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

Release 26.3.R1

Multicast Routing Protocols Guide

3HE 22313 AAAA TQZZA 01
Edition: 01
March 2026

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

1.1 About this guide

This guide describes multicast routing protocols, troubleshooting, and proprietary entities and presents configuration and implementation examples.

This guide 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.

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

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

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

Note: Unless otherwise indicated, CLI commands, contexts, and configuration examples in this guide apply for both the MD-CLI and the classic CLI.

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 the MD-CLI and the classic CLI)
- *7450 ESS, 7750 SR, 7950 XRS, and VSR MD-CLI Command Reference Guide*

Note: This guide generically covers Release 26.*x*.R*x* content and may contain some content that will be released in later maintenance loads. For information about features supported in each load of the Release 26.*x*.R*x* software or for a list of unsupported features by platform and chassis, see the *SR OS R26.x.Rx Software Release Notes*, part number 3HE 29176 000*x* TQZZA.

1.2 Conventions

This section describes the general conventions used in this guide.

1.2.1 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.

1.2.2 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.

Nested substeps within a procedure or a sequential workflow are indicated by roman numerals. 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. At substep b, the user must perform two additional substeps (i. and ii.) to complete the step.

Example: Nested 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. User must perform all nested substeps to complete this action.

i. This is a nested substep.

ii. This is another nested substep.

2 Introduction to multicast

This chapter provides information about multicast.

2.1 Multicast overview

IP multicast provides an effective method of many-to-many communication. With unicast, IP packets are sent from a single source to a single recipient. The source inserts the address of the target host in the IP header destination field of an IP datagram, and intermediate routers (if present) forward the datagram toward the target, in accordance with their respective routing tables.

Sometimes distribution needs individual IP packets be delivered to multiple destinations (such as audio or video streaming broadcasts). Multicast is a method of distributing datagrams sourced from one or more hosts to a set of receivers that may be distributed over different networks. This makes delivery of multicast datagrams significantly more complex.

Multicast sources can send a single copy of data using a single address for the entire group of recipients. The routers between the source and recipients forward the packets using the group address route. Multicast packets are delivered to a multicast group. A multicast group specifies a set of recipients who are interested in a data stream and are represented by an IP address from a specified range. Data addressed to the IP address is forwarded to the members of the group. A source host sends data to a multicast group by specifying the multicast group address in the destination IP address. A source does not have to register to send data to a group nor do they need to be a member of the group.

Routers and Layer 3 switches use IGMP to manage membership for a multicast session. When a host needs to receive one or more multicast session, it sends a join message for each multicast group it needs to join. When a host needs to leave a multicast group, it sends a leave message.

To extend multicast to the Internet, the multicast backbone (Mbone) is used. The Mbone is layered on top of portions of the Internet. These portions, or islands, are interconnected using tunnels. The tunnels allow multicast traffic to pass between the multicast-capable portions of the Internet. As more and more routers in the Internet are multicast-capable, the unicast and multicast routing table converges.

The original Mbone was based on Distance Vector Multicast Routing Protocol (DVMRP) and was very limited. However, the Mbone is converging around the following protocol set:

- IGMP
- Source-Specific Multicast Groups (SSM groups)
- Protocol Independent Multicast - Sparse Mode (PIM-SM)
- Multicast Source Discovery Protocol (MSDP)

2.2 Multicast models

This section describes the models which Nokia routers support to provide multicast.

2.2.1 ASM

Any-Source Multicast (ASM) is the IP multicast service model defined in RFC 1112, *Host Extensions for IP Multicasting*. An IP datagram is transmitted to a host group, a set of zero or more end-hosts identified by a single IP destination address (224.0.0.0 through 239.255.255.255 for IPv4). End-hosts can join and leave the group any time and there is no restriction on their location or number. This model supports multicast groups with arbitrarily many senders. Any end-host can transmit to a host group even if it is not a member of that group.

To combat the vast complexity and scaling issues that ASM represents, the IETF is developing a service model called Source Specific Multicast (SSM).

2.2.2 PIM-SSM

The Source Specific Multicast (SSM) service model defines a channel identified by an (S,G) pair, where S is a source address and G is an SSM destination address. In contrast to the ASM model, SSM only provides network-layer support for one-to-many delivery.

The SSM service model attempts to alleviate the following deployment problems that ASM has presented:

- **address allocation**

  SSM defines channels on a per-source basis. For example, the channel (S1,G) is distinct from the channel (S2,G), where S1 and S2 are source addresses, and G is an SSM destination address. This averts the problem of global allocation of SSM destination addresses and makes each source independently responsible for resolving address collisions for the various channels it creates.

- **access control**

  SSM provides an efficient solution to the access control problem. When a receiver subscribes to an (S,G) channel, it receives data sent only by the source S. In contrast, any host can transmit to an ASM host group. At the same time, when a sender picks an (S,G) channel to transmit on, it is automatically ensured that no other sender transmits on the same channel (except in the case of malicious acts such as address spoofing). This makes it harder to spam an SSM channel than an ASM multicast group.

- **handling of well-known sources**

  SSM requires only source-based forwarding trees, eliminating the need for a shared tree infrastructure. In terms of the IGMP, PIM-SM, MSDP, MBGP protocol suite, this implies that neither the RP-based shared tree infrastructure of PIM-SM nor the MSDP protocol is required. Thus, the complexity of the multicast routing infrastructure for SSM is low, making it viable for immediate deployment. MBGP is still required for distribution of multicast reachability information.

- anticipating that point-to-multipoint applications such as Internet TV will be significant in the future, the SSM model is better suited for such applications.

2.2.3 Multicast in IP-VPN networks

Multicast can be deployed as part of IP-VPN networks. For details on multicast support in IP-VPNs, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 3 Services Guide: IES and VPRN*.

2.2.4 Multicast on PXC ports

On SR OS , multicast protocol support on Port Cross-Connect (PXC) ports is limited to PIMv4/v6 on a LAG with PXC member ports, interconnecting VPRN to EVPN VPWS with SRv6 transport.

See the *7450 ESS, 7750 SR, 7950 XRS, and VSR Interface Configuration Guide* for information about configuring PXC.

3 IGMP

This chapter provides information about IGMP.

3.1 IGMP overview

Internet Group Management Protocol (IGMP) is used by IPv4 hosts and routers to report their IP multicast group memberships to neighboring multicast routers. A multicast router keeps a list of multicast group memberships for each attached network, and a timer for each membership.

Multicast group memberships include at least one member of a multicast group on a specific attached network, not a list of all the members. With respect to each of its attached networks, a multicast router can assume one of two roles, querier or non-querier. There is normally only one querier per physical network.

A querier issues two types of queries, a general query and a group-specific query. General queries are issued to solicit membership information with regard to any multicast group. Group-specific queries are issued when a router receives a leave message from the node it perceives as the last group member remaining on that network segment.

Hosts wanting to receive a multicast session issue a multicast group membership report. These reports must be sent to all multicast enabled routers.

3.1.1 IGMP versions and interoperability requirements

If routers run different versions of IGMP, they negotiate the lowest common version of IGMP that is supported on their subnet and operate in that version.

- **Version 1**
  Specified in RFC 1112, *Host extensions for IP Multicasting*, was the first widely deployed version and the first version to become an Internet standard.
- **Version 2**
  Specified in RFC 2236, *Internet Group Management Protocol, Version 2*, added support for "low leave latency", that is, a reduction in the time it takes for a multicast router to learn that there are no longer any members of a particular group present on an attached network.
- **Version 3**
  Specified in RFC 3376, *Internet Group Management Protocol, Version 3*, adds support for source filtering; that is, the ability for a system to report interest in receiving packets only from specific source addresses, as required to support PIM-SSM, or from all but specific source addresses, sent to a particular multicast address.

  IGMPv3 must keep state per group per attached network. This group state consists of a filter-mode, a list of sources, and various timers. For each attached network running IGMP, a multicast router records the needed reception state for that network.

