65001 65002 65003
        bgp
            group confed1
                peer-as 65001
                neighbor 2.2.2.2
                exit
                neighbor 3.3.3.3
                exit
                neighbor 4.4.4.4
                exit
            exit
            group external_confed
                neighbor 5.5.5.5
                    peer-as 65002
                exit
                neighbor 9.9.9.9
                    peer-as 65003
                exit
            exit
```

```
        exit
    exit

ALA-D
    config router
        autonomous-system 65001
        confederation 100 members 65001 65002 65003
        bgp
            group confed1
                peer-as 65001
                neighbor 1.1.1.1
                exit
                neighbor 2.2.2.2
                exit
                neighbor 3.3.3.3
                exit
            exit
        exit
    exit

ROUTER 1
    config router
        autonomous-system 65003
        confederation 100 members 65001 65002 65003
        bgp
            group confed1
                peer-as 65001
                neighbor 1.1.1.1
                exit
                neighbor 5.5.5.5
                    peer-as 65002
                exit
            exit
        exit
    exit
```

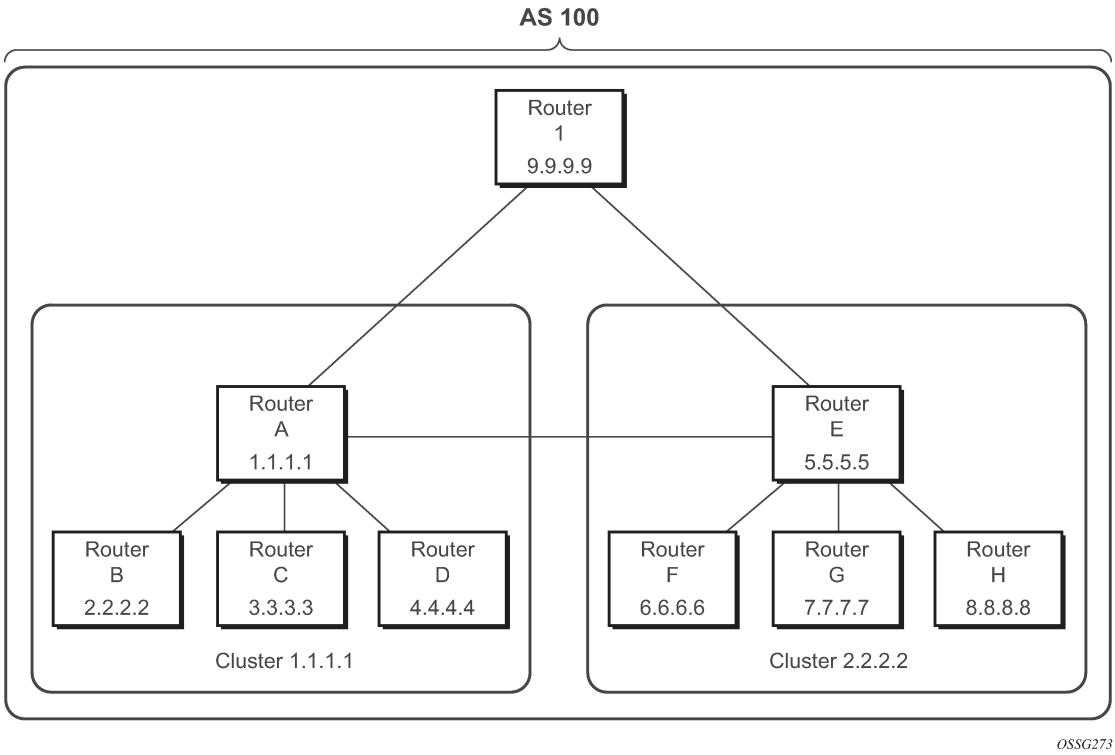
5.11.3.4 BGP router reflectors

In a standard BGP configuration, all BGP speakers within an AS must have a full BGP mesh to ensure that all externally learned routes are redistributed through the entire AS. IBGP speakers do not re-advertise routes learned from one IBGP peer to another IBGP peer. If a network grows, scaling issues could emerge because of the full mesh configuration requirement. Route reflection circumvents the full mesh requirement but still maintains the full distribution of external routing information within an AS.

Autonomous systems using route reflection arrange BGP routers into groups called clusters. Each cluster contains at least one route reflector which is responsible for redistributing route updates to all clients. Route reflector clients do not need to maintain a full peering mesh between each other. They only require a peering to the route reflectors in their cluster. The route reflectors must maintain a full peering mesh between all non-clients within the AS.

Each route reflector must be assigned a cluster ID and specify which neighbors are clients and which are non-clients to determine which neighbors should receive reflected routes and which should be treated as a standard IBGP peer. Additional configuration is not required for the route reflector besides the typical BGP neighbor parameters.

Figure 35: Route reflection network diagram example



The following configuration displays the minimum BGP configuration for routers in Cluster 1.1.1.1 described in Figure 35: Route reflection network diagram example.

Example: MD-CLI

```
[ex:/configure router "Base" bgp]
A:admin@node-2A# info
...
    group "RRs" {
        peer-as 100
    }
    group "cluster1" {
        peer-as 100
        cluster {
            cluster-id 1.1.1.1
        }
    }
    neighbor "2.2.2.2" {
        group "cluster1"
    }
    neighbor "3.3.3.3" {
        group "cluster1"
    }
    neighbor "4.4.4.4" {
        group "cluster1"
    }
    neighbor "5.5.5.5" {
        group "RRs"
    }
    neighbor "9.9.9.9" {
```

```
        group "RRs"
    }

[ex:/configure router "Base" bgp]
A:admin@node-2B# info
    group "cluster1" {
        peer-as 100
        cluster {
            cluster-id 1.1.1.1
        }
    }

[ex:/configure router "Base" bgp]
A:admin@node-2C# info
    group "cluster1" {
        peer-as 100
        cluster {
            cluster-id 1.1.1.1
        }
    }

[ex:/configure router "Base" bgp]
A:admin@node-2D# info
    group "cluster1" {
        peer-as 100
        cluster {
            cluster-id 1.1.1.1
        }
    }
```

Example: classic CLI

```
A:node-2A>config>router>bgp# info
        group cluster1
            peer-as 100
            cluster 1.1.1.1
            neighbor 2.2.2.2
            exit
            neighbor 3.3.3.3
            exit
            neighbor 4.4.4.4
            exit
        exit
        group RRs
            peer-as 100
            neighbor 5.5.5.5
            exit
            neighbor 9.9.9.9
            exit
        exit
    exit

A:node-2B>config>router>bgp# info
        group cluster1
            peer-as 100
            neighbor 1.1.1.1
            exit
        exit
    exit

A:node-2C>config>router>bgp# info
        group cluster1
            peer-as 100
```

```
            neighbor 1.1.1.1
            exit
        exit
    exit

A:node-2D>config>router>bgp# info
        group cluster1
            peer-as 100
            neighbor 1.1.1.1
            exit
        exit
    exit
```

5.11.3.5 BGP components

Use the CLI syntax displayed in the following subsections to configure BGP attributes.

5.11.3.5.1 Configuring group attributes

A group is a collection of related BGP peers. The group name should be a descriptive name for the group. Follow your group, name, and ID naming conventions for consistency and to help when troubleshooting faults.

All parameters configured for a peer group are applied to the group and are inherited by each peer (neighbor), but a group parameter can be overridden on a specific neighbor-level basis.

The following example displays the BGP group configuration:

```
ALA-B>config>router>bgp# info
----------------------------------------------
...
            group "headquarters1"
                description "HQ execs"
                local-address 10.0.0.104
                disable-communities standard extended
                ttl-security 255
                exit
            exit
...
----------------------------------------------
ALA-B>config>router>bgp#
```

5.11.3.5.2 Configuring neighbor attributes

After you create a group name and assign options, add neighbors within the same autonomous system to create IBGP connections and/or neighbors in different autonomous systems to create EBGP peers. All parameters configured for the peer group level are applied to each neighbor, but a group parameter can be overridden on a specific neighbor basis.

The following example displays neighbors configured in group "headquarters1".

```
ALA-B>config>router>bgp# info
----------------------------------------------
...
```

```
            group "headquarters1"
                description "HQ execs"
                local-address 10.0.0.104
                disable-communities standard extended
                ttl-security 255
                neighbor 10.0.0.5
                    passive
                    peer-as 300
                exit
                neighbor 10.0.0.106
                    peer-as 100
                exit
                neighbor 172.5.0.2
                    hold-time 90
                    keepalive 30
                    local-preference 170
                    peer-as 10701
                exit
                neighbor 172.5.1.2
                    hold-time 90
                    keepalive 30
                    local-preference 100
                    min-route-advertisement 30
                    preference 170
                    peer-as 10702
                exit
            exit
...
-------------------------------------------
ALA-B>config>router>bgp#
```

5.11.3.5.3 Configuring route reflection

Route reflection can be implemented in autonomous systems with a large internal BGP mesh to reduce the number of IBGP sessions required. One or more routers can be selected to act as focal points for internal BGP sessions. Several BGP speaking routers can peer with a route reflector. A route reflector forms peer connections to other route reflectors. A router assumes the role as a route reflector by configuring the **cluster** *cluster-id* command. No other command is required unless you want to disable reflection to specific peers.

If you configure the **cluster** command at the global level, then all subordinate groups and neighbors are members of the cluster. The route reflector cluster ID is expressed in dotted-decimal notation. The ID should be a significant topology-specific value. No other command is required unless you want to disable reflection to specific peers.

If a route reflector client is fully meshed, the **disable-client-reflect** command can be enabled to stop the route reflector from reflecting redundant route updates to a client.

The following example displays a route reflection configuration:

Example: MD-CLI

```
[ex:/configure router "Base" bgp]
A:admin@node-2# info
    cluster {
        cluster-id 0.0.0.100
    }
    group "Santa Clara" {
        local-address 10.0.0.103
    }
```

```
    neighbor "10.0.0.91" {
        group "Santa Clara"
        peer-as 100
    }
    neighbor "10.0.0.92" {
        group "Santa Clara"
        peer-as 100
    }
    neighbor "10.0.0.93" {
        group "Santa Clara"
        client-reflect false
        peer-as 100
    }
```

Example: classic CLI

```
A:node-2>config>router>bgp# info
----------------------------------------------
            cluster 0.0.0.100
            group "Santa Clara"
                local-address 10.0.0.103
                neighbor 10.0.0.91
                    peer-as 100
                exit
                neighbor 10.0.0.92
                    peer-as 100
                exit
                neighbor 10.0.0.93
                    disable-client-reflect
                    peer-as 100
                exit
            exit
----------------------------------------------
```

5.11.3.5.4 Configuring a confederation

Reducing a complicated IBGP mesh can be accomplished by dividing a large autonomous system into smaller autonomous systems. The smaller ASs can be grouped into a confederation. A confederation looks like a single AS to routers outside the confederation. Each confederation is identified by its own (confederation) AS number.

To configure a BGP confederation, you must specify a confederation identifier, an AS number expressed as a decimal integer. The collection of autonomous systems appears as a single autonomous system with the confederation number acting as the "all-inclusive" autonomous system number. Up to 15 members (ASs) can be added to a confederation.

The confederation command is configured in the **config>router** context.

Use the following CLI syntax to configure a confederation.

CLI syntax

```
config>router# confederation confed-as-num members
       member-as-num
```

When 4-byte AS number support is not disabled on router, the confederation and any of its members can be assigned an AS number in the range from 1 to 4294967295. The following example displays a confederation configuration command usage.

Example

```
config>router># confederation 1000 members 100 200 300
```

The following example displays the confederation configuration:

```
ALA-B>config>router# info
#------------------------------------------
# IP Configuration
#------------------------------------------
        interface "system"
            address 10.10.10.103/32
        exit
        interface "to-104"
            shutdown
            address 10.0.0.103/24
            port 1/1/1
        exit
        autonomous-system 100
        confederation 1000 members 100 200 300
        router-id 10.10.10.103
#------------------------------------------
ALA-B>config>router#
```

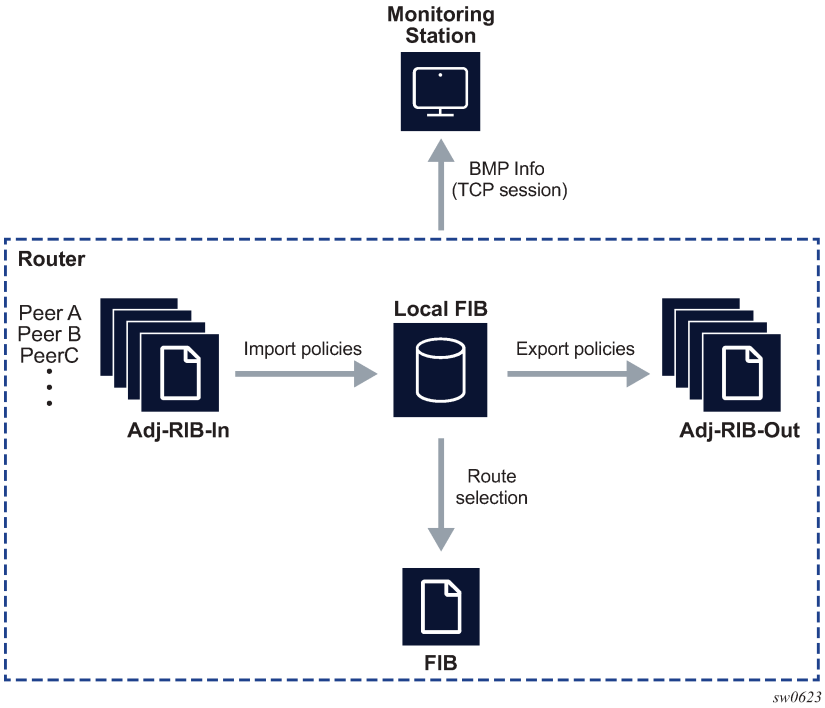
5.11.3.5.5 Configuring BMP

See the BGP monitoring protocol for more information about the protocol. To enable BMP, follow these steps.

1. Configure a BMP station where BMP information is sent.
2. Configure one or more neighbors to be monitored by that station.

Figure 36: Configuring BMP shows a BMP configuration example.

Figure 36: Configuring BMP



5.11.3.5.5.1 Configuring a BMP station

BMP stations are configured at a top-level configuration context (**config**>**bmp**). A BMP station can then monitor BGP in the base router, or in one or more virtual routers for VPRN services. For each instance of BGP, a separate BMP TCP session is set up between the router and that BMP station. A total of eight different BMP stations can be configured on an SR OS router.

The following command creates a BMP station:

config>**bmp**>**station** *station-name* [**create**]

Example

```
config>bmp>station antwerp [create]
```

This creates a BMP station named *antwerp*. The station name must be used when configuring BGP peers to be monitored by this station. The name can also be used in the **show**>**router**>**bmp** CLI commands to display associated information.

The next step is to configure how this station can be reached. To do this, the IP address of the Linux station, or container, on which the BMP collector is run is required. Also, the TCP port number on which the BMP station is listening is needed. BMP does not use a well-known port number.

- A network operator can pick any appropriate TCP port number.
- BMP sessions from an SR OS router can run over either TCP/IPv4 or TCP/IPv6.

The following is an IP address and port number configuration for a BMP station:

```
configure bmp
  station antwerp create
    connection
      station-address 1.2.3.4 port 5000
      exit
    exit
  exit
```

This creates a BMP station that can monitor one or more BGP peers.

Next, assign the BGP peers to be monitored.

- In BGP configuration mode, navigate to the monitor command, **config**>**router**>**bgp**>**monitor**. The **monitor** command can be used at the BGP-instance level the group or neighbor levels.
- Configure up to eight station names that monitor these peers, **config**>**router**>**bgp**>**monitor**>**station** *name*.
- Enable monitoring, **config**>**router**>**bgp**>**monitor**>**no shutdown**. On SR OS routers, both the BMP protocol and each individual station are in a **shutdown** state by default. To allow BMP to start BMP sessions, BMP and the station must be administratively enabled.

```
configure router bgp
  monitor
    station antwerp
    no shutdown
    exit
  exit
```

All peers in the BGP instance of the base router are now monitored by station *antwerp*. The router only sends BMP peer-up and peer-down messages to the BMP station. Sending periodic statistics messages, or reporting incoming BGP routes must be explicitly configured.

5.12 BGP configuration management tasks

This section discusses BGP configuration management tasks.

5.12.1 Modifying an AS number

Modify an AS number on a router but the new AS number is not used until the BGP instance is restarted either by administratively disabling or enabling the BGP instance or by rebooting the system with the new configuration.

Because the AS number is defined in the **configure router** context, not in the BGP configuration context, the BGP instance is not aware of the change. Re-examine the plan detailing the autonomous systems, the SRs belonging to each group, group names, and peering connections. Changing an AS number on a router could cause configuration inconsistencies if associated peer AS values are not also modified as required. At the group and neighbor levels, BGP re-establishes the peer relationships with all peers in the group with the new AS number.

Use the **configure router autonomous-system** command to change an autonomous system number.

The following example displays an AS configuration.

Example: MD-CLI

```
[ex:/configure router "Base"]
A:admin@node-2# info
    autonomous-system 400
  ...
    bgp
        group "1" {
        }
        group "headquarters1" {
        }
        neighbor "10.10.10.103" {
            group "headquarters1"
            peer-as 400
        }
    }
```

Example: classic CLI

```
A:node-2>config>router# info
#------------------------------------------------
echo "IP Configuration"
#------------------------------------------------
...
        autonomous-system 400
...
#------------------------------------------------
echo "BGP Configuration"
#------------------------------------------------
        bgp
            group "1"
            exit
            group "headquarters1"
                neighbor 10.10.10.103
                    peer-as 400
                exit
            exit
            no shutdown
        exit
-------------------------------------------------
```

5.12.2 Modifying a confederation number

Modifying a confederation number causes BGP to restart automatically. Changes immediately take effect.

5.12.3 Modifying the BGP router ID

Changing the router ID number in the BGP context causes the new value to overwrite the router ID configured on the router level, system interface level, or the value inherited from the MAC address. It triggers an immediate reset of all peering sessions.

The following example displays the BGP configuration with the BGP router ID specified.

Example: MD-CLI

```
[ex:/configure router "Base" bgp]
A:admin@node-2# info detail
```

```
 ## apply-groups
 ## apply-groups-exclude
    admin-state enable
 ## description
    connect-retry 120

    router-id 10.0.0.123
    inter-as-vpn false
```

Example: classic CLI

```
A:node-2>config>router>bgp# info detail
----------------------------------------------
            no description
            no family

            router-id 10.0.0.123
          ...
```

5.12.4 Modifying the router-level router ID

Changing the router ID number in the **configure router** context causes the new value to overwrite the router ID derive from the system interface address, or the value inherited from the MAC address.

When configuring a new router ID, protocols are not automatically restarted with the new router ID. The next time a protocol is (re) initialized the new router ID is used. An interim period of time can occur when different protocols use different router IDs. To force the new router ID, do the following or restart the entire router.

- disable the protocol
- configure the new router ID
- enable the protocol

The following commands disable and enable the BGP protocol:

- **MD-CLI**