3.1.2 IGMP version transition

Nokia routers are capable of interpreting with routers and hosts running IGMPv1, IGMPv2, and/or IGMPv3. RFC 5186, *Internet Group Management Protocol Version 3 (IGMPv3)/Multicast Listener Discovery Version 2 (MLDv2) and Multicast Routing Protocol Interaction* describes some of the interoperability issues and how they affect the various routing protocols.

IGMPv3 (RFC 3376) specifies that if a router receives an earlier version query message on an interface, it must immediately switch into a compatibility mode with that earlier version. None of the previous versions of IGMP are source aware. Therefore, if this occurs, and the interface switches to Version 1 or 2 compatibility mode, any previously learned group memberships with specific sources (learned via the IGMPv3 specific include or exclude mechanisms) must be converted to non-source-specific group memberships. The routing protocol treats this as if there is no exclude definition present.

3.1.3 SSM groups

IGMPv3 allows a receiver to join a group and specify that it only wants to receive traffic for a group if that traffic comes from a particular source. If a receiver does this, and no other receiver on the LAN requires all the traffic for the group, then the designated router (DR) can omit performing a (*,G) join to set up the shared tree, and instead issue a source-specific (S,G) join only.

The range of multicast addresses from 232.0.0.0 to 232.255.255.255 is currently set aside for source-specific multicast in IPv4. For groups in this range, receivers should only issue source-specific IGMPv3 joins. If a PIM router receives a non-source-specific join for a group in this range, it should ignore it.

A Nokia PIM router must silently ignore a received (*,G) PIM join message where G is a multicast group address from the multicast address group range that has been explicitly configured for SSM. This occurrence should generate an event. If configured, the IGMPv2 request can be translated into IGMPv3. The router allows for the conversion of an IGMPv2 (*,G) request into a IGMPv3 (S,G) request based on manual entries. A maximum of 32 SSM ranges is supported.

IGMPv3 also allows a receiver to join a group and specify that it only wants to receive traffic for a group if that traffic does not come from a specific source or sources. In this case, the DR performs a (*,G) join as normal, but can combine this with a prune for each of the sources the receiver does not want to receive.

3.1.4 Query messages

The IGMP query source address is configurable at two hierarchal levels. It can be configured globally at each router instance IGMP level and can be configured at individual at the group-interface level. The group-interface level overrides the source IP address configured at the router instance level.

By default, subscribers with IGMP policies send IGMP queries with an all zero SRC IP address (0.0.0.0). However, some systems only accept and process IGMP query messages with non-zero SRC IP addresses. This feature allows the BNG to inter-operate with such systems.

3.2 Configuring IGMP with CLI

This section provides information to configure IGMP using the CLI.

3.2.1 IGMP configuration overview

The routers use IGMP to manage membership for a multicast session. IGMP is not enabled by default. When enabled, at least one interface must be specified in the IGMP context as IGMP is an interface function. Creating an interface enables IGMP. Traffic can only flow away from the router to an IGMP interface and to and from a PIM interface. A router directly connected to a source must have PIM enabled on the interface to that source. The traffic travels in a network from PIM interface to PIM interface and arrives finally on an IGMP enabled interface.

The IGMP CLI context allows you to specify an existing IP interface and modify the interface-specific parameters. Static IGMP group memberships can be configured to test multicast forwarding without a receiver host. When IGMP static group membership is enabled, data is forwarded to an interface without receiving membership reports from host members.

When static IGMP group entries on point-to-point links that connect routers to a rendezvous point (RP) are configured, the static IGMP group entries do not generate join messages toward the RP. When a host wants to receive multicast sessions it sends a join message for each multicast group it wants to join. Then, a leave message may be sent for each multicast group it no longer needs to participate with.

A multicast router keeps a list of multicast group memberships for each attached network, and an interval timer for each membership. Hosts issue a Multicast Group Membership Report when they want to receive a multicast session. The reports are sent to all multicast routers.

3.2.2 Basic IGMP configuration

About this task

Perform the following basic multicast configuration tasks:

Procedure

Step 1. Enable IGMP.

Step 2. Configure IGMP interfaces.

Step 3. Optional: Specify the IGMP version on the interface.

Step 4. Optional: Configure static (S,G)/(*,G).

Step 5. Optional: Configure SSM translation.

3.2.3 Configuring IGMP

3.2.3.1 Enabling IGMP

Use the commands in the following context to enable IGMP.

```
configure router igmp
```

The following example shows IGMP configuration information.

Example: MD-CLI

```
[ex:/configure router "base" igmp]
A:admin@node-2# info
    admin-state enable
    query-interval 125
    query-last-member-interval 1
    query-response-interval 10
    robust-count 2
    ...
```

Example: classic CLI

```
A:node-2>config>router# info
...
#------------------------------------------
echo "IGMP Configuration"
#------------------------------------------
        igmp
            query-interval 125
            query-last-member-interval 1
            query-response-interval 10
            robust-count 2
            no shutdown
        exit
#------------------------------------------
```

3.2.3.2 Configuring an IGMP interface

Use the following command to configure an interface for IGMP. You can reference interfaces configured in the router, IES service, or IES service video-interface context.

```
configure router igmp interface
```

The following example shows interfaces configured for IGMP.

Example: Configure interfaces for IGMP (MD-CLI)

```
[ex:/configure router "2" igmp]
A:admin@node-2# info
    ...
    interface "itf1" {
        admin-state enable
    }
    interface "itf2" {
        admin-state enable
    }
    interface "ip-1.1.1.3" {
        admin-state enable
    }
 ...
```

Example: Configure interfaces for IGMP (classic CLI)

```
A:node-2>config>router>igmp# info
----------------------------------------------
           ...
            interface "itf1"
```

```
                no shutdown
            exit
            interface "itf2"
                no shutdown
            exit
            interface "ip-1.1.1.3"
                no shutdown
            exit
            no shutdown
           ...
-----------------------------------------------
```

Use the commands in the **interface** context to configure the interface for IGMP. The following example shows some IGMP interface configuration options.

Example: Configure IGMP interface options (MD-CLI)

```
[ex:/configure router "2" igmp interface "itf1"]
A:admin@node-2# info
    admin-state enable
    maximum-number-group-sources 5
    maximum-number-sources 10
    version 2
    ...
```

Example: Configure IGMP interface options (classic CLI)

```
A:node-2>config>router>igmp# interface "itf1"
A:node-2>config>router>igmp>if# info
-----------------------------------------------
           version 2
           max-groups 5
           max-sources 10
           no shutdown
           ...
-----------------------------------------------
```

3.2.3.3 Configuring IGMP static multicast

About this task

This task describes how to configure an IGMP static multicast group and add a source IP address or starg entry. Use the commands in the following context to configure an IGMP static multicast group.

```
configure router igmp interface static
```

Procedure

Step 1. Configure an IGMP static multicast group.

```
configure router igmp interface ip-int-name static group ip-address
```

Step 2. Configure a source IP address or a static (*,G) entry for the group.

```
configure router igmp interface ip-int-name static group ip-address source ip-address
configure router igmp interface ip-int-name static group ip-address starg
```

Example

MD-CLI

```
[ex:/configure router "Base" igmp]
A:admin@node-2# info
    ...
    interface "itf1" {
        ...
        static {
            group 239.255.0.2 {
                source 172.22.184.197 { }
            }
        }
    }
    interface "itf2" {
        static {
            group 239.1.1.1 {
                starg
            }
        }
    }
```

Example

classic CLI

```
A:node-2>config>router>igmp# info
----------------------------------------------
        ...
        interface "itf1"
            ...
            static
                group 239.255.0.2
                    source 172.22.184.197
                exit
            exit
        exit
        interface "itf2"
            ...
            static
                group 239.1.1.1
                    starg
                exit
            exit
        exit
----------------------------------------------
```

3.2.3.4 Configuring SSM translation

Procedure

Use the commands in the following context to configure SSM translation for IGMP.

```
configure router igmp ssm-translate
```

The following example shows an SSM translation configuration for IGMP.

Example: MD-CLI

```
[ex:/configure router "Base" igmp]
A:admin@node-2# info
    ...
    ssm-translate {
        group-range start 239.255.0.1 end 239.2.2.2 {
            source 10.1.1.1 { }
        }
    }
    ...
```

Example: classic CLI

```
A:node-2>config>router>igmp# info
----------------------------------------------
        ...
        ssm-translate
            grp-range 239.255.0.1 239.2.2.2
                source 10.1.1.1
            exit
        exit
        ...
----------------------------------------------
```

3.2.4 Disabling IGMP

IGMP is enabled by default. Use the following command to disable IGMP:

- **MD-CLI**

```
configure router igmp admin-state disable
```

- **classic CLI**

```
configure router igmp shutdown
```

4 MLD

This chapter provides information about Multicast Listener Discovery (MLD).

4.1 MLD overview

Multicast Listener Discovery (MLD) is the IPv6 version of IGMP and belongs to the Source Specific Multicast (SSM) service model (see IPv6 PIM models for more information). The purpose of MLD is to allow each IPv6 router to discover the presence of multicast listeners on its directly attached links, and to discover specifically which multicast groups are of interest to those neighboring nodes.

MLD is a sub-protocol of ICMPv6. MLD message types are a subset of the set of ICMPv6 messages, and MLD messages are identified in IPv6 packets by a preceding Next Header value of 58. All MLD messages are sent with a link-local IPv6 source address, a Hop Limit of 1, and an IPv6 Router Alert option in the Hop-by-Hop Options header.

4.1.1 MLDv1

Similar to IGMPv2, MLDv1 reports only include the multicast group addresses that listeners are interested in, and do not include the source addresses. To work with the PIM-SSM model, a similar SSM translation function is required when MLDv1 is used.

SSM translation allows an IGMPv2 device to join an SSM multicast network through the router that provides such a translation capability. Currently SSM translation can done at a box level, but this does not allow a per-interface translation to be specified. SSM translation per interface offers the ability to have a same (*,G) mapped to two different (S,G) on two different interfaces to provide flexibility.

4.1.2 MLDv2

MLDv2 is backward compatible with MLDv1 and adds the ability for a node to report interest in listening to packets with a particular multicast group only from specific source addresses or from all sources except for specific source addresses.

4.2 Configuring MLD with CLI

This section provides information about configuring MLD using the command line interface.

4.2.1 MLD configuration overview

The routers use MLD to manage membership for a multicast session. MLD is not enabled by default. Creating an interface enables MLD. When enabled, at least one interface must be specified in the **configure router mld** context because MLD is an interface function. Traffic can only flow away from the

router to an MLD interface and to and from a PIM interface. A router directly connected to a source must have PIM enabled on the interface to the source. The traffic travels in a network from PIM interface to PIM interface and arrives finally on an MLD-enabled interface.

Use the MLD CLI to specify an existing IP interface and modify the interface-specific command options. Static MLD group memberships can be configured to test multicast forwarding without a receiver host. If MLD static group membership is enabled, data is forwarded to an interface without receiving membership reports from host members.

If static MLD group entries on point-to-point links that connect routers to a rendezvous point (RP) are configured, the static MLD group entries do not generate join messages toward the RP. When a host wants to receive multicast sessions, the host sends a join message for each multicast group it wants to join. A leave message may be sent for each multicast group it no longer needs to participate with.

A multicast router keeps a list of multicast group memberships for each attached network and an interval timer for each membership. Hosts issue a multicast group membership report when they want to receive a multicast session. The reports are sent to all multicast routers.

4.2.2 Basic MLD configuration

Prerequisites

Perform the following basic multicast configuration tasks:

Procedure

Step 1. Enable MLD.

Step 2. Configure MLD interfaces.

Step 3. Optional: Specify the MLD version on the interface.

Step 4. Optional: Configure static (S,G)/(*,G).

Step 5. Optional: Configure SSM translation.

4.2.3 Configuring MLD

4.2.3.1 Enabling MLD

Use the commands in the following context to configure and enable MLD for the router.

```
configure router mld
```

The following example shows a basic configuration with MLD enabled.

Example: MD-CLI

```
[ex:/configure router "Base" mld]
A:admin@node-2# info
    ...
    admin-state enable
    group-if-query-source-address
    query-interval 125
    query-last-member-interval 1
```

```
    query-response-interval 10
    robust-count 2
    ...
```

Example: classic CLI

```
A:node-2>config>router>mld# info
----------------------------------------------
        ...
        no grp-if-query-src-ip
        query-interval 125
        query-last-listener-interval 1
        query-response-interval 10
        robust-count 2
        no shutdown
----------------------------------------------
```

4.2.3.2 Configuring MLD interfaces

Use the commands in the following context to configure MLD interfaces for the router.

```
configure router mld interface
```

The following example shows interfaces configured for MLD.

Example: MD-CLI

```
[ex:/configure router "2" mld]
A:admin@node-2# info
    ...
    interface "lax-sjc" {
        admin-state enable
    }
    interface "lax-vls" {
        admin-state enable
    }
    interface "pl-ix" {
        admin-state enable
    }
 ...
```

Example: classic CLI

```
A:node-2>config>router>mld# info
----------------------------------------------
           ...
            interface "lax-sjc"
                no shutdown
            exit
            interface "lax-vls"
                no shutdown
            exit
            interface "pl-ix"
                no shutdown
            exit
           ...
----------------------------------------------
```

4.2.3.3 Configuring MLD static multicast

About this task

This task describes how to configure a static multicast group with a source IP address or (*,G) entry for an MLD interface.

Procedure

Step 1. Use the following command to configure an MLD static multicast group for the router.

```
configure router mld interface static group grp-ipv6-address
```

Example

Static group configuration (MD-CLI)

```
[ex:/configure router "Base" mld]
A:admin@node-2# info
    ...
    interface "lax-vls" {
        static {
            group ff0e::db8:7 { }
        }
    }
    interface "lax-sjc" {
        static {
            group ff0e::db8:9 { }
        }
    }
    ...
```

Example

Static group configuration (classic CLI)

```
A:node-2>config>router>mld# info
----------------------------------------------
        ...
        interface "lax-vls"
            static
                group ff0e::db8:7
            exit
        exit
        interface "lax-sjc"
            static
                group ff0e::db8:9
            exit
        exit
        ...
----------------------------------------------
```

Step 2. Configure a source or (*,G) entry for the static multicast group.

a. Use the following command to configure a source for the static group.

```
configure router mld interface static group source ipv6-address
```

Example

Static group source configuration (MD-CLI)

```
[ex:/configure router "Base" mld]
A:admin@node-2# info
    ...
     interface "lax-vls" {
        static {
            group ff0e::db8:7 {
                source 2001:db8::1 { }
            }
        }
    }
    ...
```

Example

Static group source configuration (classic CLI)

```
A:node-2>config>router>mld# info
----------------------------------------------
        ...
        interface "lax-vls"
            static
                group ff0e::db8:7
                    source 2001:db8::1
                exit
            exit
        exit
        ...
----------------------------------------------
```

b. Use the following command to configure a (*,G) entry for the static group.

```
configure router mld interface static group starg
```

Example

Static group (*,G) entry configuration (MD-CLI)

```
[ex:/configure router "Base" mld]
A:admin@node-2# info
    ...
    interface "lax-sjc" {
        static {
            group ff0e::db8:9 {
                starg
            }
        }
    }
    ...
```

Example

Static group (*,G) entry configuration (classic CLI)

```
A:node-2>config>router>mld# info
----------------------------------------------
        ...
        interface "lax-sjc"
            static
```

```
                group ff0e::db8:9
                    starg
                exit
            exit
        exit
        ...
-----------------------------------------------
```

4.2.3.4 Configuring SSM translation

Use commands in the following context to configure SSM translation for MLD.

```
configure router mld ssm-translate
```

The following example displays the command usage to configure MLD for the router.

Note: The group range is not created until the source is specified.