```
configure router bgp admin-state disable
configure router bgp admin-state enable
```

- **classic CLI**

```
configure router bgp shutdown
configure router bgp no shutdown
```

The following example displays the router ID configuration.

Example: MD-CLI

```
[ex:/configure router "Base"]
A:admin@node-2# info
    autonomous-system 100
    router-id 10.10.10.104

    interface "system" {
        ipv4 {
            primary {
```

```
                address 10.10.10.104
                prefix-length 32
            }
        }
    }
    interface "to-103" {
        port 1/1/1
        ipv4 {
            primary {
                address 10.0.0.104
                prefix-length 24
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config>router# info
#--------------------------------------------------
echo "IP Configuration"
#--------------------------------------------------

        interface "system"
            address 10.10.10.104/32
            no shutdown
        exit
        interface "to-103"
            address 10.0.0.104/24
            port 1/1/1
            no shutdown
        exit

        autonomous-system 100
        router-id 10.10.10.104
#--------------------------------------------------
```

5.12.5 Deleting a neighbor

Use the following CLI command to delete a neighbor:

Note: For classic CLI, you must first administratively disable the group before removing the neighbor.

- **MD-CLI**

```
configure router bgp delete neighbor
```

- **classic CLI**

```
configure router bgp group shutdown
configure router bgp group no neighbor
```

The following example displays the "headquarters1" configuration with the neighbor 10.0.0.103 removed.

Example: MD-CLI

```
[ex:/configure router "Base" bgp]
A:admin@node-2# info
```

```
    router-id 10.0.0.123
    ebgp-default-reject-policy {
        import false
        export false
    }
    group "1" {
    }
    group "headquarters1" {
    }
```

Example: classic CLI

```
A:node-2>config>router>bgp# info
----------------------------------------------
            router-id 10.0.0.123
            group "1"
            exit
            group "headquarters1"
            exit
            no shutdown
----------------------------------------------
```

5.12.6 Deleting groups

To delete a group, the neighbor configurations must first be administratively disabled. After each neighbor is disabled, you must administratively disable the group before removing the group.

The following example displays the administratively disabled neighbors.

Example: MD-CLI

```
[ex:/configure router "Base" bgp]
A:admin@node-2# info
   ...
    group "headquarters1" {
        admin-state enable
    }
    neighbor "10.10.10.103" {
        admin-state disable
        group "headquarters1"
        peer-as 400
    }
    neighbor "10.10.10.105" {
        admin-state disable
        group "headquarters1"
        peer-as 400
    }
```

Example: classic CLI

```
A:node-2>config>router>bgp>group# info
----------------------------------------------
                neighbor 10.10.10.103
                    shutdown
                    peer-as 400
                exit
                neighbor 10.10.10.105
                    shutdown
                    peer-as 400
```

```
                exit
-----------------------------------------------
```

The following example displays the administratively disabled group.

Example: MD-CLI

```
[ex:/configure router "Base" bgp]
A:admin@node-2# info
    ...
    group "headquarters1" {
        admin-state disable
    }
    neighbor "10.10.10.103" {
        admin-state disable
        group "headquarters1"
        peer-as 400
    }
    neighbor "10.10.10.105" {
        admin-state disable
        group "headquarters1"
        peer-as 400
    }
```

Example: classic CLI

```
A:node-2>config>router>bgp# info
----------------------------------------------
           ...
            group "headquarters1"
                shutdown
                neighbor 10.10.10.103
                    shutdown
                    peer-as 400
                exit
                neighbor 10.10.10.105
                    shutdown
                    peer-as 400
                exit
            exit
            no shutdown
----------------------------------------------
```

Note: For classic CLI, if you try to delete the group without shutting down the peer-group, the following message appears:

```
MINOR: CLI BGP peer group must be 'shutdown' before deletion.
```

Use the following command to delete the group:

- **MD-CLI**

```
[ex:/configure router "Base" bgp]
A:admin@node-2# delete group headquarters1
```

- **classic CLI**