Example: MD-CLI

```
[ex:/configure router "Base" mld]
A:admin@node-2# info
    ...
    ssm-translate {
        group-range start ff0e::db8:7 end ff0e::db8:9 {
            source 2001:db8::1 { }
        }
    }
    ...
```

Example: classic CLI

```
A:node-2>config>router>mld# info
-----------------------------------------------
        ...
        ssm-translate
            grp-range ff0e::db8:7 ff0e::db8:9
                source 2001:db8::1
            exit
        exit
        ...
-----------------------------------------------
```

4.2.4 Disabling MLD

MLD is enabled by default. Use the following command to disable MLD:

- **MD-CLI**

```
configure router mld admin-state disable
```

- **classic CLI**

```
configure router mld shutdown
```

5 PIM

This chapter provides information about PIM.

5.1 PIM-SM

PIM-SM leverages the unicast routing protocols that are used to create the unicast routing table, OSPF, IS-IS, BGP, and static routes. Because PIM uses this unicast routing information to perform the multicast forwarding function it is effectively IP protocol independent. Unlike DVMRP, PIM does not send multicast routing tables updates to its neighbors.

PIM-SM uses the unicast routing table to perform the Reverse Path Forwarding (RPF) check function instead of building up a completely independent multicast routing table.

PIM-SM only forwards data to network segments with active receivers that have explicitly requested the multicast group. PIM-SM in the ASM model initially uses a shared tree to distribute information about active sources. Depending on the configuration options, the traffic can remain on the shared tree or switch over to an optimized source distribution tree. As multicast traffic starts to flow down the shared tree, routers along the path determine if there is a better path to the source. If a more direct path exists, then the router closest to the receiver sends a join message toward the source and then reroutes the traffic along this path.

As stated previously, PIM-SM relies on an underlying topology-gathering protocol to populate a routing table with routes. This routing table is called the Multicast Routing Information Base (MRIB). The routes in this table can be taken directly from the unicast routing table, or it can be different and provided by a separate routing protocol such as MBGP. Regardless of how it is created, the primary role of the MRIB in the PIM-SM protocol is to provide the next hop router along a multicast-capable path to each destination subnet. The MRIB is used to determine the next hop neighbor to whom any PIM join/prune message is sent. Data flows along the reverse path of the join messages. Thus, in contrast to the unicast RIB that specifies the next hop that a data packet would take to get to some subnet, the MRIB gives reverse-path information, and indicates the path that a multicast data packet would take from its origin subnet to the router that has the MRIB.

Note: For correct functioning of the PIM protocol, multicast data packets need to be received by the CPM CPU. Therefore, CPM filters and management access filters must be configured to allow forwarding of multicast data packets.

5.1.1 PIM-SM functions

PIM-SM functions have three phases.

5.1.1.1 Phase one

In this phase, a multicast receiver expresses its interest in receiving traffic destined for a multicast group. Typically it does this using IGMP, but other mechanisms may also serve this purpose. One of the local routers of the receiver is elected as the designated router (DR) for that subnet. When the expression of interest is received, the DR sends a PIM join message toward the RP for that multicast group. This join

message is known as a (*,G) join because it joins group G for all sources to that group. The (*,G) join travels hop-by-hop toward the RP for the group, and in each router it passes through, the multicast tree state for group G is instantiated.

Eventually, the (*,G) join either reaches the RP or reaches a router that already has the (*,G) join state for that group. When many receivers join the group, their join messages converge on the RP and form a distribution tree for group G that is rooted at the RP. The distribution tree is called the RP tree or the shared tree (because it is shared by all sources sending to that group). Join messages are re-sent periodically as long as the receiver remains in the group. When all receivers on a leaf network leave the group, the DR sends a PIM (*,G) prune message toward the RP for that multicast group. However, if the prune message is not sent for any reason, the state eventually times out.

A multicast data sender starts sending data destined for a multicast group. The local router of the sender (the DR) takes these data packets, unicast-encapsulates them, and sends them directly to the RP. The RP receives these encapsulated data packets, removes the encapsulation, and forwards them to the shared tree. The packets then follow the (*,G) multicast tree state in the routers on the RP tree, and are replicated wherever the RP tree branches, and eventually reach all the receivers for that multicast group. The process of encapsulating data packets to the RP is called registering, and the encapsulation packets are known as PIM register packets.

At the end of phase one, multicast traffic is flowing encapsulated to the RP, and then natively over the RP tree to the multicast receivers.

5.1.1.2 Phase two

In this phase, register-encapsulation of data packets is performed. However, register-encapsulation of data packets is inefficient for the following reasons:

- Encapsulation and de-encapsulation can be resource-intensive operations for a router to perform, depending on whether the router has appropriate hardware for the tasks.
- Traveling to the RP and then back down the shared tree can cause the packets to travel a relatively long distance to reach receivers that are close to the sender. For some applications, increased latency is unwanted.

Although register-encapsulation can continue indefinitely, for the previous reasons, the RP normally switches to native forwarding. To do this, when the RP receives a register-encapsulated data packet from source S on group G, it normally initiates an (S,G) source-specific join toward S. This join message travels hop-by-hop toward S, instantiating an (S,G) multicast tree state in the routers along the path.

The (S,G) multicast tree state is used only to forward packets for group G if those packets come from source S. Eventually, the join message reaches the S subnet or a router that already has the (S,G) multicast tree state, and packets from S start to flow following the (S,G) tree state toward the RP. These data packets can also reach routers with a (*,G) state along the path toward the RP, and if this occurs, they take a shortcut to the RP tree at this point.

While the RP is in the process of joining the source-specific tree for S, the data packets continue being encapsulated to the RP. When packets from S also start to arrive natively at the RP, the RP receives two copies of each of these packets. At this point, the RP starts to discard the encapsulated copy of these packets and sends a register-stop message back to the DR of S to prevent the DR from unnecessarily encapsulating the packets. At the end of phase two, traffic is flowing natively from S along a source-specific tree to the RP and from there along the shared tree to the receivers. Where the two trees intersect, traffic can transfer from the shared RP tree to the shorter source tree.

Note: A sender can start sending before or after a receiver joins the group, therefore phase two may occur before the shared tree to the receiver is built.

5.1.1.3 Phase three

In this phase, the RP joins back toward the source using the shortest path tree (SPT). Although having the RP join back toward the source removes the encapsulation overhead, it does not completely optimize the forwarding paths. For many receivers, the route via the RP can involve a significant detour when compared with the shortest path from the source to the receiver.

To obtain lower latencies, a router on the LAN of the receiver, typically the DR, may optionally initiate a transfer from the shared tree to a source-specific SPT. To do this, it issues an (S,G) join toward S. This instantiates the (S,G) state in the routers along the path to S. Eventually, this join either reaches the S subnet or reaches a router that already has the (S,G) state. When this happens, data packets from S flow following the (S,G) state until they reach the receiver.

At this point, the receiver (or a router upstream of the receiver) is receiving two copies of the data—one from the SPT, and one from the RP tree. When the first traffic starts to arrive from the SPT, the DR or upstream router starts to drop the packets for G from S that arrive via the RP tree. In addition, it sends an (S,G) prune message toward the RP. The prune message travels hop-by-hop, instantiating an (S,G) state along the path toward the RP, indicating that traffic from S for G should not be forwarded in this direction. The prune message is propagated until it reaches the RP or a router that still needs the traffic from S for other receivers.

By now, the receiver is receiving traffic from S along the SPT between the receiver and S. In addition, the RP is receiving the traffic from S, but this traffic is no longer reaching the receiver along the RP tree. As far as the receiver is concerned, this is the final distribution tree.

5.1.2 Encapsulating data packets in the register tunnel

Conceptually, the register tunnel is an interface with a smaller MTU than the underlying IP interface toward the RP. IP fragmentation on packets forwarded on the register tunnel is performed based on this smaller MTU. The encapsulating DR can perform path-MTU discovery to the RP to determine the effective MTU of the tunnel. This smaller MTU takes both the outer IP header and the PIM register header overhead into consideration.