```
configure router bgp no group
```

6 Route policies

6.1 Configuring route policies

Nokia's router supports two databases for routing information. The routing database is composed of the routing information learned by the routing protocols. The forwarding database is composed of the routes actually used to forward traffic through a router. In addition, link state databases are maintained by interior gateway protocols (IGPs) such as IS-IS and OSPF.

Routing protocols calculate the best route to each destination and place these routes in a forwarding table. The routes in the forwarding table are used to forward routing protocol traffic, sending advertisements to neighbors and peers.

A routing policy can be configured that will not place routes associated with a specific origin in the routing table. Those routes will not be used to forward data packets to the intended destinations and the routes are not advertised by the routing protocol to neighbors and peers.

Routing policies control the size and content of the routing tables, the routes that are advertised, and the best route to take to reach a destination. Careful planning is essential to implement route policies that can affect the flow of routing information or packets in and traversing through the router. Before configuring and applying a route policy, develop an overall plan and strategy to accomplish your intended routing actions.

There are no default route policies. Each policy must be created explicitly and applied to a routing protocol or to the forwarding table. Policy parameters are modifiable.

6.1.1 Policy statements

SR OS route policy statements consist of a sequence of ordered rules, or entries. When the policy is applied to a routing adjacency, route table, or some other context, then each route associated with that context is evaluated by the rules of the policy, in the specified order, until a matching entry is found or the end of the policy is reached. If a matching entry is found, then its actions are applied to the route. However, if there is no matching entry, then the policy **default-action** is applied to the route.

Some of the match criteria that can be used in a policy entry include:

- IP prefix-list reference
- AS path regular expression
- community list reference
- route properties such as MED, **local-preference**, IGP metric, IGP route type, BGP path type, and so on

In classic mode and mixed-mode every **policy-statement** must have numbered entries.

In full Model-Driven mode, each **policy-statement** can be configured to be **entry-type named** or **entry-type numbered**. Numbered types behave as in classic mode and mixed-mode. In named types, each entry is a specific string-format name up to 255 characters in length, and entries are evaluated in user order (the order they appear in the configuration). MD-CLI and NETCONF management interfaces support **insert** commands that change the order of the named entries in a policy statement. For example, in the MD-CLI configuration context of a **policy-statement** *X*, if a **named-entry entry2** needs to be moved so

that it is evaluated immediately before an existing **named-entry entry1** this can be achieved using the **insert named-entry entry2 before entry1** command.

6.1.1.1 Policy statement chaining and logical expressions

Multiple policy-statement names can be specified in the CLI commands that attach route policies to specific functions. A chain of routing policies is created if a list of two or more policy names is specified in the CLI command. Route policy chaining allows complex route processing logic to be broken into smaller components. This enables the reuse of common functions and facilitates the process of amending and updating route control logic as required.

Each route is evaluated against a policy chain as follows.

- The route is evaluated against the first listed policy. If the route matches an entry with **action next-policy**, or the route matches no entry and the **default-action** is **next-policy** (or there is no **default-action** specified at all), the evaluation continues into the second policy.
- The route is evaluated against the second listed policy. If the route matches an entry with **action next-policy** or a **default-action** of **next-policy** applies, the evaluation continues to the third policy.
- The sequential evaluation of policies continues until the route is accepted or rejected by an entry or explicit **default-action**.

In addition to policy chaining, the SR OS also supports policy logical expressions that enable applications to use complex multiple policy statements. A policy logical expression can be used as a standalone expression or as part of a policy chain. Each policy chain supports a maximum of one logical expression. The logical expression is usually the first element of the policy chain; however, it can appear in a non-initial position as long as its length does not exceed 64 characters.

A route policy logical expression is a string composed of logical operators (keywords AND, OR and NOT), up to 16 route policy names (each up to 64 characters in length and delimited by brackets ( ) and square brackets [ ] to group sub-expressions (with up to three levels of nesting)). The total length of the route policy expression cannot exceed 900 characters.

The following are examples of valid logical expressions in the SR OS CLI syntax:

- "NOT [policyA]"
- "[policyA] AND [policyB] OR [policyC]"
- "NOT ([policyA] OR [policyB] OR [policyC])"

The final result of a route policy evaluation against a logical expression is TRUE or FALSE. The SR OS rules for policy evaluation are as follows.

- If the expression includes a NOT operator followed by a sub-expression in brackets, the expression is flattened using De Morgan's rule.

  For example, the expression "NOT ([policyA] OR [policyB] OR [policyC])" is flattened to: "(NOT [policyA]) AND (NOT [policyB]) AND (NOT [policyC])".
- The usual rules of precedence apply: NOT is the highest priority, then AND, then OR.
- The use of "NOT <*expression*>" negates the TRUE/FALSE result of the expression.
- Where the logic is "(<*expression1*>) AND (<*expression2*>)", if *expression1* is FALSE, the result is FALSE and *expression2* is not evaluated. If *expression1* is TRUE, *expression2* is then evaluated. The final result is TRUE only if both expressions are TRUE.

- Where the logic is "(<*expression1*>) OR (<*expression2*>)", if *expression1* is TRUE, the result is TRUE and *expression2* is not evaluated. If *expression1* is FALSE, *expression2* is then evaluated. The final result is TRUE if either expression is TRUE.
- If the evaluation of a policy terminates with an **action accept**, **action next-entry** or **action next-policy**, the result is TRUE. If the evaluation of a policy terminates with an **action reject** or **action drop**, then the result is FALSE. If a route matches no entry in a policy and there is no specified **default-action**, the implied default action is **next-policy**; if there is no next policy, this translates to the default action for the protocol.
- When a route is evaluated against a policy contained in a logical expression, the route property changes (such as MED, local preference, communities) made by the matching entry or default action apply cumulatively to the route. The result of a cumulative change is that a policy evaluated later in the logical expression (or later in the entire policy chain) may undo or reverse prior changes. A later policy in the logical expression (or policy chain) may also match a route on the basis of route properties that were modified by earlier policies.

When evaluation of the logical expression is complete, the final TRUE or FALSE result is translated back to a traditional action. The FALSE value is translated to action reject; the TRUE value is translated to action accept, action next-policy or action next-entry to match the action of the last policy that produced the TRUE result.

6.1.1.2 Routing policy subroutines

It is possible to reference a routing policy from within another routing policy to construct powerful subroutine based policies.

Up to the three levels of subroutine calls are supported. Policy subroutines produce a final result of TRUE or FALSE through matching and policy entry actions. A policy entry action of 'accept' evaluates to TRUE, and a policy entry action of 'reject' evaluates to FALSE.

When using next-policy action state in the subroutine, the match value is defined by the default action behavior. The action is protocol-dependent. See Default action behavior for information about the default actions that are applied during packet processing.

Note: When subroutines are configured to reject routes, the accept action state can be used as a possible configuration in the subroutine match criteria to return a true-match, and the reject action state can be applied in the main policy entry that has called the subroutine.

If a match is not found during the evaluation of one or more routing policies, the final evaluation returns the accept or the reject provided by the default behavior based on the policy type (import/export) and the destination and/or source protocol.

6.1.1.3 Policy evaluation command

Operators can evaluate a routing policy against a BGP neighbor, routing context, or individual prefix before applying the policy to the neighbor or routing context. This command displays prefixes that are rejected by a policy and what modifications are made by a policy.

6.1.1.4 Exclusive editing for policy configuration

Operators can set an exclusive lock on policy edit sessions. When the exclusive flag is set by an operator that is editing policy, other users (console or SNMP) are restricted from being able to begin, edit, commit, or abort policy. An administrative override is made available to reset the exclusive flag in the event of a session failure.

6.1.1.5 Default action behavior

The default action of a policy applies to a route when the route does not match any of the entries of the policy. If a policy does not have any match entries, all routes are subject to the default action. If no default action is specified and the policy is the last one in a chain of policies, the default action is determined by the protocol that called the policy.

If a default action is defined for one or more of the configured route policies, then the default action is handled as follows.

- The default action can be set to all available action states including accept, reject, next-entry, and next-policy.
- If the action states accept or reject, then the policy evaluation terminates and the appropriate result is returned.
- If a default action is defined and no matches occurred with the entries in the policy, then the default action is used.
- If a default action is defined and one or more matches occurred with the entries of the policy, then the default action is not used.

6.1.1.6 Denied IP unicast prefixes

The Route Table Manager does not inherently restrict any IP prefixes from the forwarding table, but individual routing protocols may not accept prefixes that are not globally routable.

6.1.1.7 Controlling route flapping

Route damping is a controlled acceptance of unstable routes from BGP peers so that any ripple effect caused by route flapping across BGP AS border routers is minimized. The motive is to delay the use of unstable routes (flapping routes) to forward data and advertisements until the route stabilizes.

Nokia’s implementation of route damping is based on the following parameters:

- **Figure of merit**

  A route is assigned a Figure of Merit (FoM), which is proportional to the frequency of flaps. FoM should be able to characterize a route’s behavior over a period of time.

- **Route flap**

  A route flap is not limited to the withdrawn route. It also applies to any change in the AS path or the next hop of a reachable route. A change in AS path or next hop indicates that the intermediate AS or the route-advertising peer is not suppressing flapping routes at the source or during the propagation. Even if the route is accepted as a stable route, the data packets destined for the route could experience unstable routing because of the unstable AS path or next hop.

- **Suppress threshold**

  The threshold is a configured value that, when exceeded, the route is suppressed and not advertised to other peers. The state is considered to be down from the perspective of the routing protocol.

- **Reuse threshold**

  When FoM value falls below a configured reuse threshold and the route is still reachable, the route is advertised to other peers. The FoM value decays exponentially after a route is suppressed. This requires the BGP implementation to decay thousands of routes from a misbehaving peer.

The two events that could trigger the route flapping algorithm are:

- **Route flapping**

  If a route flap is detected within a configured maximum route flap history time, the route's FoM is initialized and the route is marked as a potentially unstable route. Every time a route flaps, the FoM is increased and the route is suppressed if the FoM crosses the suppress threshold.

- **Route reuse timer trigger**

  A suppressed route's FoM decays exponentially. When it crosses the reuse threshold, the route is eligible for advertisement if it is still reachable.

If the route continues to flap, the FoM, with respect to time scale, looks like a sawtooth waveform with the exponential rise and decay of FoM. To control flapping, the following parameters can be configured:

- **half-life**

  The half-life value is the time, expressed in minutes, required for a route to remain stable in order for one half of the FoM value to be reduced. For example, if the half-life value is 6 (minutes) and the route remains stable for 6 minutes, then the new FoM value is 3. After another 6 minutes passes and the route remains stable, the new FoM value is 1.5.

- **max-suppress**

  The maximum suppression time, expressed in minutes, is the maximum amount of time that a route can remain suppressed.

- **suppress**

  If the FoM value exceeds the configured integer value, the route is suppressed for use or inclusion in advertisements.

- **reuse**

  If the suppress value falls below the configured **reuse** value, then the route can be reused.

6.1.2 Regular expressions

SR OS supports using regular expression syntax to match BGP routes based on their AS path or communities.

A regular expression is made up of terms and operators and must be enclosed in quotes.

The elementary term for an AS path regular expression is an AS number. The following are more complex terms:

- a range term composed of two elementary terms separated by a hyphen (-), such as 200-300.
- the dot wild-card (.) character, which matches any elementary term
- a regular expression enclosed in parentheses

- a regular expression enclosed in square brackets, with the brackets used to specify a set of choices of elementary or range terms; for example, [100-300 400] matches any AS number between 100 and 300 (inclusive) or the AS number 400. Precede the elementary and range terms inside the square brackets with a caret sign (^) to reverse the match criteria; for example, [^100-300 400] matches all AS numbers except 100-300 or 400.

The elementary term for a community member regular expression is a single digit. The following are more complex terms:

- a range term composed of two elementary terms separated by a hyphen (-), such as 2-3.
- a colon (:) delimiting the parts of a standard community value
- a regular expression enclosed in parentheses
- a regular expression enclosed in square brackets, with the square brackets used to specify a set of choices of elementary or range terms; for example, [51-37] matches digit 5, any single digit between 1 and 3, or digit 7. Precede the elementary and range terms inside the square brackets with a caret sign (^) to reverse the match criteria; for example, [^13579]0:100 matches the standard community 20:100, but does not match 30:100.

Route target and route origin extended communities can embed two regular expressions separated by one ampersand sign (&). The first expression is applied to the AS value of the community string, and the second expression is applied to the local administrative value.

Large communities can embed three regular expressions separated by two ampersands (&). The first expression is applied to the Global Administrator, the second expression is applied to the Local Data part 1, and the third expression is applied to the Local Data part 2.

A raw hex format can be keyed to represent all extended communities; for example, ext:0102:dc0000020032 is equal to target:65530:20. Hex values "ext:" and "&" are also used to filter extended communities. The first expression is applied to the type or subtype of the extended community, and the second expression is applied to the value. In this case, hex values can also be used in the operands; for example, the value {3,f} matches a minimum of three and a maximum of 15 repetitions of the term.

The following table describes regular expression operators.

Table 16: Regular expression operators

Operator	Description
|	Matches the term on alternate sides of the pipe.
*	Matches multiple occurrences of the term.
?	Matches 0 or 1 occurrence of the term.
+	Matches 1 or more occurrence of the term.
( )	Used to parenthesize so a regular expression is considered as one term.
[ ]	Used to demarcate a set of elementary or range terms.
[^]	Used to demarcate a set of elementary or range terms that are explicitly not matching.
-	Used between the start and end of a range.

Operator	Description
{m,n}	Matches least m and at most n repetitions of the term.
{m}	Matches exactly m repetitions of the term.
{m,}	Matches m or more repetitions of the term.
^	Matches the beginning of the string - only allowed for communities.
$	Matches the end of the string - only allowed for communities.
\	An escape character to indicate that the following character is a match criteria and not a grouping delimiter.
<>	Matches any AS path numbers containing a confederation SET or SEQ.
&	Matches ":" between terms of a community string (applicable to extended communities origin, target, bandwidth, ext only).

Examples of "target:", "origin:" and "ext:" community strings are listed in Table 17: Community strings examples .

Table 17: Community strings examples

Example expression	Example matches
"ext:..&f(.*)[af]$"	Matches the community "ext:0002:fffa0000001a".
'target:1&22	Matches community target:100:221.
target:^1&^22	Matches community target:100:221.
target:(.*)0$&(.*)1$	Matches community target:100:221.
origin:^1&(.*)1$	Matches community origin:100:221.

Examples of AS path and community string regular expressions are listed in Table 18: AS path and community regular expression examples .

Table 18: AS path and community regular expression examples

AS path to match criteria	Regular expression	Example matches
Null AS path	null[4]	Null AS path
AS path is 11	11	11
AS path is 11 22 33	11 22 33	11 22 33
Zero or more occurrences of AS number 11	11*	Null AS path

[4] The **null** keyword matches an empty AS path.

AS path to match criteria	Regular expression	Example matches
		11 11 11 11 11 11 11 … 11
Path of any length that begins with AS numbers 11, 22, 33	11 22 33 .*	11 22 33 11 22 33 400 500 600
Path of any length that ends with AS numbers 44, 55, 66	.* 44 55 66	44 55 66 100 44 55 66 100 200 44 55 66 100 200 300 44 55 66 100 200 300 … 44 55 66
One occurrence of the AS numbers 100 and 200, followed by one or more occurrences of the number 33	100 200 33+	100 200 33 100 200 33 33 100 200 33 33 33 100 200 33 33 33 … 33
One or more occurrences of AS number 11, followed by one or more occurrences of AS number 22, followed by one or more occurrences of AS number 33	11+ 22+ 33+	11 22 33 11 11 22 33 11 11 22 22 33 11 11 22 22 33 33 11 … 11 22 … 22 33 …33
Path whose second AS number must be 11 or 22	(. 11) | (. 22) .* or . (11 | 22) .*	100 11 200 22 300 400 …
Path of length one or two whose second AS number may be 11 or 22	. (11 | 22)?	100 200 11 300 22
Path whose first AS number is 100 and second AS number is either 11 or 22	100 (11 | 22) .*	100 11 100 22 200 300
Either AS path 11, 22, or 33	[11 22 33]	11 22 33
Range of AS numbers to match a single AS number	10-14	10 or 11 or 12 or 13 or 14

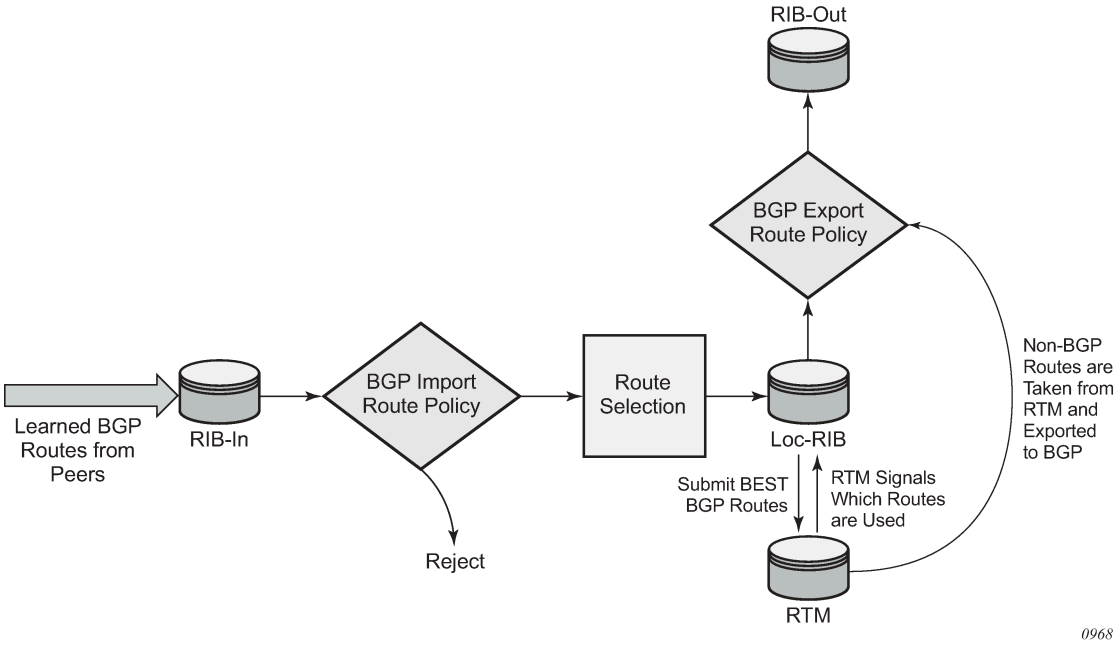
AS path to match criteria	Regular expression	Example matches
	[10-12]*	Null AS path 10 or 11 or 12 10 10 or 10 11 or 10 12 11 10 or 11 11 or 11 12 12 10 or 12 11 or 12 12 …
Zero or one occurrence of AS number 11	11? or 11{0,1}	Null AS path 11
One through four occurrences of AS number 11	11{1,4}	11 11 11 11 11 11 11 11 11 11
One through four occurrences of AS number 11 followed by one occurrence of AS number 22	11{1,4} 22	11 22 11 11 22 11 11 11 22 11 11 11 11 22
Path of any length, except nonexistent, whose second AS number can be anything, including nonexistent	. .* or . .{0,}	100 100 200 11 22 33 44 55
AS number is 100. Community value is 200.	^100:200$	100:200
AS number is 11 or 22. Community value is any number.	^((11)|(22)):(.*)$	11:100 22:100 11:200 …
AS number is 11. Community value is any number that starts with 1.	^11:(1.*)$	11:1 11:100 11:1100 …
AS number is any number. Community value is any number that ends with 1, 2, or 3.	^(.*):(.*[1-3])$	11:1 100:2002 333:55553 …

AS path to match criteria	Regular expression	Example matches
AS number is 11 or 22. Community value is any number that starts with 3 and ends with 4, 5 or 9.	^((11)|(22)):(3.*[459])$	11:34 22:3335 11:3777779 …
AS number is 11 or 22. Community value ends in 33 or 44.	[^((11|22)):(.*((33)|(44)))$	11:33 22:99944 22:555533 …
Range of Community values	100&^([1-9][0-9]|[1-9][0-9][0-9]|1[0-9][0-9][0-9]|2000)$"	100:10 100:11 100: … 100:2000

6.1.3 BGP and OSPF route policy support

OSPF and BGP requires route policy support. Figure 37: BGP route policy diagram and Figure 38: OSPF route policy diagram display where route policies are evaluated in the protocol. Figure 37: BGP route policy diagram depicts BGP which applies a route policy as an internal part of the BGP route selection process. Figure 38: OSPF route policy diagram depicts OSPF which applies routing policies at the edge of the protocol, to control only the routes that are announced to or accepted from the Route Table Manager (RTM).

Figure 37: BGP route policy diagram

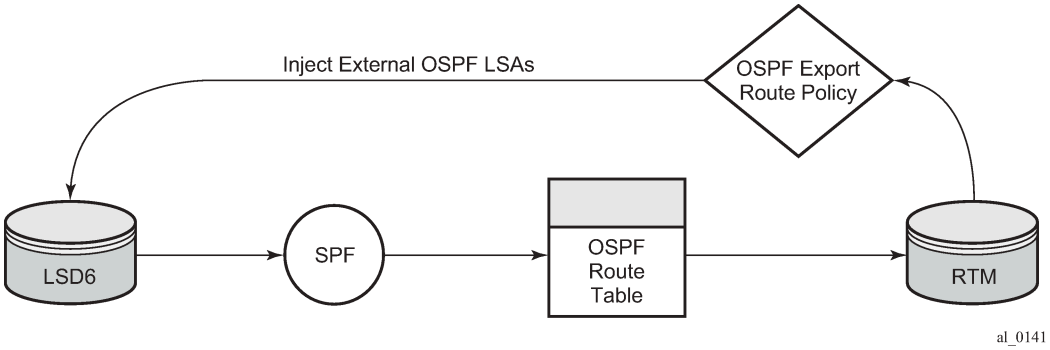


6.1.3.1 BGP route policies

Nokia's implementation of BGP uses route policies extensively. The implied or default route policies can be overridden by customized route policies. The default BGP properties, with no route policies configured, behave as follows:

- accept all BGP routes into the RTM for consideration
- announce all used BGP learned routes to other BGP peers
- announce none of the IGP, static or local routes to BGP peers

Figure 38: OSPF route policy diagram



6.1.3.2 Re-advertised route policies

Occasionally, BGP routes may be re advertised from BGP into OSPF, IS-IS, and RIP. OSPF export policies (policies control which routes are exported to OSPF) are not handled by the main OSPF task but are handled by a separate task or an RTM task that filters the routes before they are presented to the main OSPF task.

6.1.3.3 Triggered policies

With triggered policy enabled, deletion and re-addition of a peer after making changes to export policy causes the new updates sent out to all peers.

Triggered policy is not honored if a new peer added to BGP. Update with the old policy is sent to the newly added peer. New policy does not get applied to the new peer until the peer is flapped.

With triggered policy enabled, if a new BGP/static route comes in, new addition or modification of an export policy causes the updates to send out dynamically to all peers with the new/modified export policy.

When multiple peers, say P1, P2 and P3 share the same export policy, any modifications to export policy followed by clear soft on one of the peer P1, send out routes to P1 only according to newly modified policy.

Though routes with newly modified policy are not sent to other peers (P2, and P3) as no clear soft issues on these peers, RIB-OUT shows that new routes with modified policy are sent to all the peers. RIB-IN on peers P2 and P3 are shown correctly.

Note: With the triggered policy enabled, deletion and re-addition of a peer after making changes to export policy causes the new updates sent out to all peers.

The triggered policy is not honored if a new peer is added to BGP. An update with the old policy is sent to the newly added peer. The new policy is not applied to the new peer until the peer is flapped.

With the triggered policy enabled, if a new BGP/static route comes in, the new addition or modification of an export policy causes the updates to be sent out dynamically to all peers with the new/modified export policy.

When multiple peers, such as P1, P2 and P3, share the same export policy, any modifications to the export policy followed by **clear soft** on one of the peer P1s, send out routes to P1 only according to the newly modified policy. Though routes with the newly modified policy are not sent to other peers (P2 and P3) as there are no clear soft issues on these peers, RIB-OUT shows that the new routes with the modified policy are sent to all the peers. RIB-IN on peers P2 and P3 are shown correctly.

6.1.3.4 Set MED to IGP cost using route policies

This feature sets MED to the IGP cost of a route exported into BGP as an action in route policies. The **med-out** command in the bgp, group, and neighbor configuration context supports this option, but this method lacks per-prefix granularity. The enhanced **metric** command supported as a route policy action supports setting MED to a fixed number, or adding, or subtracting a fixed number from the received MED, and sets IGP cost option. The enhanced **metric** {**set** {**igp** | **number1**} | {**add** | **subtract**} **number2**} command is under **config**>**router**>**policy-options**>**policy-statement**>**entry**>**action**.

The **metric set igp** command, when used in a BGP export policy, have the same effect as the current **med-out igp** command, except that it applies only to the routes matched by the policy entry.

The effect of the **metric set igp** command depends on the route type and policy type as summarized in Table 19: Metric set IGP effect.

Table 19: Metric set IGP effect

BGP policy type	Matched route type	Set metric IGP effect
Export	Non-BGP route (static, OSPF, ISIS, and so on)	Add MED attribute. Set value to M.
Export	BGP route w/o MED	Add MED attribute. Set value to D.
Export	BGP route with MED (value A)	Overwrite MED attribute with value D.

6.1.3.5 BGP policy subroutines

A BGP policy can call another policy (subroutine) and that subroutine can call another subroutine, and so on. This facilitates re-usability of common logic and a structured approach to writing BGP policies. Up to three levels of subroutine are supported.

6.1.3.6 Route policies for BGP next-hop resolution and peer tracking

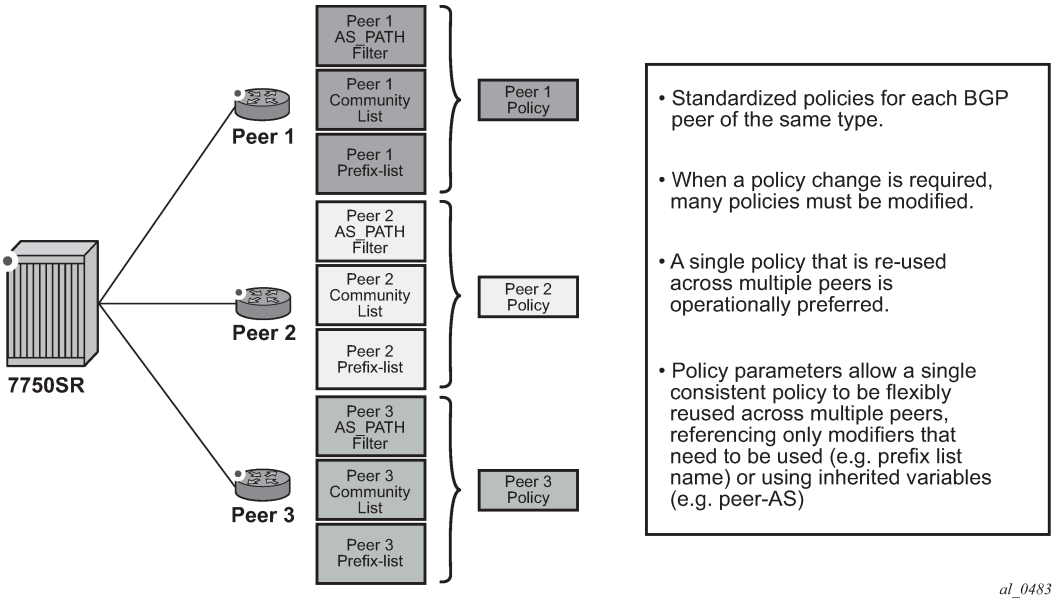
This feature adds the flexibility to attach a route policy to the BGP next-hop resolution process; it also allows a route policy to be associated with the optional BGP peer-tracking function. BGP next-hop resolution is a fundamental part of BGP protocol operation; it determines the best matching route (or tunnel) for the BGP next-hop address and uses information about this resolving route in the best path selection algorithm and to program the forwarding table. Attaching a policy to BGP next-hop resolution provides more control over which IP routes in the routing table can become resolving routes. Similar flexibility is also available for BGP peer-tracking, which is an optional feature that allows the session with a BGP neighbor to be taken down if there is no IP route to the neighbor address or if the best matching IP route is rejected by the policy.

6.1.4 Routing policy parameterization

Routing policy parameterization provides users with a powerful and flexible configuration approach for policies that are often reused across BGP peers of a common type (such as transit, peer, customer, and so on).

Without routing policy parameterization, the user must create individual routing policies, prefix-lists, AS-Path lists, community lists, and so on for each peer. This is despite the policy ultimately being the same in many cases, as shown in the following figure. For example, if there are 100 peers with a common policy behavior but unique policies, and the user needs to update entry 135 in the policy, this change must be made on all policies. This is a significant amount of work that can result in incorrect or inconsistent policy behavior.

Figure 39: Route policy mode of operation without parameterization



Using a parameter-based system allows an operator to have a single policy that is consistent across all peers of a type, while retaining the flexibility to reference different policy functions (such as, prefixes, prefix-lists, community lists) with unique names if required, by defining variables and the variable value.

Additionally, instead of fixed policies that require many statements, parameters and variables may be passed to simplify policy configuration. This reduces the number of policies required on a peering edge router with a large numbers of peers that have only minor configuration changes between the peers, such as the ASN and prefix-list name.

The following types of policy variables are supported.

- **global**
  Use global policy variables for variables used by multiple policies. Global policy variables are configured using the following command and can be referenced from any policy:
  - **MD-CLI**

    ```
    configure policy-options global-variables
    ```

  - **classic CLI**

    ```
    configure router policy-options global-variables
    ```

- **local**
  Use local policy variables when the variables are specific to a policy and are not used in other policies. Local policy variables are configured using the following command in each policy, and can be referenced in a sub-policy:

– **MD-CLI**

```
configure policy-options policy-statement entry from policy-variables
```

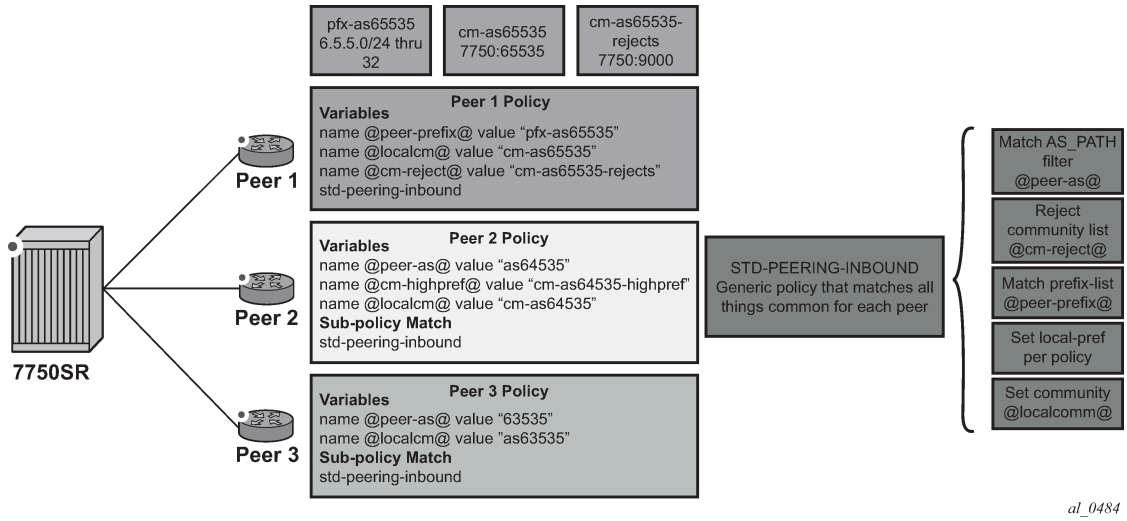
– **classic CLI**