5.1.3 PIM bootstrap router mechanism

For correct operation, every PIM-SM router within a PIM domain must be able to map a particular global-scope multicast group address to the same RP. If this is not possible, black holes can appear (this is where some receivers in the domain cannot receive some groups). A domain in this context is a contiguous set of routers that all implement PIM and are configured to operate within a common boundary.

The Bootstrap Router (BSR) mechanism provides a way in which viable group-to-RP mappings can be created and distributed to all the PIM-SM routers in a domain. Each candidate BSR originates Bootstrap Messages (BSMs). Every BSM contains a BSR priority field. Routers within the domain flood the BSMs throughout the domain. A candidate BSR that hears about a higher-priority candidate BSR suppresses sending more BSMs for a period of time. The single remaining candidate BSR becomes the elected BSR, and its BSMs inform the other routers in the domain that it is the elected BSR.

The PIM bootstrap routing mechanism is adaptive, meaning that if an RP becomes unreachable, the event is detected and the mapping tables are modified so that the unreachable RP is no longer used, and the new tables are rapidly distributed throughout the domain.

5.1.4 PIM-SM routing policies

Multicast traffic can be restricted from specific source addresses by creating routing policies. Join messages can be filtered using import filters. PIM join policies can be used to reduce denial of service attacks and subsequent PIM state explosion in the router and to remove unwanted multicast streams at the edge of the network before it is carried across the core.

Use the commands in the following context to configure route policies:

- **MD-CLI**

```
configure policy-options
```

- **classic CLI**

```
configure router policy-options
```

Join and register route policy match criteria for PIM-SM can specify the following:

- router interface or interfaces specified by name or IP address
- neighbor address (the source address in the IP header of the join and prune message)
- multicast group address embedded in the join and prune message
- multicast source address embedded in the join and prune message

Join policies can be used to filter PIM join messages so no (*,G) or (S,G) state is created on the router.

The following table lists the join filter policy match conditions.

Table 1: Join filter policy match conditions

Match condition	Matches
Interface	RTR interface by name
Neighbor	The neighbors source address in the IP header
Group Address	Multicast Group address in the join/prune message
Source Address	Source address in the join/prune message

PIM register message are sent by the first hop designated router that has a direct connection to the source. This serves a dual purpose:

- notifies the RP that a source has active data for the group
- delivers the multicast stream in register encapsulation to the RP and its potential receivers
- if no one has joined the group at the RP, the RP ignores the registers

In an environment where the sources to particular multicast groups are always known, it is possible to apply register filters at the RP to prevent any unwanted sources from transmitting multicast stream. You

can apply these filters at the edge so that register data does not travel unnecessarily over the network toward the RP.

The following table lists the register filter policy match conditions.

Table 2: Register filter policy match conditions

Match condition	Matches
Interface	RTR interface by name
Group Address	Multicast Group address in the join/prune message
Source Address	Source address in the join/prune message

5.1.5 RPF checks

Multicast implements a Reverse Path Forwarding (RPF) check. RPF checks the path that multicast packets take between their sources and the destinations to prevent loops. Multicast requires that an incoming interface be the outgoing interface used by unicast routing to reach the source of the multicast packet. RPF forwards a multicast packet only if it is received on an interface that is used by the router to route to the source.

If the forwarding paths are modified because of routing topology changes, any dynamic filters that may have been applied must be re-evaluated. If filters are removed, the associated alarms are also cleared.

5.1.6 Anycast RP for PIM-SM

The implementation of anycast RP for PIM-SM environments enables fast convergence if a PIM RP router fails by allowing receivers and sources to rendezvous at the closest RP. It allows an arbitrary number of RPs per group in a single shared tree PIM-SM domain. This is particularly important for triple play configurations that choose to distribute multicast traffic using PIM-SM, not SSM. In this case, RP convergence must be fast enough to avoid the loss of multicast streams that could cause loss-of-TV delivery to the customer.

Anycast RP for PIM-SM environments is supported in the base routing/PIM-SM instance of the service router. This feature is supported in Layer 3-VPRN instances that are configured with PIM.

5.1.6.1 Anycast RP for PIM-SM implementation

The anycast RP for PIM-SM implementation is defined in RFC 4610, *Anycast-RP Using Protocol Independent Multicast (PIM)*, and is similar to that described in RFC 3446, *Anycast Rendezvous Point (RP) Mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)*. The implementation extends the register mechanism in PIM so that anycast RP functionality can be retained without using MSDP.

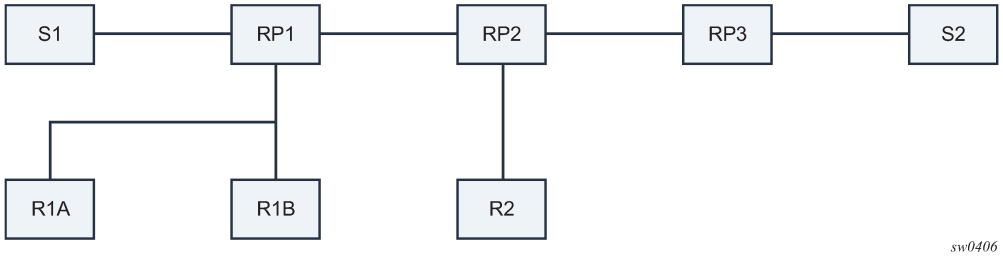
The mechanism works as follows:

- An IP address is chosen as the RP address. This address is statically configured or distributed using a dynamic protocol to all PIM routers throughout the domain.

- A set of routers in the domain are chosen to act as RPs for this RP address. These routers are called the anycast-RP set.
- Each router in the anycast-RP set is configured with a loopback interface using the RP address.
- Each router in the anycast-RP set also needs a separate IP address to be used for communication between the RPs.
- The RP address, or a prefix that covers the RP address, is injected into the unicast routing system inside the domain.
- Each router in the anycast-RP set is configured with the addresses of all other routers in the anycast-RP set. This must be consistently configured for all RPs in the set.

The following figure shows a scenario where all routers are connected, and where R1A, R1B, and R2 are receivers for a group, and S1 and S2 send to that group. In the example, RP1, RP2, and RP3 are all assigned the same IP address that is used as the anycast-RP address (RPA).

Figure 1: Anycast RP for PIM-SM implementation



Note: The address used for the RP address in the domain (the RPA address) must be different from the addresses used by the anycast-RP routers to communicate with each other.

The following procedure is used when S1 starts sourcing traffic.

1. S1 sends a multicast packet.
2. The DR directly attached to S1 forms a PIM register message to send to the RPA. The unicast routing system delivers the PIM register message to the nearest RP, in this case RP1.
3. RP1 receives the PIM register message, de-encapsulates it, and sends the packet down the shared tree to receivers R1A and R1B.
4. RP1 is configured with the IP addresses of RP2 and RP3. Because the register message did not come from one of the RPs in the anycast-RP set, RP1 assumes the packet came from a DR. If the register message is not addressed to the RPA, an error has occurred and it should be rate-limited logged.
5. RP1 sends a copy of the register message from the DR of S1 to both RP2 and RP3. RP1 uses its own IP address as the source address for the PIM register message.
6. RP1 may join back to the source tree by triggering an (S1,G) join message toward S1; however, RP1 must create an (S1,G) state.
7. RP2 receives the register message from RP1, de-encapsulates it, and also sends the packet down the shared tree to receiver R2.
8. RP2 sends a register-stop message back to RP1. RP2 may wait to send the register-stop message if it decides to join the source tree. RP2 should wait until it has received data from the source on the source tree before sending the register-stop message. If RP2 decides to wait, the register-stop

message is sent when the next register is received. If RP2 decides not to wait, the register-stop message is sent immediately.