```
configure router policy-options policy-statement entry from policy-
variables
```

Note: If policy variables have the same name, local policy variables have precedence over global policy variables.

The following figure shows an example of route policy parameterization using sub-policies.

Figure 40: Route policy parameterization using sub-policies



Variables expansion can use two formats, with the variable name delimited by at-signs (@) at the beginning and the end:

- standard variables must start and end with at-signs (@), for example: @variable@
- midstring variables must be enclosed with at-signs (@) and may be midstring, for example: @variable@, start@variable@end, @variable@end, start@variable@

Note: Midstring variables are only supported in the object name. Standard variables are supported in policy parameters inside as-path or as-path-group expression objects; for example: as-path "as" expression ".{65001,@variable@}65022".

The following table lists route policy variables supported in policy parameters.

Table 20: Route policy variable support in policy parameters

Parameter name	Variable in policy "from" statement	Variable in policy "action" statement	Variable format	Standard format release	Midstring format release
aigp-metric	No	Yes	Standard	13.0.R4	—
as-path	Yes	Yes	Midstring	12.0.R1	13.0.R1
as-path expression	Yes	No	Midstring	12.0.R4	13.0.R1
as-path-group	Yes	No	Midstring	12.0.R1	13.0.R1
as-path-group expression	Yes	No	Midstring	12.0.R4	13.0.R1
as-path-length	Yes	—	Standard	15.0.R1	—
as-path-prepend	—	Yes	Standard	13.0.R4	—
cluster-id	No	—	—	—	—
community	Yes	Yes	Midstring	12.0.R1	13.0.R1
community-count	Yes	—	Standard	15.0.R1	N/A
damping	No	Yes	Midstring	13.0.R4	13.0.R4
local-preference	Yes (15.0.R1)	Yes (13.0.R4)	Standard	13.0.R4	—
metric	Yes (15.0.R1)	Yes (13.0.R4)	Standard	13.0.R4	—
next-hop	No	Yes	Standard	13.0.R4	—
origin	No	Yes	Standard	13.0.R4	—
path-type	No	—	—	—	—
preference	No	Yes	Standard	13.0.R4	—
prefix-list	Yes	No	Midstring	12.0.R1	13.0.R1
route-distinguisher-list	Yes	No	Midstring	23.3.R1	23.3.R1
tag	No	Yes	Standard	13.0.R4	—
type	No	Yes	Standard	13.0.R4	—

For the definition of the variables, there are three different possible types:

- **name** *name-string* **value** *value-string*
- **name** *name-string* **address** *ip-address*

- **name** *name-string* **number** *value-number*

Depending on the parameter referenced, specify the correct type as follows:

- *value-string*: **as-path**, **as-path-group**, **community**, **prefix-list**, **damping**
- *ip-address*: **next-hop**
- *value-number*: **aigp-metric**, **as-path-length, as-path-prepend**, **community-count, local-preference, metric, origin**, **origin-validation**, **preference**, **tag**, **type**

The logical flow of this is to configure a policy per peer, in which the variable names and values are defined. Using Figure 40: Route policy parameterization using sub-policies as the example, the following configuration is applied.

Example: Routing policy parameterization using sub-policies (MD-CLI)

```
[ex:/configure policy-options]
A:admin@node-2# info
    as-path "as63535" {
        expression "^63535$"
    }
    as-path "as64535" {
        expression "^64535$"
    }
    as-path "as65535" {
        expression "^65535$"
    }
    community "cm-as63535" {
        member "7750:63535" { }
    }
    community "cm-as64535-highpref" {
        member "7777:64535" { }
    }
    community "cm-as64535-rejects" {
        member "64535:14" { }
    }
    community "cm-as65535" {
        member "7750:65535" { }
    }
    community "cm-as65535-rejects" {
        member "65535:14" { }
    }
    prefix-list "pfx-as63535" {
        prefix 6.3.5.0/24 type through {
            through-length 32
        }
    }
    prefix-list "pfx-as64535" {
        prefix 6.4.5.0/24 type through {
            through-length 32
        }
    }
    prefix-list "pfx-as65535" {
        prefix 6.5.5.0/24 type through {
            through-length 32
        }
    }
    policy-statement "peer1" {
        entry 5 {
            from {
                policy "std-peering-inbound-drop"
                policy-variables {
                    name "@cm-reject@" {
```

```
                        value "cm-as65535-rejects"
                    }
                }
            }
            action {
                action-type reject
            }
        }
        entry 10 {
            from {
                policy "std-peering-inbound-main"
                policy-variables {
                    name "@cm-highpref@" {
                        value "cm-as65535-highpref"
                    }
                    name "@localcm@" {
                        value "cm-as65535"
                    }
                    name "@peer-as@" {
                        value "as65535"
                    }
                    name "@peer-prefix@" {
                        value "pfx-as65535"
                    }
                }
            }
            action {
                action-type accept
            }
        }
    }
    policy-statement "peer2" {
        entry 5 {
            from {
                policy "std-peering-inbound-drop"
                policy-variables {
                    name "@cm-reject@" {
                        value "cm-as64535-rejects"
                    }
                }
            }
            action {
                action-type reject
            }
        }
        entry 10 {
            from {
                policy "std-peering-inbound-main"
                policy-variables {
                    name "@cm-highpref@" {
                        value "cm-as64535-highpref"
                    }
                    name "@localcm@" {
                        value "cm-as64535"
                    }
                    name "@peer-as@" {
                        value "as64535"
                    }
                    name "@peer-prefix@" {
                        value "pfx-as64535"
                    }
                }
            }
            action {
```

```
                action-type accept
            }
        }
    }
    policy-statement "peer3" {
        entry 5 {
            from {
                policy "std-peering-inbound-drop"
                policy-variables {
                    name "@cm-reject@" {
                        value "cm-as63535-rejects"
                    }
                }
            }
            action {
                action-type reject
            }
        }
        entry 10 {
            from {
                policy "std-peering-inbound-main"
                policy-variables {
                    name "@cm-highpref@" {
                        value "cm-as63535-highpref"
                    }
                    name "@localcm@" {
                        value "cm-as635325"
                    }
                    name "@peer-as@" {
                        value "as63535"
                    }
                    name "@peer-prefix@" {
                        value "pfx-as63535"
                    }
                }
            }
            action {
                action-type accept
            }
        }
    }
    policy-statement "std-peering-inbound-drop" {
        entry 10 {
            from {
                community {
                    name "@cm-reject@"
                }
            }
            action {
                action-type accept
            }
        }
        default-action {
            action-type reject
        }
    }
    policy-statement "std-peering-inbound-main" {
        entry 10 {
            from {
                prefix-list ["@peer-prefix@"]
                as-path {
                    name "@peer-as@"
                }
            }
```

```
            action {
                action-type accept
                local-preference 400
                community {
                    add ["@localcm@"]
                }
            }
        }
        entry 20 {
            from {
                community {
                    name "@cm-highpref@"
                }
            }
            action {
                action-type accept
                local-preference 4000
                community {
                    add ["@localcm@"]
                }
            }
        }
        default-action {
            action-type reject
        }
    }
```

Example: Routing policy parameterization using sub-policies (classic CLI)

```
A:node-2>config>router>policy-options# info
----------------------------------------------
prefix-list "pfx-as63535"
    prefix 6.3.5.0/24 through 32
exit
prefix-list "pfx-as64535"
    prefix 6.4.5.0/24 through 32
exit
prefix-list "pfx-as65535"
    prefix 6.5.5.0/24 through 32
exit
community "cm-as63535"
    members "7750:63535
exit
community "cm-as65535"
    members "7750:65535"
exit
community "cm-as64535-rejects"
    members "64535:14"
exit
community "cm-as-65535-rejects"
    members "65535:14"
exit
community "cm-as64535-highpref"
    members "7777:64535"
exit
as-path "as63535"
    expression "^63535$"
exit
as-path "as64535"
    expression "^64535$"
exit
as-path "as65535"
    expression "^65535$"
```

```
exit
policy-statement "peer1"
    entry 5
        from
            policy-variables
                name "@cm-reject@" value "cm-as65535-rejects"
            exit
            policy "std-peering-inbound-drop"
        exit
        action reject
    entry 10
        from
            policy-variables
                name "@localcm@" value "cm-as65535"
                name "@peer-as@" value "as65535"
                name "@cm-highpref@" value "cm-as65535-highpref"
                name "@peer-prefix@" value "pfx-as65535"
            exit
            policy "std-peering-inbound-main"
        exit
        action accept
        exit
    exit
exit
policy-statement "peer2"
    entry 5
        from
            policy-variables
                name "@cm-reject@" value "cm-as64535-rejects"
            exit
            policy "std-peering-inbound-drop"
        exit
        action reject
    entry 10
        from
            policy-variables
                name "@localcm@" value "cm-as64535"
                name "@peer-as@" value "as64535"
                name "@cm-highpref@" value "cm-as64535-highpref"
                name "@peer-prefix@" value "pfx-as64535"
            exit
            policy "std-peering-inbound-main"
        exit
        action accept
        exit
    exit
exit
policy-statement "peer3"
    entry 5
        from
            policy-variables
                name "@cm-reject@" value "cm-as63535-rejects"
            exit
            policy "std-peering-inbound-drop"
        exit
        action reject
    entry 10
        from
            policy-variables
                name "@localcm@" value "cm-as63535"
                name "@peer-as@" value "as63535"
                name "@cm-highpref@" value "cm-as63535-highpref"
                name "@peer-prefix@" value "pfx-as63535"
            exit
```

```
            policy "std-peering-inbound-main"
        exit
        action accept
        exit
    exit
exit
policy-statement "std-peering-inbound-drop"
    default-action reject
    entry 10
        from
            community "@cm-reject@"
        exit
        action accept
    exit
policy-statement "std-peering-inbound-main"
    default-action reject
    entry 10
        from
            prefix-list "@peer-prefix@"
            as-path "@peer-as@"
        exit
        action accept
            community add "@localcm@"
            local-preference 400
        exit
    exit
    entry 20
        from
            community "@cm-highpref@"
        exit
        action accept
            community add "@localcm@"
            local-preference 4000
        exit
    exit
exit
```

This configuration would take slightly different actions depending on the peer.

Peer 1

- Routes that have a community matching "cm-as65535-rejects" are dropped.
- Routes matching the prefix list "pfx-as65535" that originated in the peer AS=65535 are accepted with a local preference of 400.
- Community "7750:65535" is added to accepted prefixes.
- As community-list "cm-65535-highpref" does not exist, no routes are modified with a local preference of 4000.

Peer 2

- Routes that have a community matching "cm-as64535-rejects" are dropped.
- Routes matching the prefix list "pfx-as65535" that originated in the peer AS=65535 are accepted with a local preference of 400.
- Prefixes matching "cm-as64535-highpref" are set to a local-preference of 4000.

Peer 3

- As community-list "cm-as63535-rejects" does not exist, no routes are dropped by the first entry.

- Routes matching the prefix list "pfx-as63535" that originated in the peer AS=63535 are accepted with a local preference of 400.
- Community "7750:63535" is added to accepted prefixes.
- As community-list "cm-63535-highpref" does not exist, no routes are modified with a local preference of 4000.

6.1.5 When to use route policies

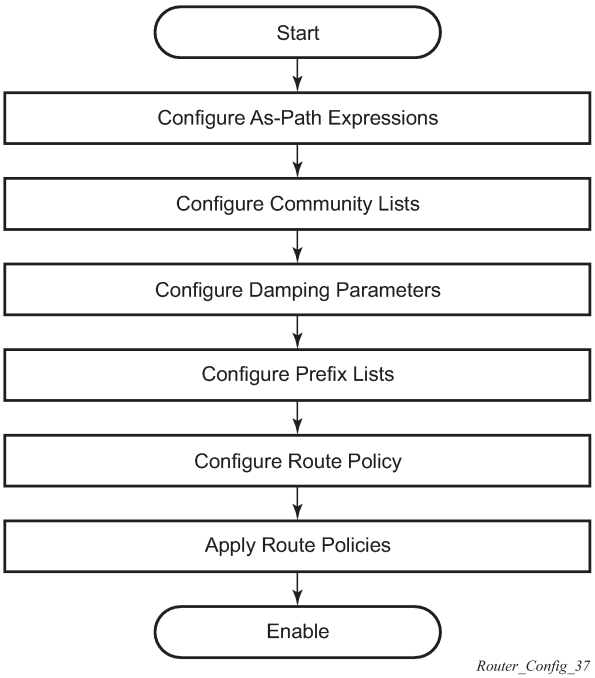
The following are examples of circumstances of when to configure and apply unique route policies.

- When you want to control the protocol to allow all routes to be imported into the routing table. This enables the routing table to learn about particular routes to enable packet forwarding and redistributing packets into other routing protocols.
- When you want to control the exporting of a protocol's learned active routes.
- When you want a routing protocol to announce active routes learned from another routing protocol, which is sometimes called route redistribution.
- When you want unique behaviors to control route characteristics. For example, change the route preference.
- When you want unique behaviors to control route characteristics. For example, change the route preference, AS path, or community values to manipulate the control the route selection.
- When you want to control BGP route flapping (damping).

6.2 Route policy configuration process overview

Figure 41: Route policy configuration and implementation flow displays the process to provision basic route policy parameters.

Figure 41: Route policy configuration and implementation flow



6.3 Configuration notes

This section describes route policy configuration restrictions.

6.3.1 General

When configuring policy statements, the policy statement name must be unique.

6.3.2 Policy reference checks

The policy reference checks functionality can be enabled with **config**>**router**>**policy-reference-checks** to indicate policies and policy objects in policy statements that are referenced but do not exist. An INFO or WARNING message is displayed in the CLI.

Note: The policy reference checks feature is available only in the classic CLI and not via SNMP or model-driven interfaces. Model-driven interfaces and **mixed** management interface configuration mode use strict policy reference checks starting in Release 16.0.R3, where the checks cannot be disabled.

If a policy is referenced which has not yet been configured, and **no policy-reference-checks** is set, the configuration line succeeds with the CLI message:

```
WARNING: CLI Policy "bar" does not exist
```

If a policy is referenced which has not yet been configured, and **policy-reference-checks** is set, the configuration line errors and fails with the CLI message:

```
MINOR: CLI Policy "foo" does not exist
```

Policy reference checks are required in router configuration contexts that reference a policy name, including the following:

- **configure router bgp export**
- **configure router bgp group export**
- **configure router bgp group import**
- **configure router bgp group neighbor export**
- **configure router bgp group neighbor import**
- **configure router bgp import**
- **configure service vprn bgp export**
- **configure service vprn bgp group export**
- **configure service vprn bgp group import**
- **configure service vprn bgp group neighbor export**
- **configure service vprn bgp group neighbor import**
- **configure service vprn bgp import**
- **configure service vprn vrf-export**
- **configure service vprn vrf-import**

Policy reference checks are required in policy configuration contexts that reference a policy element, including the following:

- **entry from flow-spec-dest** *prefix-list-name*
- **entry from flow-spec-source** *prefix-list-name*
- **entry from group-address** *prefix-list-name*
- **entry from host-ip** *prefix-list-name*
- **entry from neighbor** {*ip-address* | *prefix-list-name*}
- **entry from next-hop** *prefix-list-name*
- **entry from prefix-list** *name* [*name*]
- **entry from source-address prefix-list** *prefix-list-name*
- **entry to neighbor** {*ip-address* | *prefix-list-name*}
- **entry to prefix-list** *name* [*name*]
- **as-path**
- **as-path-group**

- **community**
- **damping**
- policy statements used in sub-policies

6.3.2.1 Known limitations

Policy elements must be entered in the correct order to resolve references. For example, if a prefix-list is referenced in a policy statement, the prefix-list must be entered first. This feature does not order policy statements so that references to policy elements exist.

Variable expansion is not supported with the policy reference checks feature. Variables that reference policy elements that do not exist are not checked and are permitted.

This feature is available only for configuring policies. If a policy is deleted, the command succeeds with no info or warning messages displayed. Information from the **show router policy-edits** command can be used to see warnings about which elements do not exist.

The policy reference checks functionality is not supported with policy expressions.

6.4 Configuring route policies with CLI

This section provides information to configure route policies.

6.4.1 Route policy configuration overview

Route policies allow you to configure routing according to specifically defined policies. You can create policies and entries to allow or deny paths based on various parameters such as destination address, protocol, packet size, and community list.

Policies can be as simple or complex as required. A simple policy can block routes for a specific location or IP address. More complex policies can be configured using numerous policy statement entries containing matching conditions to specify whether to accept or reject the route, control how a series of policies are evaluated, and manipulate the characteristics associated with a route.

6.4.1.1 When to create routing policies

Route policies are created in the **config**>**router** context. There are no default route policies. Each route policy must be explicitly created and applied. Applying route policies can introduce more efficiency as well as more complexity to routers.

A route policy impacts the flow of routing information or packets within and through the router. A routing policy can be specified to prevent a particular customer’s routes to be placed in the route table which causes those routes to not forward traffic to various destinations and the routes are not advertised by the routing protocol to neighbors.

Route policies can be created to control:

- a protocol to export all the active routes learned by that protocol

- route characteristics to control which route is selected to act as the active route to reach a destination and advertise the route to neighbors
- protocol to import all routes into the routing table. A routing table must learn about particular routes to be able to forward packets and redistribute to other routing protocols
- damping

Before a route policy is applied, analyze the policy's purpose and be aware of the results (and consequences) when packets match the specified criteria and the associated actions and default actions, if specified, are executed. Membership reports can be filtered based on a specific source address.

6.4.1.2 Default route policy actions

Each routing protocol has default behaviors for the import and export of routing information. Table 21: Default route policy actions shows the default behavior for each routing protocol.

Table 21: Default route policy actions

Protocol	Import	Export
OSPF	Not applicable. All OSPF routes are accepted from OSPF neighbors and cannot be controlled via route policies.	• Internal routes: All OSPF routes are automatically advertised to all neighbors. • External routes: By default, all non-OSPF learned routes are not advertised to OSPF neighbors.
IS-IS	Not applicable. All IS-IS routes are accepted from IS-IS neighbors and cannot be controlled via route policies	• Internal routes: All IS-IS routes are automatically advertised to all neighbors. • External routes: By default, all non-IS-IS learned routes are not advertised to IS-IS peers.
RIP	By default, all RIP-learned routes are accepted.	External routes: By default, all non-RIP learned routes are not advertised to RIP peers.
BGP	By default, all routes from BGP peers are accepted and passed to the BGP route selection process.	• Internal routes: By default, all active BGP routes are advertised to BGP peers. • External routes: By default, all non-BGP learned routes are not advertised to BGP peers.

6.4.1.3 Policy evaluation

Routing policy statements can consist of as few as one or several entries. The entries specify the matching criteria. A route is compared to the first entry in the policy statement. If it matches, the specified entry

action is taken, either accepted or rejected. If the action is to accept or reject the route, that action is taken and the evaluation of the route ends.

If the route does not match the first entry, the route is compared to the next entry (if more than one is configured) in the policy statement. If there is a match with the second entry, the specified action is taken. If the action is to accept or reject the route, that action is taken and the evaluation of the route ends, and so on.

Each route policy statement can have a default-action clause defined. If a default-action is defined for one or more of the configured route policies, then the default actions should be handled in the following ways.

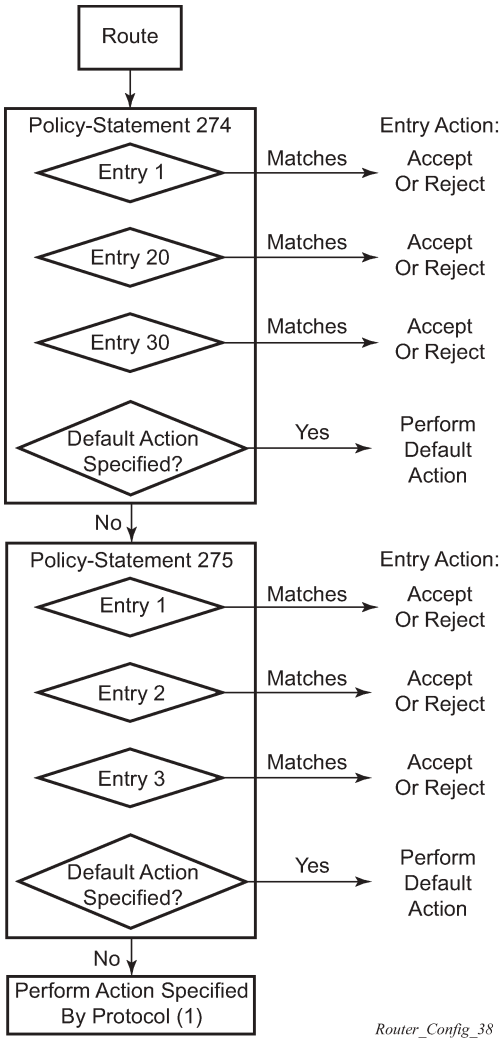
- The process stops when the first complete match is found and executes the action defined in the entry.
- If the packet does not match any of the entries, the system executes the default action specified in the policy statement.

Figure 42: Route policy process example depicts an example of the route policy process.

Route policies can also match a specific route policy entry and continue to search for other entries within either the same route policy or the next route policy by specifying the *next-entry* or *next-policy* option in the entry's **action** command. Policies can be constructed to support multiple states to the evaluation and setting of various route attributes.

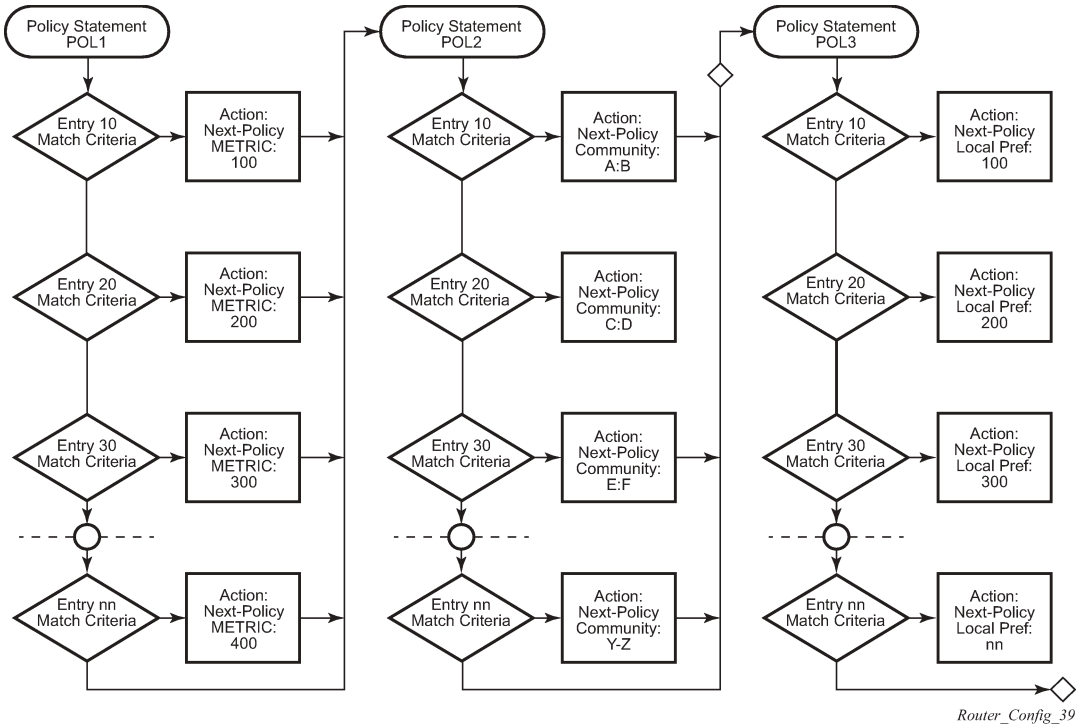
Figure 43: Next policy logic example depicts the next-policy and next-entry route processes.

Figure 42: Route policy process example



For the default route policy actions, see Table 21: Default route policy actions.

Figure 43: Next policy logic example



6.4.1.4 Damping

Damping initiates controls when routes flap. Route flapping can occur when an advertised route between nodes alternates (flaps) back and forth between two paths because of network problems which cause intermittent route failures. It is necessary to reduce the amount of routing state change updates propagated to limit processing requirements. Thus, when a route flaps beyond a configured value (the suppress value), then that route is removed from the routing tables and routing protocols until the value falls below the reuse value.

A route can be suppressed according to the Figure of Merit (FoM) value. The FoM is a value that is added to a route each time it flaps. A new route begins with an FoM value of 0.

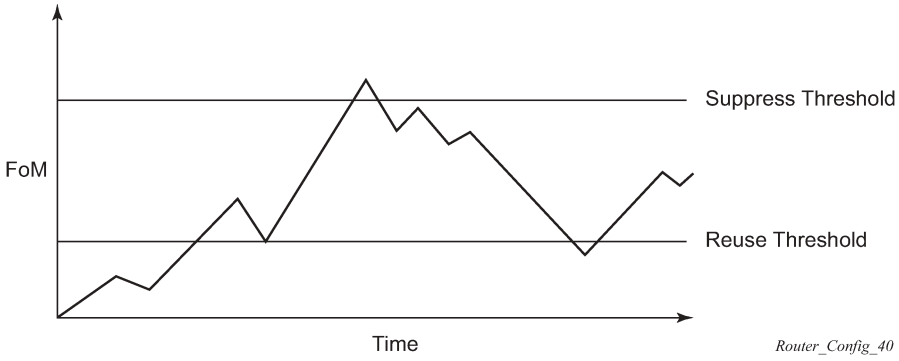
Damping is optional. If damping is configured, the following parameter values must be explicitly specified as there are no default values:

- suppress
- half-life
- reuse
- max-suppress

When a route's FoM value exceeds the suppress value, then the route is removed from the routing table. The route is considered to be stable when the FoM drops below the reuse value by means of the specified half-life parameter. The route is returned to the routing tables. When routes have higher FoM and half-life values, they are suppressed for longer periods of time. Figure 44: Damping example depicts an example

of a flapping route, the suppress threshold, the half-life decay (time), and reuse threshold. The peaks represent route flaps, the slopes represent half-life decay.

Figure 44: Damping example



6.4.2 Basic configurations

This section provides information to configure route policies and configuration examples of common tasks. The minimal route policy parameters that need to be configured are described below.

Policy statement with the following parameters specified:

- at least one entry
- entry action

Following is an example route policy configuration:

```
A:ALA-B>config>router>policy-options# info
----------------------------------------------
            community "all-types" members "5000:[1-6][1-9][0-9]"
            community "all-normal" members "5000:[1-5][1-9][0-9]"
. . .
as-path "Outside madeup paths" ".* 5001 .*"
            as-path "Outside Internet paths" ".* 5002 .*"
            policy-statement "RejectOutsideASPaths"
                entry 1
                    from
                        protocol bgpospf
                        as-path "Outside madeup paths"
                    exit
                    action reject
                    exit
                exit
                entry 2
                    from
                        protocol bgpospf
                        as-path "Outside Internet paths"
                    exit
                    action reject
                    exit
                exit
                entry 3
                    from
                        protocol ospf
                    exit
```

```
                    to
                        protocol bgpospf
                    exit
                    action reject
                    exit
                exit
                entry 4
                    from
                        protocol isis
                    exit
                    to
                        protocol bgpospf
                    exit
                    action reject
                    exit
                exit
                default-action accept
                exit
            exit
            policy-statement "aggregate-customer-peer-only"
                entry 1
                    from
                        community "all-customer-announce"
                    exit
                    action accept
                    exit
                exit
                default-action reject
                exit
            exit
-------------------------------------------
A:ALA-B>config>router>policy-options#
```

6.4.3 Configuring route policy components

This section describes information to configure route policy components using the Classic CLI engine.

Note: The MD-CLI has significant differences from the classic CLI for configuring route policy components. In the MD-CLI, there is no separate transaction (**begin** or **commit**) for the policy changes. Also, the MD-CLI supports policy statements with entry type **named** (see Policy statements for information about policy statements).

6.4.3.1 Beginning the policy statement

Use the following CLI syntax to begin a policy statement configuration. In order for a policy statement to be complete an entry must be specified (see Configuring an entry).

```
config>router>policy-options
  begin
    policy-statement name
      description text
```

The following error message displays when you try to modify a policy options command without entering **begin** first.

```
A:ALA-B>config>router>policy-options# policy-statement "allow all"
MINOR: CLI The policy-options must be in edit mode by calling begin before
```

```
any changes can be made.
```

The following example displays policy statement configuration command usage. These commands are configured in the **config**>**router** context.

Example:

```
config>router# policy-options
policy-options# begin
```

There are no default policy statement options. All parameters must be explicitly configured.

6.4.3.2 Creating a route policy

To enter the mode to create or edit route policies, you must enter the **begin** keyword at the **config**>**router**>**policy-options** prompt. Other editing commands include:

- the **commit** command saves changes made to route policies during a session
- the **abort** command discards changes that have been made to route policies during a session

The following error message displays when you try to modify a policy options command without entering **begin** first.

```
A:ALA-B>config>router>policy-options# policy-statement “allow all”
MINOR: CLI The policy-options must be in edit mode by calling begin before any
changes can be made.

A:ALA-B>config>router>policy-options# info
#------------------------------------------
# Policy
#------------------------------------------

        policy-options
            begin
            policy-statement "allow all"
description "General Policy"
...
            exit
exit
----------------------------------------------
A:ALA-B>config>router>policy-options#
```

6.4.3.3 Configuring a default action

Specifying a default action is optional. The default action controls those packets not matching any policy statement entries. If no default action is specified for the policy, then the action associated with the protocol to which the routing policy was applied is performed. The default action is applied only to those routes that do not match any policy entries.

A policy statement must include at least one entry (see Configuring an entry).

To enter the mode to create or edit route policies, you must enter the begin keyword at the **config**>**router**>**policy-options** prompt. Other editing commands include:

- the **commit** command saves changes made to route policies during a session
- the **abort** command discards changes that have been made to route policies during a session

The following example displays the default action configuration:

```
A:ALA-B>config>router>policy-options# info
----------------------------------------------
            policy-statement "1"
                default-action accept
                    as-path add "test"
                    community add "365"
                    damping "flaptest"
                    next-hop 10.10.10.104
                exit
            exit
----------------------------------------------
A:ALA-B>config>router>policy-options#
```

6.4.3.4 Configuring an entry

An entry action must be specified. The other parameters in the **entry action** context are optional.

The following example displays entry parameters and includes the default action parameters which were displayed in the previous section.

```
A:ALA-B>config>router>policy-options# info
----------------------------------------------
            policy-statement "1"
                entry 1
                    to
neighbor 10.10.10.104
                    exit
                    action accept
                    exit
                exit
                entry 2
                    from
                        protocol ospf 1
                    exit
                    to
                        protocol ospf
                        neighbor 10.10.0.91
                    exit
                    action accept
                    exit
                exit
                default-action accept
                    . . .
                exit
            exit
```

The following example displays entry parameters and includes the default action parameters which were displayed in the previous section.

```
A:ALA-B>config>router>policy-options# info
----------------------------------------------
            policy-statement "1"
                entry 1
                    to
                        protocol bgp
                        neighbor 10.10.10.104
                    exit
                    action accept
```

```
                exit
            exit
            entry 2
                from
                    protocol ospf 1
                exit
                to
                    protocol ospf
                    neighbor 10.10.0.91
                exit
                action accept
                exit
            exit
            default-action accept
                . . .
            exit
        exit
```

6.4.3.5 Configuring a community list

Community lists are composed of a group of destinations which share a common property. Community lists allow you to administer actions on a configured group instead of having to execute identical commands for each member.

The following example displays a community list configuration:

```
A:ALA-B>config>router>policy-options# info
----------------------------------------------
community "eastern" members "100:200"
community "western" members "100:300"
community "northern" members "100:400"
community "southern" members "100:500"
community "headquarters" members "100:1000"
policy-statement "1"
            entry 1
                to
                    protocol bgp
                    neighbor 10.10.10.104
                exit
                action accept
. . .
----------------------------------------------
A:ALA-B>config>router>policy-options#
```

6.4.3.6 Configuring damping

The following considerations apply.

- For each damping profile, all parameters must be configured.
- The **suppress** value must be greater than the reuse value (see Figure 44: Damping example).
- Damping can be enabled in the **config**>**router**>**bgp** context on the BGP global, group, and neighbor levels. If damping is enabled, but route policy does not specify a damping profile, the default damping profile is used. This profile is always present and consists of the following parameters:
  - half-life: 15 minutes
  - max-suppress: 60 minutes

– suppress: 3000
– reuse: 750

The following example displays a damping configuration:

```
*A:cses-A13>config>router>policy-options# info
----------------------------------------------
            damping "damptest123"
                half-life 15
                max-suppress 60
                reuse 750
                suppress 1000
            exit
----------------------------------------------
*A:cses-A13>config>router>policy-options#
```

6.4.3.7 Configuring a prefix list

The following example displays a prefix list configuration:

```
A:ALA-B>config>router>policy-options# info
----------------------------------------------
            prefix-list "western"
                    prefix 10.10.0.1/32 exact
                    prefix 10.10.0.2/32 exact
                    prefix 10.10.0.3/32 exact
                    prefix 10.10.0.4/32 exact
            exit
            damping "damptest123"
                half-life 15
                max-suppress 60
                reuse 750
            exit
----------------------------------------------
A:ALA-B>config>router>policy-options#
```

6.5 Route policy configuration management tasks

This section discusses route policy configuration management tasks.

6.5.1 Editing policy statements and parameters

Route policy statements can be edited to modify, add, or delete parameters. To enter the mode to edit route policies, you must enter the begin keyword at the config>router> policy-options prompt. Other editing commands include:

- the **commit** command saves changes made to route policies during a session
- the **abort** command discards changes that have been made to route policies during a session

The following example displays a changed configuration:

```
A:ALA-B>config>router>policy-options>policy-statement# info
```

```
-----------------------------------------------
                description "Level 1"
                entry 1
                    to
                        protocol bgp
                        neighbor 10.10.10.104
                    exit
                    action accept
                    exit
                exit
                entry 2
                    from
                        protocol ospf
                    exit
                    to
                        protocol ospf
                        neighbor 10.10.0.91
                    exit
                    action accept
                    exit
                exit
                entry 4
                    description "new entry"
                    from
                        protocol isis
                        area 0.0.0.20
                    exit
                    action reject
                exit
                default-action accept
                    as-path add "test"
                    community add "365"
                    damping "flapper"
                    next-hop 10.10.10.104
exit
-----------------------------------------------
```

6.5.2 Deleting an entry

Use the following CLI syntax to delete a policy statement entry.

CLI syntax

```
config>router>policy-options
   begin
   commit
   abort
   policy-statement name
      no entry entry-id
```

The following example displays the commands required to delete a policy statement entry.

Example

```
config>router>policy-options# begin
policy-options# policy-statement "1"
policy-options>policy-statement# no entry 4
policy-options>policy-statement# commit
```

6.5.3 Deleting a policy statement

Use the following CLI syntax to delete a policy statement.

CLI syntax

```
config>router>policy-options
   begin
   commit
   abort
   no policy-statement name
```

The following example displays the commands required to delete a policy statement.

Example