9. RP2 may join back to the source tree by triggering an (S1,G) join message toward S1; however, RP2 must create an (S1,G) state.
10. RP3 receives the register message from RP1 and de-encapsulates it, but because there are no receivers joined for the group, it discards the packet.
11. RP3 sends a register-stop message back to RP1.
12. RP3 creates an (S1,G) state so when a receiver joins after S1 starts sending, RP3 can join quickly to the source tree for S1.
13. RP1 processes the register-stop messages from RP2 and RP3. RP1 may cache–on a per-RP/per-(S,G) basis—the receipt of register-stop messages from the RPs in the anycast-RP set. This option is performed to increase the reliability of register message delivery to each RP. When this option is used, subsequent register messages received by RP1 are sent only to the RPs in the anycast-RP set that have not previously sent register-stop messages for the (S,G) entry.
14. RP1 sends a register-stop message back to the DR the next time a register message is received from the DR and, when the option in step 13 is in use, if all RPs in the anycast-RP set have returned register-stop messages for a particular (S,G) route.

The procedure for S2 sending follows these same steps, but it is RP3 that sends a copy of the register originated by the DR of S2 to RP1 and RP2. Therefore, this example shows how sources anywhere in the domain, associated with different RPs, can reach all receivers, also associated with different RPs, in the same domain.

5.1.7 Distributing PIM joins over multiple ECMP paths

The per bandwidth/round robin method is commonly used for multicast load-balancing, but the interface in an ECMP set can also be used for a specific channel to be predictable without knowledge of other channels that use the ECMP set.

Use the following command to distribute PIM joins over multiple ECMP paths based on a hash of S and G:

- **MD-CLI**

```
configure router pim mc-ecmp-hashing
```

- **classic CLI**

```
configure router pim mc-ecmp-hashing-enabled
```

When a link in the ECMP set is removed, multicast streams that use this link are redistributed over the remaining ECMP links using the same hash algorithm. When a link is added to the ECMP set, new joins may be allocated to the new link based on the hash algorithm. Existing multicast streams using the other ECMP links stay on those links until they are pruned, unless the **rebalance** command option is specified.

The default is not enabled, which means that the use of multiple ECMP paths (if enabled at the **configure service vprn** context) is controlled through the existing implementation and the **mc-ecmp-balance** command.

Note: You cannot use the **mc-ecmp-balance** command when MC ECMP hashing is enabled in the same context.

To achieve distribution of streams across the ECMP links, the hashing steps are as follows:

1. For a specific (S,G) get all possible next hops.
2. Sort these next hops based on next hop address.
3. XOR S and G addresses.
4. Hash the XORed address over the number of PIM next hops.
5. Use the hash value obtained in step 4, and set that element in the sorted list that was obtained in step 2 as the preferred next hop
6. If this element is not available or is not a PIM next hop (PIM neighbor), the next available next hop is chosen.

Use the following command to display the PIM status for the router instance.

```
show router 100 pim status
```

The following example displays the PIM status indicating ECMP hashing is disabled.

Output example: PIM status indicating ECMP hashing is disabled

```
===============================================================================
PIM Status ipv4
===============================================================================
Admin State                       : Up
Oper State                        : Up

IPv4 Admin State                  : Up
IPv4 Oper State                   : Up

BSR State                         : Accept Any

Elected BSR
    Address                       : None
    Expiry Time                   : N/A
    Priority                      : N/A
    Hash Mask Length              : 30
    Up Time                       : N/A
    RPF Intf towards E-BSR        : N/A

Candidate BSR
    Admin State                   : Down
    Oper State                    : Down
    Address                       : None
    Priority                      : 0
    Hash Mask Length              : 30

Candidate RP
    Admin State                   : Down
    Oper State                    : Down
    Address                       : 0.0.0.0
    Priority                      : 192
    Holdtime                      : 150

SSM-Default-Range                 : Enabled
SSM-Group-Range
    None

MC-ECMP-Hashing                   : Disabled

Policy                            : None
```

```
RPF Table                         : rtable-u

Non-DR-Attract-Traffic            : Disabled
===============================================================================
```

Use commands in the following context to configure PIM.

```
configure service vprn pim
```

PIM configuration includes:

- group-prefix shortest-path switchover thresholds
- interfaces
- import policies
- MC-ECMP traffic balancing or hash-based multicast balancing over ECMP links
- RP
- SSM group ranges

Example: PIM configuration for a VPRN service (MD-CLI)

```
[ex:/configure service vprn "5" pim]
A:admin@node-2# info
    admin-state enable
    apply-to all
    mc-ecmp-balance false
    mc-ecmp-hashing {
        rebalance true
    }
    rp {
        ipv4 {
            bsr-candidate {
                admin-state disable
            }
            static {
                address 10.3.3.3 {
                    group-prefix 224.0.0.0/4 { }
                }
            }
        }
            rp-candidate {
                admin-state disable
            }
        }
    }
```

Example: PIM configuration for a VPRN service (classic CLI)

```
-----------------------------------------------
A:node-2>config>service>vprn>pim# info
-----------------------------------------------
                apply-to all
                rp
                    static
                        address 10.3.3.3
                            group-prefix 224.0.0.0/4
                        exit
                    exit
```

```
                    bsr-candidate
                        shutdown
                    exit
                    rp-candidate
                        shutdown
                    exit
                exit
                no mc-ecmp-balance
                mc-ecmp-hashing-enabled
                no shutdown
-------------------------------------------
```

Use the following command to show distribution of PIM joins over multiple ECMP paths for the specified router instance.

```
show router 100 pim group
```

Output example: Distribution of PIM joins over multiple ECMP paths

```
===============================================================================
PIM Groups ipv4
===============================================================================
Group Address                          Type     Spt Bit Inc Intf       No.Oifs
   Source Address                         RP
-------------------------------------------------------------------------------
239.1.1.1                              (S,G)    spt     to_C0          1
   172.0.100.33                           10.20.1.6
239.1.1.2                              (S,G)    spt     to_C3          1
   172.0.100.33                           10.20.1.6
239.1.1.3                              (S,G)    spt     to_C2          1
   172.0.100.33                           10.20.1.6
239.1.1.4                              (S,G)    spt     to_C1          1
   172.0.100.33                           10.20.1.6
239.1.1.5                              (S,G)    spt     to_C0          1
   172.0.100.33                           10.20.1.6
239.1.1.6                              (S,G)    spt     to_C3          1
   172.0.100.33                           10.20.1.6

239.2.1.1                              (S,G)    spt     to_C0          1
   172.0.100.33                           10.20.1.6
239.2.1.2                              (S,G)    spt     to_C3          1
   172.0.100.33                           10.20.1.6
239.2.1.3                              (S,G)    spt     to_C2          1
   172.0.100.33                           10.20.1.6
239.2.1.4                              (S,G)    spt     to_C1          1
   172.0.100.33                           10.20.1.6
239.2.1.5                              (S,G)    spt     to_C0          1
   172.0.100.33                           10.20.1.6
239.2.1.6                              (S,G)    spt     to_C3          1
   172.0.100.33                           10.20.1.6

239.3.1.1                              (S,G)    spt     to_C0          1
   172.0.100.33                           10.20.1.6
239.3.1.2                              (S,G)    spt     to_C3          1
   172.0.100.33                           10.20.1.6
239.3.1.3                              (S,G)    spt     to_C2          1
   172.0.100.33                           10.20.1.6
239.3.1.4                              (S,G)    spt     to_C1          1
   172.0.100.33                           10.20.1.6
239.3.1.5                              (S,G)    spt     to_C0          1
   172.0.100.33                           10.20.1.6
239.3.1.6                              (S,G)    spt     to_C3          1
```