```
config>router>policy-options# begin
policy-options# no policy-statement 1
policy-options# commit
```

7 Standards and protocol support

Note:
The information provided in this chapter is subject to change without notice and may not apply to all platforms.

Nokia assumes no responsibility for inaccuracies.

7.1 Access Node Control Protocol (ANCP)

draft-ietf-ancp-protocol-02, *Protocol for Access Node Control Mechanism in Broadband Networks*

RFC 5851, *Framework and Requirements for an Access Node Control Mechanism in Broadband Multi-Service Networks*

7.2 Bidirectional Forwarding Detection (BFD)

draft-ietf-lsr-ospf-bfd-strict-mode-10, *OSPF BFD Strict-Mode*

RFC 5880, *Bidirectional Forwarding Detection (BFD)*

RFC 5881, *Bidirectional Forwarding Detection (BFD) IPv4 and IPv6 (Single Hop)*

RFC 5882, *Generic Application of Bidirectional Forwarding Detection (BFD)*

RFC 5883, *Bidirectional Forwarding Detection (BFD) for Multihop Paths*

RFC 7130, *Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Interfaces*

RFC 7880, *Seamless Bidirectional Forwarding Detection (S-BFD)*

RFC 7881, *Seamless Bidirectional Forwarding Detection (S-BFD) for IPv4, IPv6, and MPLS*

RFC 7883, *Advertising Seamless Bidirectional Forwarding Detection (S-BFD) Discriminators in IS-IS*

RFC 7884, *OSPF Extensions to Advertise Seamless Bidirectional Forwarding Detection (S-BFD) Target Discriminators*

RFC 9247, *BGP - Link State (BGP-LS) Extensions for Seamless Bidirectional Forwarding Detection (S-BFD)*

7.3 Border Gateway Protocol (BGP)

draft-gredler-idr-bgplu-epe-14, *Egress Peer Engineering using BGP-LU*

draft-hares-idr-update-attrib-low-bits-fix-01, *Update Attribute Flag Low Bits Clarification*

draft-ietf-idr-add-paths-guidelines-08, *Best Practices for Advertisement of Multiple Paths in IBGP*

draft-ietf-idr-best-external-03, *Advertisement of the best external route in BGP*

draft-ietf-idr-bgp-flowspec-oid-03, *Revised Validation Procedure for BGP Flow Specifications*

draft-ietf-idr-bgp-gr-notification-01, *Notification Message support for BGP Graceful Restart*

draft-ietf-idr-bgp-ls-app-specific-attr-16, *Application-Specific Attributes Advertisement with BGP Link-State*

draft-ietf-idr-bgp-ls-flex-algo-06, *Flexible Algorithm Definition Advertisement with BGP Link-State*

draft-ietf-idr-bgp-optimal-route-reflection-10, *BGP Optimal Route Reflection (BGP-ORR)*

draft-ietf-idr-error-handling-03, *Revised Error Handling for BGP UPDATE Messages*

draft-ietf-idr-flowspec-interfaceset-03, *Applying BGP flowspec rules on a specific interface set*

draft-ietf-idr-flowspec-path-redirect-05, *Flowspec Indirection-id Redirect* – localised ID

draft-ietf-idr-flowspec-redirect-ip-02, *BGP Flow-Spec Redirect to IP Action*

draft-ietf-idr-link-bandwidth-03, *BGP Link Bandwidth Extended Community*

draft-ietf-idr-long-lived-gr-00, *Support for Long-lived BGP Graceful Restart*

RFC 1772, *Application of the Border Gateway Protocol in the Internet*

RFC 1997, *BGP Communities Attribute*

RFC 2385, *Protection of BGP Sessions via the TCP MD5 Signature Option*

RFC 2439, *BGP Route Flap Damping*

RFC 2545, *Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*

RFC 2858, *Multiprotocol Extensions for BGP-4*

RFC 2918, *Route Refresh Capability for BGP-4*

RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*

RFC 4360, *BGP Extended Communities Attribute*

RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 4456, *BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)*

RFC 4486, *Subcodes for BGP Cease Notification Message*

RFC 4659, *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*

RFC 4684, *Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*

RFC 4724, *Graceful Restart Mechanism for BGP* – helper mode

RFC 4760, *Multiprotocol Extensions for BGP-4*

RFC 4798, *Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*

RFC 5004, *Avoid BGP Best Path Transitions from One External to Another*

RFC 5065, *Autonomous System Confederations for BGP*

RFC 5291, *Outbound Route Filtering Capability for BGP-4*

RFC 5396, *Textual Representation of Autonomous System (AS) Numbers* – asplain

RFC 5492, *Capabilities Advertisement with BGP-4*

RFC 5668, *4-Octet AS Specific BGP Extended Community*

RFC 6286, *Autonomous-System-Wide Unique BGP Identifier for BGP-4*

RFC 6368, *Internal BGP as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 6793, *BGP Support for Four-Octet Autonomous System (AS) Number Space*

RFC 6810, *The Resource Public Key Infrastructure (RPKI) to Router Protocol*

RFC 6811, *Prefix Origin Validation*

RFC 6996, *Autonomous System (AS) Reservation for Private Use*

RFC 7311, *The Accumulated IGP Metric Attribute for BGP*

RFC 7606, *Revised Error Handling for BGP UPDATE Messages*

RFC 7607, *Codification of AS 0 Processing*

RFC 7674, *Clarification of the Flowspec Redirect Extended Community*

RFC 7752, *North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP*

RFC 7854, *BGP Monitoring Protocol (BMP)*

RFC 7911, *Advertisement of Multiple Paths in BGP*

RFC 7999, *BLACKHOLE Community*

RFC 8092, *BGP Large Communities Attribute*

RFC 8097, *BGP Prefix Origin Validation State Extended Community*

RFC 8212, *Default External BGP (EBGP) Route Propagation Behavior without Policies*

RFC 8277, *Using BGP to Bind MPLS Labels to Address Prefixes*

RFC 8571, *BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions*

RFC 8950, *Advertising IPv4 Network Layer Reachability Information (NLRI) with an IPv6 Next Hop*

RFC 8955, *Dissemination of Flow Specification Rules*

RFC 8956, *Dissemination of Flow Specification Rules for IPv6*

RFC 9086, *Border Gateway Protocol - Link State (BGP-LS) Extensions for Segment Routing BGP Egress Peer Engineering*

7.4 Bridging and management

IEEE 802.1AB, *Station and Media Access Control Connectivity Discovery*

IEEE 802.1ad, *Provider Bridges*

IEEE 802.1ag, *Connectivity Fault Management*

IEEE 802.1ah, *Provider Backbone Bridges*

IEEE 802.1ak, *Multiple Registration Protocol*

IEEE 802.1aq, *Shortest Path Bridging*

IEEE 802.1AX, *Link Aggregation*

IEEE 802.1D, *MAC Bridges*

IEEE 802.1p, *Traffic Class Expediting*

IEEE 802.1Q, *Virtual LANs*

IEEE 802.1s, *Multiple Spanning Trees*

IEEE 802.1w, *Rapid Reconfiguration of Spanning Tree*

IEEE 802.1X, *Port Based Network Access Control*

7.5 Broadband Network Gateway (BNG) Control and User Plane Separation (CUPS)

3GPP TS 23.003, *Numbering, addressing and identification*

3GPP TS 23.007, *Restoration procedures*

3GPP TS 23.402, *Architecture enhancements for non-3GPP accesses* – S2a roaming based on GPRS

3GPP TS 23.501, *System architecture for the 5G System (5GS)*

3GPP TS 23.502, *Procedures for the 5G System (5GS)*

3GPP TS 23.503, *Policy and charging control framework for the 5G System (5GS)*

3GPP TS 24.501, *Non-Access-Stratum (NAS) protocol for 5G System (5GS)*

3GPP TS 29.244, *Interface between the Control Plane and the User Plane nodes*

3GPP TS 29.281, *General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U)*

3GPP TS 29.500, *Technical Realization of Service Based Architecture*

3GPP TS 29.501, *Principles and Guidelines for Services Definition*

3GPP TS 29.502, *Session Management Services*

3GPP TS 29.503, *Unified Data Management Services*

3GPP TS 29.512, *Session Management Policy Control Service*

3GPP TS 29.518, *Access and Mobility Management Services*

3GPP TS 32.255, *5G data connectivity domain charging*

3GPP TS 32.290, *Services, operations and procedures of charging using Service Based Interface (SBI)*

3GPP TS 32.291, *5G system, charging service*

BBF TR-459, *Control and User Plane Separation for a Disaggregated BNG*

BBF TR-459.2, *Multi-Service Disaggregated BNG with CUPS: Integrated Carrier Grade NAT function*

RFC 8300, *Network Service Header (NSH)*

RFC 8910, *Captive-Portal Identification in DHCP and Router Advertisements (RAs)*

7.6 Certificate management

RFC 4210, *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*

RFC 4211, *Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)*

RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*

RFC 6712, *Internet X.509 Public Key Infrastructure -- HTTP Transfer for the Certificate Management Protocol (CMP)*

RFC 7030, *Enrollment over Secure Transport*

RFC 7468, *Textual Encodings of PKIX, PKCS, and CMS Structures*

7.7 Circuit emulation

RFC 4553, *Structure-Agnostic Time Division Multiplexing (TDM) over Packet (SAToP)*

RFC 5086, *Structure-Aware Time Division Multiplexed (TDM) Circuit Emulation Service over Packet Switched Network (CESoPSN)*

RFC 5287, *Control Protocol Extensions for the Setup of Time-Division Multiplexing (TDM) Pseudowires in MPLS Networks*

7.8 Ethernet

IEEE 802.3ah, *Media Access Control Parameters, Physical Layers, and Management Parameters for Subscriber Access Networks*

IEEE 802.3x, *Ethernet Flow Control*

ITU-T G.8031/Y.1342, *Ethernet Linear Protection Switching*

ITU-T G.8032/Y.1344, *Ethernet Ring Protection Switching*

ITU-T Y.1731, *OAM functions and mechanisms for Ethernet based networks*

7.9 Ethernet VPN (EVPN)

draft-ietf-bess-bgp-srv6-args-00, *SRv6 Argument Signaling for BGP Services*

draft-ietf-bess-evpn-ip-aliasing-00, *EVPN Support for L3 Fast Convergence and Aliasing/Backup Path* – IP Prefix routes

draft-ietf-bess-evpn-ipvpn-interworking-06, *EVPN Interworking with IPVPN*

draft-ietf-bess-evpn-irb-mcast-09, *EVPN Optimized Inter-Subnet Multicast (OISM) Forwarding* – ingress replication and mLDP

draft-ietf-bess-evpn-pref-df-06, *Preference-based EVPN DF Election*

draft-ietf-bess-evpn-unequal-lb-16, *Weighted Multi-Path Procedures for EVPN Multi-Homing* – section 9

draft-ietf-bess-evpn-virtual-eth-segment-06, *EVPN Virtual Ethernet Segment*

draft-ietf-bess-pbb-evpn-isid-cmacflush-00, *PBB-EVPN ISID-based CMAC-Flush*

draft-sr-bess-evpn-vpws-gateway-03, *Ethernet VPN Virtual Private Wire Services Gateway Solution*

RFC 7432, *BGP MPLS-Based Ethernet VPN*

RFC 7623, *Provider Backbone Bridging Combined with Ethernet VPN (PBB-EVPN)*

RFC 8214, *Virtual Private Wire Service Support in Ethernet VPN*

RFC 8317, *Ethernet-Tree (E-Tree) Support in Ethernet VPN (EVPN) an Provider Backbone Bridging EVPN (PBB-EVPN)*

RFC 8365, *A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)*

RFC 8560, *Seamless Integration of Ethernet VPN (EVPN) with Virtual Private LAN Service (VPLS) and Their Provider Backbone Bridge (PBB) Equivalents*

RFC 8584, *DF Election and AC-influenced DF Election*

RFC 9047, *Propagation of ARP/ND Flags in an Ethernet Virtual Private Network (EVPN)*

RFC 9135, *Integrated Routing and Bridging in Ethernet VPN (EVPN)* – Asymmetric IRB Procedures and Mobility Procedure

RFC 9136, *IP Prefix Advertisement in Ethernet VPN (EVPN)*

RFC 9161, *Operational Aspects of Proxy ARP/ND in Ethernet Virtual Private Networks*

RFC 9251, *Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Proxies for Ethernet VPN (EVPN)*

7.10 gRPC Remote Procedure Calls (gRPC)

cert.proto version 0.1.0, *gRPC Network Operations Interface (gNOI) Certificate Management Service*

file.proto version 0.1.0, *gRPC Network Operations Interface (gNOI) File Service*

gnmi.proto version 0.8.0, *gRPC Network Management Interface (gNMI) Service Specification*

PROTOCOL-HTTP2, *gRPC over HTTP2*

system.proto Version 1.0.0, *gRPC Network Operations Interface (gNOI) System Service*

7.11 Intermediate System to Intermediate System (IS-IS)

draft-ietf-isis-mi-02, *IS-IS Multi-Instance*

draft-ietf-lsr-igp-ureach-prefix-announce-01, *IGP Unreachable Prefix Announcement* – without U-Flag and UP-Flag

draft-kaplan-isis-ext-eth-02, *Extended Ethernet Frame Size Support*

ISO/IEC 10589:2002 Second Edition, *Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)*

RFC 1195, *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*

RFC 2973, *IS-IS Mesh Groups*

RFC 3359, *Reserved Type, Length and Value (TLV) Codepoints in Intermediate System to Intermediate System*

RFC 3719, *Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)*

RFC 3787, *Recommendations for Interoperable IP Networks using Intermediate System to Intermediate System (IS-IS)*

RFC 5120, *M-ISIS: Multi Topology (MT) Routing in IS-IS*

RFC 5130, *A Policy Control Mechanism in IS-IS Using Administrative Tags*

RFC 5301, *Dynamic Hostname Exchange Mechanism for IS-IS*

RFC 5302, *Domain-wide Prefix Distribution with Two-Level IS-IS*

RFC 5303, *Three-Way Handshake for IS-IS Point-to-Point Adjacencies*

RFC 5304, *IS-IS Cryptographic Authentication*

RFC 5305, *IS-IS Extensions for Traffic Engineering TE*

RFC 5306, *Restart Signaling for IS-IS* – helper mode

RFC 5308, *Routing IPv6 with IS-IS*

RFC 5309, *Point-to-Point Operation over LAN in Link State Routing Protocols*

RFC 5310, *IS-IS Generic Cryptographic Authentication*

RFC 6119, *IPv6 Traffic Engineering in IS-IS*

RFC 6213, *IS-IS BFD-Enabled TLV*

RFC 6232, *Purge Originator Identification TLV for IS-IS*

RFC 6233, *IS-IS Registry Extension for Purges*

RFC 6329, *IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging*

RFC 7775, *IS-IS Route Preference for Extended IP and IPv6 Reachability*

RFC 7794, *IS-IS Prefix Attributes for Extended IPv4 and IPv6 Reachability* – sections 2.1 and 2.3

RFC 7981, *IS-IS Extensions for Advertising Router Information*

RFC 7987, *IS-IS Minimum Remaining Lifetime*

RFC 8202, *IS-IS Multi-Instance* – single topology

RFC 8570, *IS-IS Traffic Engineering (TE) Metric Extensions* – Min/Max Unidirectional Link Delay metric for flex-algo, RSVP, SR-TE

RFC 8919, *IS-IS Application-Specific Link Attributes*

7.12 Internet Protocol (IP) Fast Reroute (FRR)

draft-ietf-rtgwg-lfa-manageability-08, *Operational management of Loop Free Alternates*

RFC 5286, *Basic Specification for IP Fast Reroute: Loop-Free Alternates*

RFC 7431, *Multicast-Only Fast Reroute*

RFC 7490, *Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)*

RFC 8518, *Selection of Loop-Free Alternates for Multi-Homed Prefixes*

7.13 Internet Protocol (IP) general

draft-grant-tacacs-02, *The TACACS+ Protocol*

RFC 768, *User Datagram Protocol*

RFC 793, *Transmission Control Protocol*

RFC 854, *Telnet Protocol Specifications*

RFC 1350, *The TFTP Protocol (revision 2)*

RFC 2347, *TFTP Option Extension*

RFC 2348, *TFTP Blocksize Option*

RFC 2349, *TFTP Timeout Interval and Transfer Size Options*

RFC 2428, *FTP Extensions for IPv6 and NATs*

RFC 2617, *HTTP Authentication: Basic and Digest Access Authentication*

RFC 2784, *Generic Routing Encapsulation (GRE)*

RFC 2818, *HTTP Over TLS*

RFC 2890, *Key and Sequence Number Extensions to GRE*

RFC 3164, *The BSD syslog Protocol*

RFC 4250, *The Secure Shell (SSH) Protocol Assigned Numbers*

RFC 4251, *The Secure Shell (SSH) Protocol Architecture*

RFC 4252, *The Secure Shell (SSH) Authentication Protocol* – publickey, password

RFC 4253, *The Secure Shell (SSH) Transport Layer Protocol*

RFC 4254, *The Secure Shell (SSH) Connection Protocol*

RFC 4511, *Lightweight Directory Access Protocol (LDAP): The Protocol*

RFC 4513, *Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms* – TLS

RFC 4632, *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*

RFC 5082, *The Generalized TTL Security Mechanism (GTSM)*

RFC 5246, *The Transport Layer Security (TLS) Protocol Version 1.2* – TLS client, RSA public key

RFC 5425, *Transport Layer Security (TLS) Transport Mapping for Syslog* – RFC 3164 with TLS

RFC 5656, *Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer* – ECDSA

RFC 5925, *The TCP Authentication Option*

RFC 5926, *Cryptographic Algorithms for the TCP Authentication Option (TCP-AO)*

RFC 6398, *IP Router Alert Considerations and Usage* – MLD

RFC 6528, *Defending against Sequence Number Attacks*

RFC 7011, *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information*

RFC 7012, *Information Model for IP Flow Information Export*

RFC 7230, *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*

RFC 7231, *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*

RFC 7232, *Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests*

RFC 7301, *Transport Layer Security (TLS) Application Layer Protocol Negotiation Extension*

RFC 7616, *HTTP Digest Access Authentication*

RFC 8446, *The Transport Layer Security (TLS) Protocol Version 1.3*

7.14 Internet Protocol (IP) multicast

cisco-ipmulticast/pim-autorp-spec01, *Auto-RP: Automatic discovery of Group-to-RP mappings for IP multicast* – version 1

draft-ietf-bier-pim-signaling-08, *PIM Signaling Through BIER Core*

draft-ietf-idmr-traceroute-ipm-07, *A "traceroute" facility for IP Multicast*

draft-ietf-l2vpn-vpls-pim-snooping-07, *Protocol Independent Multicast (PIM) over Virtual Private LAN Service (VPLS)*

RFC 1112, *Host Extensions for IP Multicasting*

RFC 2236, *Internet Group Management Protocol, Version 2*

RFC 2365, *Administratively Scoped IP Multicast*

RFC 2375, *IPv6 Multicast Address Assignments*

RFC 2710, *Multicast Listener Discovery (MLD) for IPv6*

RFC 3306, *Unicast-Prefix-based IPv6 Multicast Addresses*

RFC 3376, *Internet Group Management Protocol, Version 3*

RFC 3446, *Anycast Rendevous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)*

RFC 3590, *Source Address Selection for the Multicast Listener Discovery (MLD) Protocol*

RFC 3618, *Multicast Source Discovery Protocol (MSDP)*

RFC 3810, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*

RFC 3956, *Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*

RFC 3973, *Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)* – auto-RP groups

RFC 4541, *Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches*

RFC 4604, *Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast*

RFC 4607, *Source-Specific Multicast for IP*

RFC 4608, *Source-Specific Protocol Independent Multicast in 232/8*

RFC 4610, *Anycast-RP Using Protocol Independent Multicast (PIM)*

RFC 4611, *Multicast Source Discovery Protocol (MSDP) Deployment Scenarios*

RFC 5059, *Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)*

RFC 5186, *Internet Group Management Protocol Version 3 (IGMPv3) / Multicast Listener Discovery Version 2 (MLDv2) and Multicast Routing Protocol Interaction*

RFC 5384, *The Protocol Independent Multicast (PIM) Join Attribute Format*

RFC 5496, *The Reverse Path Forwarding (RPF) Vector TLV*

RFC 6037, *Cisco Systems' Solution for Multicast in MPLS/BGP IP VPNs*