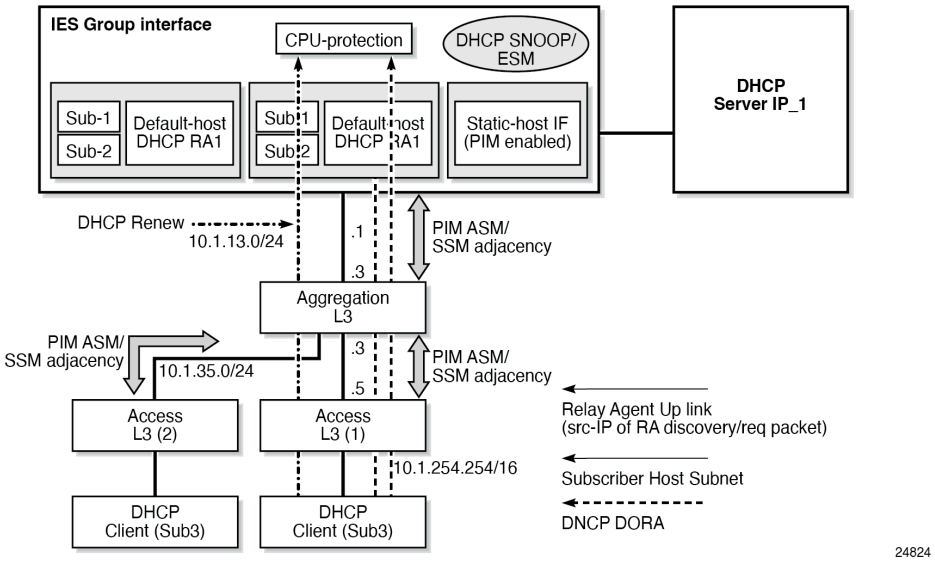
```
   172.0.100.33                              10.20.1.6

239.4.1.1                               (S,G)    spt     to_C0          1
   172.0.100.33                              10.20.1.6
239.4.1.2                               (S,G)    spt     to_C3          1
   172.0.100.33                              10.20.1.6
239.4.1.3                               (S,G)    spt     to_C2          1
   172.0.100.33                              10.20.1.6
239.4.1.4                               (S,G)    spt     to_C1          1
   172.0.100.33                              10.20.1.6
239.4.1.5                               (S,G)    spt     to_C0          1
   172.0.100.33                              10.20.1.6
239.4.1.6                               (S,G)    spt     to_C3          1
   172.0.100.33                              10.20.1.6
-------------------------------------------------------------------------------
Groups : 24
===============================================================================
```

5.1.8 PIM interface on IES subscriber group interfaces

PIM on a subscriber group interface allows for SAP-level replication over an ESM Group interface by establishing PIM adjacency to a downstream router. The following figure depicts the model.

Figure 2: PIM interface on IES subscriber group interface



On an IES subscriber-interface, an Ethernet SAP is configured (LAG or physical port). On the SAP, a static-host is configured for connectivity to downstream Layer 3 aggregation devices (including PIM adjacency) while multiple default-hosts can be configured for subscriber traffic. Single SAP with a single static-host per group interface is supported to establish PIM adjacency on a subscriber group interface. Both IPv4 PIM ASM and SSM are supported.

Feature restrictions:

- Only IPv4 PIM is supported with a single static host used to form a PIM interface under a group interface. Using multiple hosts or non-static hosts is not supported. Configuring IPv6 in the following context is not blocked, but takes no effect.

  ```
  configure router pim interface
  ```

- The following command does not apply to PIM interfaces on IES subscriber group interfaces.

  ```
  configure router pim apply-to
  ```

- PIM on group interfaces is not supported in VPRN context.
- Extranet is not supported.
- Locally attached receivers are not supported (no IGMP or MLD and PIM mix in OIF list).
- Default anti-spoofing must be configured (IP+MAC).
- A subscriber profile with a PIM policy enabled (**configure subscriber-mgmt sub-profile**) cannot combine with the following policies:
  - **host tracking policy**
    This option applies a host tracking policy.
  - **IGMP policy**
    This option applies an IGMP policy.
  - **MLD policy**
    This option applies an MLD policy.
  - **NAT policy**
    This option applies a NAT policy.
  - **Subscriber MCAC policy**
    This option applies a subscriber MCAC policy that can be used when configured in PIM interface context.

5.1.9 MoFRR

With large scale multicast deployments, a link or nodal failure impacts multiple subscribers or a complete region or segment of receivers. This failure interrupts the receiver client experience. Besides the impact on user experience, though multicast client applications may buffer streams for short period of time, the loss of stream data may trigger unicast request for the missing stream data to the source in specific middleware implementations. Those requests can overload the network resources if a traffic loss persists for a prolonged period.

To minimize service interruption to end-users and protect the network from sudden surge of unicast requests, SR OS implements a fast failover scheme for native IP networks. SR OS Multicast-Only Fast Reroute (MoFRR) implementation is based on RFC 7431, *Multicast-Only Fast Reroute*, and relies on the following:

- sending a join to a primary and a single standby upstream nodes over disjoined paths
- fast failover to a standby stream upon detection of a failure

The functionality relies on failure detection on the primary path to switch to forwarding the traffic from the standby path. The traffic failure can happen with or without physical links or nodes going down. Various mechanisms for link or node failure detections are supported; however, to achieve best performance and

resilience, it is recommended to enable MoFRR on every node in the network and use hop-by-hop BFD for fast link failure or data plane failure detection on each upstream link. Without BFD, the PIM adjacency loss or route change could be used to detect traffic failure. Figure 3: MoFRR steady state no failure and Figure 4: MoFRR switch to standby stream on a link failure depict MoFRR behavior.

Figure 3: MoFRR steady state no failure

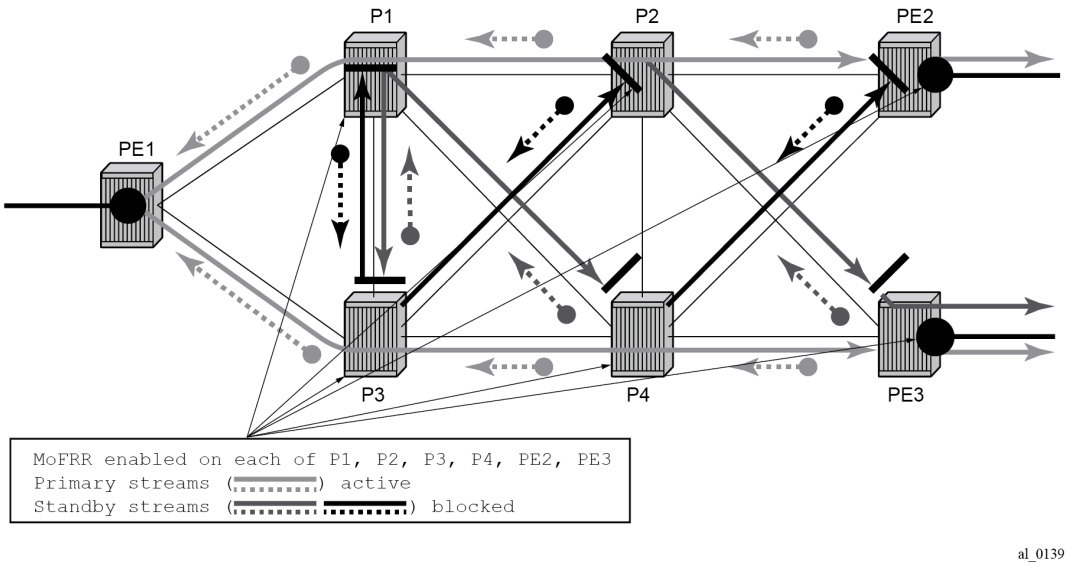
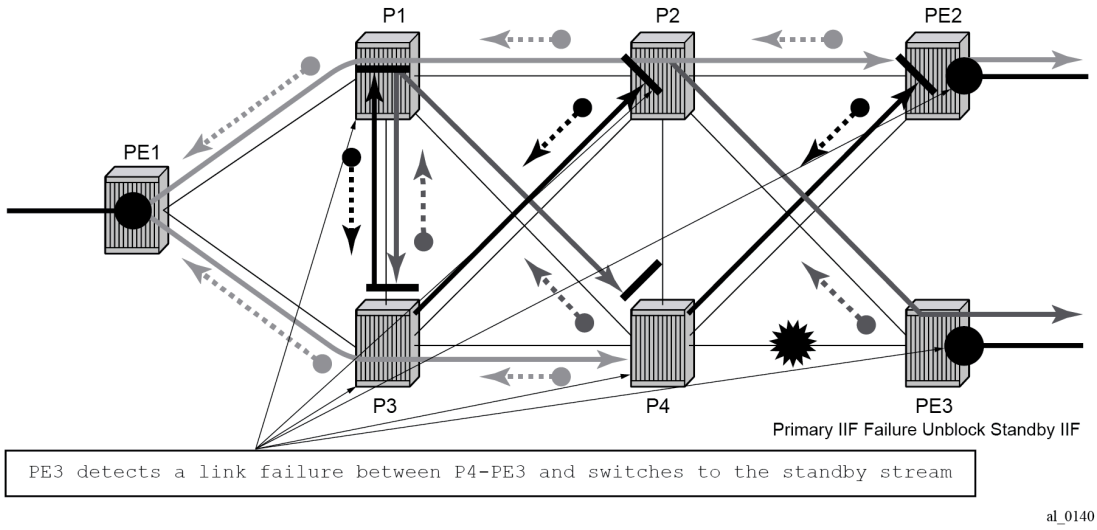