RFC 6512, *Using Multipoint LDP When the Backbone Has No Route to the Root*

RFC 6513, *Multicast in MPLS/BGP IP VPNs*

RFC 6514, *BGP Encodings and Procedures for Multicast in MPLS/IP VPNs*

RFC 6515, *IPv4 and IPv6 Infrastructure Addresses in BGP Updates for Multicast VPNs*

RFC 6516, *IPv6 Multicast VPN (MVPN) Support Using PIM Control Plane and Selective Provider Multicast Service Interface (S-PMSI) Join Messages*

RFC 6625, *Wildcards in Multicast VPN Auto-Discover Routes*

RFC 6826, *Multipoint LDP In-Band Signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Path*

RFC 7246, *Multipoint Label Distribution Protocol In-Band Signaling in a Virtual Routing and Forwarding (VRF) Table Context*

RFC 7385, *IANA Registry for P-Multicast Service Interface (PMSI) Tunnel Type Code Points*

RFC 7716, *Global Table Multicast with BGP Multicast VPN (BGP-MVPN) Procedures*

RFC 7761, *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*

RFC 8279, *Multicast Using Bit Index Explicit Replication (BIER)*

RFC 8296, *Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks* – MPLS encapsulation

RFC 8401, *Bit Index Explicit Replication (BIER) Support via IS-IS*

RFC 8444, *OSPFv2 Extensions for Bit Index Explicit Replication (BIER)*

RFC 8487, *Mtrace Version 2: Traceroute Facility for IP Multicast*

RFC 8534, *Explicit Tracking with Wildcard Routes in Multicast VPN* – (C-*,C-*) wildcard

RFC 8556, *Multicast VPN Using Bit Index Explicit Replication (BIER)*

7.15 Internet Protocol (IP) version 4

RFC 791, *Internet Protocol*

RFC 792, *Internet Control Message Protocol*

RFC 826, *An Ethernet Address Resolution Protocol*

RFC 951, *Bootstrap Protocol (BOOTP)* – relay

RFC 1034, *Domain Names - Concepts and Facilities*

RFC 1035, *Domain Names - Implementation and Specification*

RFC 1191, *Path MTU Discovery* – router specification

RFC 1519, *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*

RFC 1534, *Interoperation between DHCP and BOOTP*

RFC 1542, *Clarifications and Extensions for the Bootstrap Protocol*

RFC 1812, *Requirements for IPv4 Routers*

RFC 1918, *Address Allocation for Private Internets*

RFC 2003, *IP Encapsulation within IP*

RFC 2131, *Dynamic Host Configuration Protocol*

RFC 2132, *DHCP Options and BOOTP Vendor Extensions*

RFC 2401, *Security Architecture for Internet Protocol*

RFC 3021, *Using 31-Bit Prefixes on IPv4 Point-to-Point Links*

RFC 3046, *DHCP Relay Agent Information Option (Option 82)*

RFC 3768, *Virtual Router Redundancy Protocol (VRRP)*

RFC 4884, *Extended ICMP to Support Multi-Part Messages* – ICMPv4 and ICMPv6 Time Exceeded

7.16 Internet Protocol (IP) version 6

RFC 2464, *Transmission of IPv6 Packets over Ethernet Networks*

RFC 2529, *Transmission of IPv6 over IPv4 Domains without Explicit Tunnels*

RFC 3122, *Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification*

RFC 3315, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*

RFC 3587, *IPv6 Global Unicast Address Format*

RFC 3596, *DNS Extensions to Support IP version 6*

RFC 3633, *IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6*

RFC 3646, *DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*

RFC 3736, *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6*

RFC 3971, *SEcure Neighbor Discovery (SEND)*

RFC 3972, *Cryptographically Generated Addresses (CGA)*

RFC 4007, *IPv6 Scoped Address Architecture*

RFC 4191, *Default Router Preferences and More-Specific Routes* – Default Router Preference

RFC 4193, *Unique Local IPv6 Unicast Addresses*

RFC 4291, *Internet Protocol Version 6 (IPv6) Addressing Architecture*

RFC 4443, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*

RFC 4861, *Neighbor Discovery for IP version 6 (IPv6)*

RFC 4862, *IPv6 Stateless Address Autoconfiguration* – router functions

RFC 4890, *Recommendations for Filtering ICMPv6 Messages in Firewalls*

RFC 4941, *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*

RFC 5007, *DHCPv6 Leasequery*

RFC 5095, *Deprecation of Type 0 Routing Headers in IPv6*

RFC 5722, *Handling of Overlapping IPv6 Fragments*

RFC 5798, *Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6* – IPv6

RFC 5952, *A Recommendation for IPv6 Address Text Representation*

RFC 6092, *Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service* – Internet Control and Management, Upper-Layer Transport Protocols, UDP Filters, IPsec and Internet Key Exchange (IKE), TCP Filters

RFC 6106, *IPv6 Router Advertisement Options for DNS Configuration*

RFC 6164, *Using 127-Bit IPv6 Prefixes on Inter-Router Links*

RFC 6221, *Lightweight DHCPv6 Relay Agent*

RFC 6437, *IPv6 Flow Label Specification*

RFC 6603, *Prefix Exclude Option for DHCPv6-based Prefix Delegation*

RFC 8021, *Generation of IPv6 Atomic Fragments Considered Harmful*

RFC 8200, *Internet Protocol, Version 6 (IPv6) Specification*

RFC 8201, *Path MTU Discovery for IP version 6*

7.17 Internet Protocol Security (IPsec)

draft-ietf-ipsec-isakmp-mode-cfg-05, *The ISAKMP Configuration Method*

draft-ietf-ipsec-isakmp-xauth-06, *Extended Authentication within ISAKMP/Oakley (XAUTH)*

RFC 2401, *Security Architecture for the Internet Protocol*

RFC 2403, *The Use of HMAC-MD5-96 within ESP and AH*

RFC 2404, *The Use of HMAC-SHA-1-96 within ESP and AH*

RFC 2405, *The ESP DES-CBC Cipher Algorithm With Explicit IV*

RFC 2406, *IP Encapsulating Security Payload (ESP)*

RFC 2407, *IPsec Domain of Interpretation for ISAKMP (IPsec DoI)*

RFC 2408, *Internet Security Association and Key Management Protocol (ISAKMP)*

RFC 2409, *The Internet Key Exchange (IKE)*

RFC 2410, *The NULL Encryption Algorithm and Its Use With IPsec*

RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*

RFC 3526, *More Modular Exponential (MODP) Diffie-Hellman group for Internet Key Exchange (IKE)*

RFC 3566, *The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec*

RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*

RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*

RFC 3947, *Negotiation of NAT-Traversal in the IKE*

RFC 3948, *UDP Encapsulation of IPsec ESP Packets*

RFC 4106, *The Use of Galois/Counter Mode (GCM) in IPsec ESP*

RFC 4109, *Algorithms for Internet Key Exchange version 1 (IKEv1)*

RFC 4301, *Security Architecture for the Internet Protocol*

RFC 4303, *IP Encapsulating Security Payload*

RFC 4307, *Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)*

RFC 4308, *Cryptographic Suites for IPsec*

RFC 4434, *The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)*

RFC 4543, *The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH*

RFC 4754, *IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)*

RFC 4835, *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*

RFC 4868, *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec*

RFC 4945, *The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2 and PKIX*

RFC 5019, *The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments*

RFC 5282, *Using Authenticated Encryption Algorithms with the Encrypted Payload of the IKEv2 Protocol*

RFC 5903, *ECP Groups for IKE and IKEv2*

RFC 5996, *Internet Key Exchange Protocol Version 2 (IKEv2)*

RFC 5998, *An Extension for EAP-Only Authentication in IKEv2*

RFC 6379, *Suite B Cryptographic Suites for IPsec*

RFC 6380, *Suite B Profile for Internet Protocol Security (IPsec)*

RFC 6960, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*

RFC 7296, *Internet Key Exchange Protocol Version 2 (IKEv2)*

RFC 7321, *Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)*

RFC 7383, *Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation*

RFC 7427, *Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)*

7.18 Label Distribution Protocol (LDP)

draft-pdutta-mpls-ldp-adj-capability-00, *LDP Adjacency Capabilities*

draft-pdutta-mpls-ldp-v2-00, *LDP Version 2*

draft-pdutta-mpls-mldp-up-redundancy-00, *Upstream LSR Redundancy for Multi-point LDP Tunnels*

draft-pdutta-mpls-multi-ldp-instance-00, *Multiple LDP Instances*

draft-pdutta-mpls-tldp-hello-reduce-04, *Targeted LDP Hello Reduction*

RFC 3037, *LDP Applicability*

RFC 3478, *Graceful Restart Mechanism for Label Distribution Protocol* – helper mode

RFC 5036, *LDP Specification*

RFC 5283, *LDP Extension for Inter-Area Label Switched Paths (LSPs)*

RFC 5443, *LDP IGP Synchronization*

RFC 5561, *LDP Capabilities*

RFC 5919, *Signaling LDP Label Advertisement Completion*

RFC 6388, *Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths*

RFC 6512, *Using Multipoint LDP When the Backbone Has No Route to the Root*

RFC 6826, *Multipoint LDP in-band signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths*

RFC 7032, *LDP Downstream-on-Demand in Seamless MPLS*

RFC 7473, *Controlling State Advertisements of Non-negotiated LDP Applications*

RFC 7552, *Updates to LDP for IPv6*

7.19 Layer Two Tunneling Protocol (L2TP) Network Server (LNS)

draft-mammoliti-l2tp-accessline-avp-04, *Layer 2 Tunneling Protocol (L2TP) Access Line Information Attribute Value Pair (AVP) Extensions*

RFC 2661, *Layer Two Tunneling Protocol "L2TP"*

RFC 2809, *Implementation of L2TP Compulsory Tunneling via RADIUS*

RFC 3438, *Layer Two Tunneling Protocol (L2TP) Internet Assigned Numbers: Internet Assigned Numbers Authority (IANA) Considerations Update*

RFC 3931, *Layer Two Tunneling Protocol - Version 3 (L2TPv3)*

RFC 4719, *Transport of Ethernet Frames over Layer 2 Tunneling Protocol Version 3 (L2TPv3)*

RFC 4951, *Fail Over Extensions for Layer 2 Tunneling Protocol (L2TP) "failover"*

7.20 Multiprotocol Label Switching (MPLS)

draft-ietf-mpls-lsp-ping-ospfv3-codepoint-02, *OSPFv3 CodePoint for MPLS LSP Ping*

RFC 3031, *Multiprotocol Label Switching Architecture*

RFC 3032, *MPLS Label Stack Encoding*

RFC 3270, *Multi-Protocol Label Switching (MPLS) Support of Differentiated Services* – E-LSP

RFC 3443, *Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks*

RFC 4023, *Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)*

RFC 4182, *Removing a Restriction on the use of MPLS Explicit NULL*

RFC 4950, *ICMP Extensions for Multiprotocol Label Switching*

RFC 5332, *MPLS Multicast Encapsulations*

RFC 5884, *Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)*

RFC 6374, *Packet Loss and Delay Measurement for MPLS Networks* – Delay Measurement, Channel Type 0x000C

RFC 6424, *Mechanism for Performing Label Switched Path Ping (LSP Ping) over MPLS Tunnels*

RFC 6425, *Detecting Data Plane Failures in Point-to-Multipoint Multiprotocol Label Switching (MPLS) - Extensions to LSP Ping*

RFC 6790, *The Use of Entropy Labels in MPLS Forwarding*

RFC 7308, *Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)*

RFC 7510, *Encapsulating MPLS in UDP*

RFC 7746, *Label Switched Path (LSP) Self-Ping*

RFC 7876, *UDP Return Path for Packet Loss and Delay Measurement for MPLS Networks* – Delay Measurement

RFC 8029, *Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures*

7.21 Multiprotocol Label Switching - Transport Profile (MPLS-TP)

RFC 5586, *MPLS Generic Associated Channel*

RFC 5921, *A Framework for MPLS in Transport Networks*

RFC 5960, *MPLS Transport Profile Data Plane Architecture*

RFC 6370, *MPLS Transport Profile (MPLS-TP) Identifiers*

RFC 6378, *MPLS Transport Profile (MPLS-TP) Linear Protection*

RFC 6426, *MPLS On-Demand Connectivity and Route Tracing*

RFC 6427, *MPLS Fault Management Operations, Administration, and Maintenance (OAM)*

RFC 6428, *Proactive Connectivity Verification, Continuity Check and Remote Defect indication for MPLS Transport Profile*

RFC 6478, *Pseudowire Status for Static Pseudowires*

RFC 7213, *MPLS Transport Profile (MPLS-TP) Next-Hop Ethernet Addressing*

7.22 Network Address Translation (NAT)

draft-ietf-behave-address-format-10, *IPv6 Addressing of IPv4/IPv6 Translators*

draft-ietf-behave-v6v4-xlate-23, *IP/ICMP Translation Algorithm*

draft-miles-behave-l2nat-00, *Layer2-Aware NAT*

draft-nishitani-cgn-02, *Common Functions of Large Scale NAT (LSN)*

RFC 4787, *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*

RFC 5382, *NAT Behavioral Requirements for TCP*

RFC 5508, *NAT Behavioral Requirements for ICMP*

RFC 6146, *Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers*

RFC 6333, *Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion*

RFC 6334, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite*

RFC 6887, *Port Control Protocol (PCP)*

RFC 6888, *Common Requirements For Carrier-Grade NATs (CGNs)*

RFC 7753, *Port Control Protocol (PCP) Extension for Port-Set Allocation*

RFC 7915, *IP/ICMP Translation Algorithm*

7.23 Network Configuration Protocol (NETCONF)

RFC 5277, *NETCONF Event Notifications*

RFC 6020, *YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)*

RFC 6022, *YANG Module for NETCONF Monitoring*

RFC 6241, *Network Configuration Protocol (NETCONF)*

RFC 6242, *Using the NETCONF Protocol over Secure Shell (SSH)*

RFC 6243, *With-defaults Capability for NETCONF*

RFC 8342, *Network Management Datastore Architecture (NMDA)* – Startup, Candidate, Running and Intended datastores

RFC 8525, *YANG Library*

RFC 8526, *NETCONF Extensions to Support the Network Management Datastore Architecture* – <get-data> operation

7.24 Open Shortest Path First (OSPF)

RFC 1765, *OSPF Database Overflow*

RFC 2328, *OSPF Version 2*

RFC 3101, *The OSPF Not-So-Stubby Area (NSSA) Option*

RFC 3509, *Alternative Implementations of OSPF Area Border Routers*

RFC 3623, *Graceful OSPF Restart Graceful OSPF Restart* – helper mode

RFC 3630, *Traffic Engineering (TE) Extensions to OSPF Version 2*

RFC 4222, *Prioritized Treatment of Specific OSPF Version 2 Packets and Congestion Avoidance*

RFC 4552, *Authentication/Confidentiality for OSPFv3*

RFC 4576, *Using a Link State Advertisement (LSA) Options Bit to Prevent Looping in BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 4577, *OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 5185, *OSPF Multi-Area Adjacency*

RFC 5187, *OSPFv3 Graceful Restart* – helper mode

RFC 5243, *OSPF Database Exchange Summary List Optimization*

RFC 5250, *The OSPF Opaque LSA Option*

RFC 5309, *Point-to-Point Operation over LAN in Link State Routing Protocols*

RFC 5340, *OSPF for IPv6*

RFC 5642, *Dynamic Hostname Exchange Mechanism for OSPF*

RFC 5709, *OSPFv2 HMAC-SHA Cryptographic Authentication*

RFC 5838, *Support of Address Families in OSPFv3*

RFC 6549, *OSPFv2 Multi-Instance Extensions*

RFC 6987, *OSPF Stub Router Advertisement*

RFC 7471, *OSPF Traffic Engineering (TE) Metric Extensions* – Min/Max Unidirectional Link Delay metric for flex-algo, RSVP, SR-TE

RFC 7684, *OSPFv2 Prefix/Link Attribute Advertisement*

RFC 7770, *Extensions to OSPF for Advertising Optional Router Capabilities*

RFC 8362, *OSPFv3 Link State Advertisement (LSA) Extensibility*

RFC 8920, *OSPF Application-Specific Link Attributes*

7.25 OpenFlow

TS-007 Version 1.3.1, *OpenFlow Switch Specification* – OpenFlow-hybrid switches

7.26 Path Computation Element Protocol (PCEP)

draft-alvarez-pce-path-profiles-04, *PCE Path Profiles*

draft-dhs-spring-pce-sr-p2mp-policy-00, *PCEP extensions for p2mp sr policy*

draft-ietf-pce-binding-label-sid-15, *Carrying Binding Label/Segment Identifier (SID) in PCE-based Networks.* – MPLS binding SIDs

draft-ietf-pce-pceps-tls13-04, *Updates for PCEPS: TLS Connection Establishment Restrictions*

RFC 5440, *Path Computation Element (PCE) Communication Protocol (PCEP)*

RFC 8231, *Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE*

RFC 8253, *PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)*

RFC 8281, *PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model*

RFC 8408, *Conveying Path Setup Type in PCE Communication Protocol (PCEP) Messages*

RFC 8664, *Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing*

7.27 Point-to-Point Protocol (PPP)

RFC 1332, *The PPP Internet Protocol Control Protocol (IPCP)*

RFC 1990, *The PPP Multilink Protocol (MP)*

RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*

RFC 2516, *A Method for Transmitting PPP Over Ethernet (PPPoE)*

RFC 4638, *Accommodating a Maximum Transit Unit/Maximum Receive Unit (MTU/MRU) Greater Than 1492 in the Point-to-Point Protocol over Ethernet (PPPoE)*

RFC 5072, *IP Version 6 over PPP*

7.28 Policy management and credit control

3GPP TS 29.212 Release 11, *Policy and Charging Control (PCC); Reference points* – Gx support as it applies to wireline environment (BNG)

RFC 4006, *Diameter Credit-Control Application*

RFC 6733, *Diameter Base Protocol*

7.29 Pseudowire (PW)

draft-ietf-l2vpn-vpws-iw-oam-04, *OAM Procedures for VPWS Interworking*

MFA Forum 12.0.0, *Multiservice Interworking - Ethernet over MPLS*

MFA Forum 13.0.0, *Fault Management for Multiservice Interworking v1.0*

MFA Forum 16.0.0, *Multiservice Interworking - IP over MPLS*

RFC 3916, *Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)*

RFC 3985, *Pseudo Wire Emulation Edge-to-Edge (PWE3)*

RFC 4385, *Pseudo Wire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN*

RFC 4446, *IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)*

RFC 4447, *Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*

RFC 4448, *Encapsulation Methods for Transport of Ethernet over MPLS Networks*

RFC 5085, *Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires*

RFC 5659, *An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge*

RFC 5885, *Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)*

RFC 6073, *Segmented Pseudowire*

RFC 6310, *Pseudowire (PW) Operations, Administration, and Maintenance (OAM) Message Mapping*

RFC 6391, *Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network*

RFC 6575, *Address Resolution Protocol (ARP) Mediation for IP Interworking of Layer 2 VPNs*

RFC 6718, *Pseudowire Redundancy*

RFC 6829, *Label Switched Path (LSP) Ping for Pseudowire Forwarding Equivalence Classes (FECs) Advertised over IPv6*

RFC 6870, *Pseudowire Preferential Forwarding Status bit*

RFC 7023, *MPLS and Ethernet Operations, Administration, and Maintenance (OAM) Interworking*

RFC 7267, *Dynamic Placement of Multi-Segment Pseudowires*

RFC 7392, *Explicit Path Routing for Dynamic Multi-Segment Pseudowires* – ER-TLV and ER-HOP IPv4 Prefix

RFC 8395, *Extensions to BGP-Signaled Pseudowires to Support Flow-Aware Transport Labels*

7.30 Quality of Service (QoS)

RFC 2430, *A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)*

RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*

RFC 2597, *Assured Forwarding PHB Group*

RFC 3140, *Per Hop Behavior Identification Codes*

RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*

7.31 Remote Authentication Dial In User Service (RADIUS)

draft-oscca-cfrg-sm3-02, *The SM3 Cryptographic Hash Function*

RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*

RFC 2866, *RADIUS Accounting*

RFC 2867, *RADIUS Accounting Modifications for Tunnel Protocol Support*

RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*

RFC 2869, *RADIUS Extensions*

RFC 3162, *RADIUS and IPv6*

RFC 4818, *RADIUS Delegated-IPv6-Prefix Attribute*

RFC 5176, *Dynamic Authorization Extensions to RADIUS*

RFC 6613, *RADIUS over TCP* – with TLS

RFC 6614, *Transport Layer Security (TLS) Encryption for RADIUS*

RFC 6929, *Remote Authentication Dial-In User Service (RADIUS) Protocol Extensions*

RFC 6911, *RADIUS attributes for IPv6 Access Networks*

7.32 Resource Reservation Protocol - Traffic Engineering (RSVP-TE)

draft-newton-mpls-te-dynamic-overbooking-00, *A Diffserv-TE Implementation Model to dynamically change booking factors during failure events*

RFC 2702, *Requirements for Traffic Engineering over MPLS*

RFC 2747, *RSVP Cryptographic Authentication*

RFC 2961, *RSVP Refresh Overhead Reduction Extensions*

RFC 3097, *RSVP Cryptographic Authentication -- Updated Message Type Value*

RFC 3209, *RSVP-TE: Extensions to RSVP for LSP Tunnels*

RFC 3477, *Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)*

RFC 3564, *Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering*

RFC 3906, *Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels*

RFC 4090, *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*

RFC 4124, *Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering*

RFC 4125, *Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*

RFC 4127, *Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*

RFC 4561, *Definition of a Record Route Object (RRO) Node-Id Sub-Object*

RFC 4875, *Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)*

RFC 5712, *MPLS Traffic Engineering Soft Preemption*

RFC 5817, *Graceful Shutdown in MPLS and Generalized MPLS Traffic Engineering Networks*

7.33 Routing Information Protocol (RIP)

RFC 1058, *Routing Information Protocol*

RFC 2080, *RIPng for IPv6*

RFC 2082, *RIP-2 MD5 Authentication*

RFC 2453, *RIP Version 2*

7.34 Segment Routing (SR)

draft-ietf-bess-mvpn-evpn-sr-p2mp-07, *Multicast and Ethernet VPN with Segment Routing P2MP and Ingress Replication* – MVPN

draft-bashandy-rtgwg-segment-routing-uloop-15, *Loop avoidance using Segment Routing*

draft-filsfils-spring-net-pgm-extension-srv6-usid-15, *Network Programming extension: SRv6 uSID instruction*

draft-filsfils-spring-srv6-net-pgm-insertion-08, *SRv6 NET-PGM extension: Insertion*

draft-ietf-idr-bgpls-srv6-ext-14, *BGP Link State Extensions for SRv6*

draft-ietf-idr-segment-routing-te-policy-23, *Advertising Segment Routing Policies in BGP*

draft-ietf-idr-ts-flowspec-srv6-policy-03, *Traffic Steering using BGP FlowSpec with SR Policy*

draft-ietf-pim-p2mp-policy-ping-03, *P2MP Policy Ping*

draft-ietf-pim-sr-p2mp-policy-06, *Segment Routing Point-to-Multipoint Policy* – MPLS

draft-ietf-rtgwg-segment-routing-ti-lfa-11, *Topology Independent Fast Reroute using Segment Routing*

draft-ietf-spring-conflict-resolution-05, *Segment Routing MPLS Conflict Resolution*

draft-ietf-spring-sr-replication-segment-16, *SR Replication segment for Multi-point Service Delivery* – MPLS

draft-ietf-spring-srv6-srh-compression-xx, *Compressed SRv6 Segment List Encoding in SRH*

draft-voyer-6man-extension-header-insertion-10, *Deployments With Insertion of IPv6 Segment Routing Headers*

RFC 8287, *Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes*

RFC 8426, *Recommendations for RSVP-TE and Segment Routing (SR) Label Switched Path (LSP) Coexistence*

RFC 8476, *Signaling Maximum SID Depth (MSD) Using OSPF* – node MSD

RFC 8491, *Signaling Maximum SID Depth (MSD) Using IS-IS* – node MSD

RFC 8660, *Segment Routing with the MPLS Data Plane*

RFC 8661, *Segment Routing MPLS Interworking with LDP*

RFC 8663, *MPLS Segment Routing over IP* – BGP SR with SR-MPLS-over-UDP/IP

RFC 8665, *OSPF Extensions for Segment Routing*

RFC 8666, *OSPFv3 Extensions for Segment Routing*

RFC 8667, *IS-IS Extensions for Segment Routing*

RFC 8669, *Segment Routing Prefix Segment Identifier Extensions for BGP*

RFC 8754, *IPv6 Segment Routing Header (SRH)*

RFC 8814, *Signaling Maximum SID Depth (MSD) Using the Border Gateway Protocol - Link State*

RFC 8986, *Segment Routing over IPv6 (SRv6) Network Programming*

RFC 9085, *Border Gateway Protocol - Link State (BGP-LS) Extensions for Segment Routing*

RFC 9088, *Signaling Entropy Label Capability and Entropy Readable Label Depth Using IS-IS* – advertising ELC

RFC 9089, *Signaling Entropy Label Capability and Entropy Readable Label Depth Using OSPF* – advertising ELC

RFC 9252, *BGP Overlay Services Based on Segment Routing over IPv6 (SRv6)*

RFC 9256, *Segment Routing Policy Architecture*

RFC 9259, *Operations, Administration, and Maintenance (OAM) in Segment Routing over IPv6 (SRv6)*

RFC 9350, *IGP Flexible Algorithm*

RFC 9352, *IS-IS Extensions to Support Segment Routing over the IPv6 Data Plane*

7.35 Simple Network Management Protocol (SNMP)

draft-blumenthal-aes-usm-04, *The AES Cipher Algorithm in the SNMP's User-based Security Model* – CFB128-AES-192 and CFB128-AES-256

draft-ietf-isis-wg-mib-06, *Management Information Base for Intermediate System to Intermediate System (IS-IS)*

draft-ietf-mboned-msdp-mib-01, *Multicast Source Discovery protocol MIB*

draft-ietf-mpls-ldp-mib-07, *Definitions of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol (LDP)*

draft-ietf-mpls-lsr-mib-06, *Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base Using SMIv2*

draft-ietf-mpls-te-mib-04, *Multiprotocol Label Switching (MPLS) Traffic Engineering Management Information Base*

draft-ietf-ospf-mib-update-08, *OSPF Version 2 Management Information Base*

draft-ietf-vrrp-unified-mib-06, *Definitions of Managed Objects for the VRRP over IPv4 and IPv6* – IPv6

ESO-CONSORTIUM-MIB revision 200406230000Z, *esoConsortiumMIB*

IANA-ADDRESS-FAMILY-NUMBERS-MIB revision 200203140000Z, *ianaAddressFamilyNumbers*

IANAifType-MIB revision 200505270000Z, *ianaifType*

IANA-RTPROTO-MIB revision 200009260000Z, *ianaRtProtoMIB*

IEEE8021-CFM-MIB revision 200706100000Z, *ieee8021CfmMib*

IEEE8021-PAE-MIB revision 200101160000Z, *ieee8021paeMIB*

IEEE8023-LAG-MIB revision 200006270000Z, *lagMIB*

LLDP-MIB revision 200505060000Z, *lldpMIB*

RFC 1157, *A Simple Network Management Protocol (SNMP)*

RFC 1212, *Concise MIB Definitions*

RFC 1215, *A Convention for Defining Traps for use with the SNMP*

RFC 1724, *RIP Version 2 MIB Extension*

RFC 1901, *Introduction to Community-based SNMPv2*

RFC 2021, *Remote Network Monitoring Management Information Base Version 2 using SMIv2*

RFC 2206, *RSVP Management Information Base using SMIv2*

RFC 2213, *Integrated Services Management Information Base using SMIv2*

RFC 2494, *Definitions of Managed Objects for the DS0 and DS0 Bundle Interface Type*

RFC 2578, *Structure of Management Information Version 2 (SMIv2)*

RFC 2579, *Textual Conventions for SMIv2*

RFC 2580, *Conformance Statements for SMIv2*

RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol*

RFC 2819, *Remote Network Monitoring Management Information Base*

RFC 2856, *Textual Conventions for Additional High Capacity Data Types*

RFC 2863, *The Interfaces Group MIB*

RFC 2864, *The Inverted Stack Table Extension to the Interfaces Group MIB*

RFC 2933, *Internet Group Management Protocol MIB*

RFC 3014, *Notification Log MIB*

RFC 3165, *Definitions of Managed Objects for the Delegation of Management Scripts*

RFC 3231, *Definitions of Managed Objects for Scheduling Management Operations*

RFC 3273, *Remote Network Monitoring Management Information Base for High Capacity Networks*

RFC 3410, *Introduction and Applicability Statements for Internet Standard Management Framework*

RFC 3411, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*

RFC 3412, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*

RFC 3413, *Simple Network Management Protocol (SNMP) Applications*

RFC 3414, *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*

RFC 3415, *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*

RFC 3416, *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*

RFC 3417, *Transport Mappings for the Simple Network Management Protocol (SNMP)* – SNMP over UDP over IPv4

RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*

RFC 3419, *Textual Conventions for Transport Addresses*

RFC 3498, *Definitions of Managed Objects for Synchronous Optical Network (SONET) Linear Automatic Protection Switching (APS) Architectures*

RFC 3584, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*

RFC 3592, *Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Interface Type*

RFC 3593, *Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals*

RFC 3635, *Definitions of Managed Objects for the Ethernet-like Interface Types*

RFC 3637, *Definitions of Managed Objects for the Ethernet WAN Interface Sublayer*

RFC 3826, *The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model*

RFC 3877, *Alarm Management Information Base (MIB)*

RFC 3895, *Definitions of Managed Objects for the DS1, E1, DS2, and E2 Interface Types*

RFC 3896, *Definitions of Managed Objects for the DS3/E3 Interface Type*

RFC 4001, *Textual Conventions for Internet Network Addresses*

RFC 4022, *Management Information Base for the Transmission Control Protocol (TCP)*

RFC 4113, *Management Information Base for the User Datagram Protocol (UDP)*

RFC 4220, *Traffic Engineering Link Management Information Base*

RFC 4273, *Definitions of Managed Objects for BGP-4*

RFC 4292, *IP Forwarding Table MIB*

RFC 4293, *Management Information Base for the Internet Protocol (IP)*

RFC 4631, *Link Management Protocol (LMP) Management Information Base (MIB)*

RFC 4878, *Definitions and Managed Objects for Operations, Administration, and Maintenance (OAM) Functions on Ethernet-Like Interfaces*

RFC 7420, *Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module*

RFC 7630, *HMAC-SHA-2 Authentication Protocols in the User-based Security Model (USM) for SNMPv3*

SFLOW-MIB revision 200309240000Z, *sFlowMIB*

7.36 Timing

GR-1244-CORE Issue 3, *Clocks for the Synchronized Network: Common Generic Criteria*

GR-253-CORE Issue 3, *SONET Transport Systems: Common Generic Criteria*

IEEE 1588-2008, *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*

ITU-T G.781, *Synchronization layer functions*

ITU-T G.811, *Timing characteristics of primary reference clocks*

ITU-T G.813, *Timing characteristics of SDH equipment slave clocks (SEC)*

ITU-T G.8261, *Timing and synchronization aspects in packet networks*

ITU-T G.8262, *Timing characteristics of synchronous Ethernet equipment slave clock (EEC)*

ITU-T G.8262.1, *Timing characteristics of an enhanced synchronous Ethernet equipment slave clock (eEEC)*

ITU-T G.8264, *Distribution of timing information through packet networks*

ITU-T G.8265.1, *Precision time protocol telecom profile for frequency synchronization*

ITU-T G.8272 , *Timing characteristics of primary reference time clocks* – PRTC-A, PRTC-B

ITU-T G.8275.1, *Precision time protocol telecom profile for phase/time synchronization with full timing support from the network*

ITU-T G.8275.2, *Precision time protocol telecom profile for phase/time synchronization with partial timing support from the network*

RFC 3339, *Date and Time on the Internet: Timestamps*

RFC 5905, *Network Time Protocol Version 4: Protocol and Algorithms Specification*

RFC 8573, *Message Authentication Code for the Network Time Protocol*

7.37 Two-Way Active Measurement Protocol (TWAMP)

RFC 5357, *A Two-Way Active Measurement Protocol (TWAMP)* – server, unauthenticated mode

RFC 5938, *Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)*

RFC 6038, *Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features*

RFC 8545, *Well-Known Port Assignments for the One-Way Active Measurement Protocol (OWAMP) and the Two-Way Active Measurement Protocol (TWAMP)* – TWAMP

RFC 8762, *Simple Two-Way Active Measurement Protocol* – unauthenticated

RFC 8972, *Simple Two-Way Active Measurement Protocol Optional Extensions* – unauthenticated

7.38 Virtual Private LAN Service (VPLS)

RFC 4761, *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*

RFC 4762, *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*

RFC 5501, *Requirements for Multicast Support in Virtual Private LAN Services*

RFC 6074, *Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)*

RFC 7041, *Extensions to the Virtual Private LAN Service (VPLS) Provider Edge (PE) Model for Provider Backbone Bridging*

RFC 7117, *Multicast in Virtual Private LAN Service (VPLS)*

7.39 Voice and video

DVB BlueBook A86, *Transport of MPEG-2 TS Based DVB Services over IP Based Networks*

ETSI TS 101 329-5 Annex E, *QoS Measurement for VoIP - Method for determining an Equipment Impairment Factor using Passive Monitoring*

ITU-T G.1020 Appendix I, *Performance Parameter Definitions for Quality of Speech and other Voiceband Applications Utilizing IP Networks - Mean Absolute Packet Delay Variation & Markov Models*

ITU-T G.107, *The E Model - A computational model for use in planning*

ITU-T P.564, *Conformance testing for voice over IP transmission quality assessment models*

RFC 3550, *RTP: A Transport Protocol for Real-Time Applications* – Appendix A.8

RFC 4585, *Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)*

RFC 4588, *RTP Retransmission Payload Format*

7.40 Yet Another Next Generation (YANG)

RFC 6991, *Common YANG Data Types*

RFC 7950, *The YANG 1.1 Data Modeling Language*

RFC 7951, *JSON Encoding of Data Modeled with YANG*

7.41 Yet Another Next Generation (YANG) OpenConfig Models

openconfig-aaa.yang version 0.4.0, *OpenConfig AAA Model*

openconfig-aaa-radius.yang version 0.3.0, *OpenConfig AAA RADIUS Model*

openconfig-aaa-tacacs.yang version 0.3.0, *OpenConfig AAA TACACS+ Model*

openconfig-acl.yang version 1.0.0, *OpenConfig ACL Model*

openconfig-alarms.yang version0.3.2, *OpenConfig System Alarms Model*

openconfig-bfd.yang version 0.2.2, *OpenConfig BFD Model*

openconfig-bgp.yang version 6.1.0, *OpenConfig BGP Model*

openconfig-bgp-common.yang version 6.0.0, *OpenConfig BGP Common Model*

openconfig-bgp-common-multiprotocol.yang version 6.0.0, *OpenConfig BGP Common Multiprotocol Model*

openconfig-bgp-common-structure.yang version 6.0.0, *OpenConfig BGP Common Structure Model*

openconfig-bgp-global.yang version 6.0.0, *OpenConfig BGP Global Model*

openconfig-bgp-neighbor.yang version 6.1.0, *OpenConfig BGP Neighbor Model*

openconfig-bgp-peer-group.yang version 6.1.0, *OpenConfig BGP Peer Group Model*

openconfig-bgp-policy.yang version 4.0.1, *OpenConfig BGP Policy Model*

openconfig-if-aggregate.yang version 2.4.3, *OpenConfig Interfaces Aggregated Model*

openconfig-if-ethernet.yang version 2.12.1, *OpenConfig Interfaces Ethernet Model*

openconfig-if-ip.yang version 3.1.0, *OpenConfig Interfaces IP Model*

openconfig-if-ip-ext.yang version 2.3.1, *OpenConfig Interfaces IP Extensions Model*

openconfig-igmp.yang version 0.2.0, *OpenConfig IGMP Model*

openconfig-interfaces.yang version 3.0.0, *OpenConfig Interfaces Model*

openconfig-isis.yang version 1.1.0, *OpenConfig IS-IS Model*

openconfig-isis-policy.yang version 0.5.0, *OpenConfig IS-IS Policy Model*

openconfig-isis-routing.yang version 1.1.0, *OpenConfig IS-IS Routing Model*

openconfig-lacp.yang version 1.3.0, *OpenConfig LACP Model*

openconfig-lldp.yang version 0.1.0, *OpenConfig LLDP Model*

openconfig-local-routing.yang version 1.2.0, *OpenConfig Local Routing Model*

openconfig-mpls.yang version 2.3.0, *OpenConfig MPLS Model*

openconfig-mpls-ldp.yang version 3.0.2, *OpenConfig MPLS LDP Model*

openconfig-mpls-rsvp.yang version 2.3.0, *OpenConfig MPLS RSVP Model*

openconfig-mpls-te.yang version 2.3.0, *OpenConfig MPLS TE Model*

openconfig-network-instance.yang version 1.1.0, *OpenConfig Network Instance Model*

openconfig-network-instance-l3.yang version 0.11.1, *OpenConfig L3 Network Instance Model* – static routes

openconfig-ospfv2.yang version 0.4.0, *OpenConfig OSPFv2 Model*

openconfig-ospfv2-area.yang version 0.4.0, *OpenConfig OSPFv2 Area Model*

openconfig-ospfv2-area-interface.yang version 0.4.0, *OpenConfig OSPFv2 Area Interface Model*

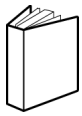
openconfig-ospfv2-common.yang version 0.4.0, *OpenConfig OSPFv2 Common Model*

openconfig-ospfv2-global.yang version 0.4.0, *OpenConfig OSPFv2 Global Model*

openconfig-packet-match.yang version 1.0.0, *OpenConfig Packet Match Model*

openconfig-pim.yang version 0.2.0, *OpenConfig PIM Model*
openconfig-platform.yang version 0.15.0, *OpenConfig Platform Model*
openconfig-platform-fan.yang version 0.1.1, *OpenConfig Platform Fan Model*
openconfig-platform-linecard.yang version 0.1.2, *OpenConfig Platform Linecard Model*
openconfig-platform-port.yang version 0.4.2, *OpenConfig Port Model*
openconfig-platform-transceiver.yang version 0.9.0, *OpenConfig Transceiver Model*
openconfig-procmon.yang version 0.4.0, *OpenConfig Process Monitoring Model*
openconfig-relay-agent.yang version 0.1.0, *OpenConfig Relay Agent Model*
openconfig-routing-policy.yang version 3.0.0, *OpenConfig Routing Policy Model*
openconfig-rsvp-sr-ext.yang version 0.1.0, *OpenConfig RSVP-TE and SR Extensions Model*
openconfig-system.yang version 0.10.1, *OpenConfig System Model*
openconfig-system-grpc.yang version 1.0.0, *OpenConfig System gRPC Model*
openconfig-system-logging.yang version 0.3.1, *OpenConfig System Logging Model*
openconfig-system-terminal.yang version 0.3.0, *OpenConfig System Terminal Model*
openconfig-telemetry.yang version 0.5.0, *OpenConfig Telemetry Model*
openconfig-terminal-device.yang version 1.9.0, *OpenConfig Terminal Optics Device Model*
openconfig-vlan.yang version 2.0.0, *OpenConfig VLAN Model*

Customer document and product support



Customer documentation

Customer documentation welcome page

Technical support

Product support portal

Documentation feedback

Customer documentation feedback