Figure 4: MoFRR switch to standby stream on a link failure



MoFRR functionality supports the following:

- IPv4 or IPv6 link or node failure protection in global routing instance
- Rosen PIM-SSM with MDT SAFI
- active streams and a single standby stream over disjoint ECMP paths

- active streams and a single standby stream joins over IS-IS or OSPF Loop-Free Alternate paths
- all regular PIM interfaces supporting MoFRR for all multicast streams (tunnel interfaces are ignored)

Note: You cannot configure MoFRR in the following contexts indicated, when GTM auto-discovery is enabled in the indicated context:

- **MD-CLI**
  - MoFRR

```
configure router pim ipv4 multicast-fast-failover
configure router pim ipv6 multicast-fast-failover
```

  - GTM auto-discovery

```
configure router pim ipv4 gtm auto-discovery
```

- **classic CLI**
  - MoFRR

```
configure router pim multicast-fast-failover
configure router pim multicast6-fast-failover
```

  - GTM auto-discovery

```
configure router pim gtm auto-discovery
```

5.1.10 Auto-RP

Automatic discovery of group-to-RP mappings (auto-RP) is a proprietary group-discovery and mapping mechanism for IPv4 PIM that is described in cisco-ipmulticast/pim-autorp-spec, *Auto-RP: Automatic discovery of Group-to-RP mappings for IP multicast*. The functionality is similar to the IETF standard bootstrap router (BSR) mechanism, which is described in RFC 5059, *Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)*, to dynamically learn the availability of Rendezvous Points (RPs) in a network.

When it is configured as an RP-mapping agent, the router listens to the CISCO-RP-ANNOUNCE (224.0.1.39) group and caches the announced mappings. The RP-mapping agent then periodically sends out RP-mapping packets to the CISCO-RP-DISCOVERY (224.0.1.40) group. The auto-RP groups use the PIM dense-mode (PIM-DM) to support multihoming and redundancy, as described in RFC 3973, *Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)*. The RP-mapping agent supports announcing, mapping, and discovery functions; candidate RP functionality is not supported. SR OS supports version 1 of the Auto-RP specification; the ability to deny RP-mappings by advertising negative group prefixes is not supported.

Auto-RP is supported for IPv4 in multicast VPNs and in the global routing instance. Either BSR or auto-RP for IPv4 can be configured; the two mechanisms cannot be enabled together. BSR for IPv6 and auto-RP for IPv4 can be enabled together. In a multicast VPN, auto-RP cannot be enabled together with sender-only or receiver-only multicast distribution trees (MDTs), or wildcard S-PMSI configurations that could block flooding.

Use the following command to configure the router as an RP-mapping agent:

- **MD-CLI**

```
configure router pim rp ipv4 auto-rp-discovery
```

- **classic CLI**

```
configure router pim rp auto-rp-discovery
```

5.1.11 VRRP aware PIM

The Virtual Router Redundancy Protocol (VRRP) eliminates the single point of failure inherent in the static default-routed environment. VRRP describes a method of implementing a redundant IP interface that provides dynamic failover if the VRRP master router (MR) becomes unavailable.

VRRP provides information about the state of a router. However, PIM operates independently of VRRP group states. The PIM DR and the VRRP MR may not be the same router and IP multicast traffic may not necessarily follow the same path as elected by VRRP.

To leverage the redundancy capabilities of VRRP that are lacking in PIM, the VRRP Aware PIM mechanism allows PIM to monitor and react to changes in the VRRP MR. This ensures that the multicast traffic follows the unicast traffic through the same gateway as the VRRP MR, providing consistent IP multicast forwarding in a redundant network.

5.1.11.1 Configuring VRRP aware PIM

The VRRP Aware PIM feature enables PIM to track the state of a VRRP instance and to identify whether the associated VRRP interface is the master. PIM uses an operational group option (**oper-group** *group-name*) to monitor the state of VRRP. One operational group can be created for IPv4, and another for IPv6. When VRRP is the MR, the operational group is up; for all other VRRP states, the operational group is down. A VRRP instance can only be associated with one operational group, and an operational group can have one or more associated VRRP instances. This feature is supported on base router, IES, and VPRN interfaces.

If the monitored interface is the VRRP MR, PIM becomes the DR by setting its priority to the configured oper-group active-priority value. For the router to become the DR, the correct priorities must be configured so the active priority of the **oper-group** is the highest priority on the IP interface.

If a PIM router is the DR and then receives an indication from VRRP that the interface is no longer the VRRP MR, PIM relinquishes the DR role by setting its priority back to the default or configured priority value.

If the configured VRRP instance or **oper-group** is not configured, PIM operates as normal with the default or configured priority value. A change in the operational group status is independent of the address family; IPv4 and IPv6 priorities are configured independently of each other. Two operational groups are supported per PIM interface, one for IPv4 and one for IPv6.

5.1.11.2 Guidelines for configuring VRRP Aware PIM

When configuring VRRP Aware PIM, consider the following recommendations:

- Configure VRRP to use BFD to speed up failure detection in addition to the functionality provided by VRRP Aware PIM.

- To optimize failover, enable the following command on the primary and secondary routers to make them a hot-standby redundant pair.

```
configure router pim non-dr-attract-traffic
```

Note: This configuration ignores the DR state and attracts traffic to populate the router's PIM database. Do not use this configuration if multicast traffic must only follow the VRRP MR.

- Configure the group **up** time on the primary router and the group **down** time on the secondary router to the time needed to repopulate the PIM database; for example, 10 seconds. This allows the primary router to populate its PIM database again before becoming the DR and recover from the secondary back to the primary router if a failure occurs from the primary to secondary router. Use the following commands to configure the **up** and **down** times.

```
configure service oper-group hold-time group up
configure service oper-group hold-time group down
```

  Configure the **up** time on the secondary router to 0, so that it assumes the DR role immediately if the primary router fails. The **up** hold time is set to 4 seconds by default, which delays the DR change unnecessarily.
- Sticky DR enables the secondary router to continue to act as the DR after the primary router comes back up. Sticky DR is incompatible with the VRRP Aware PIM mechanism that tracks the VRRP MR. You should disable it if it is configured with the following command.

```
configure router pim interface sticky-dr
```

The following example shows a basic configuration for VRRP Aware PIM.

Example: MD-CLI

```
[ex:/configure service]
A:admin@node-2# info
    interface "to-lan" {
        ipv4 {
            vrrp 1 {
                oper-group "VAwP1"
            }
        }
    }
    interface "to-lan2" {
        ipv4 {
            vrrp 1 {
                oper-group "VAwP2"
            }
        }
    }
    oper-group "VAwP1" {
    }
    oper-group "VAwP2" (
    }
    vprn "1" {
        customer "1"
        pim {
            interface "to-lan" {
                ipv4 {
                    monitor-oper-group {
                        name "VAwP1"
```

```
                        operation add
                        priority-delta 90
                    }
                }
                ipv6 {
                    monitor-oper-group {
                        name "VAwP1"
                        operation add
                        priority-delta 90
                    }
                }
            }
            interface "to-lan2" {
                ipv4 {
                    monitor-oper-group {
                        name "VAwP2"
                        operation add
                        priority-delta 90
                    }
                }
                ipv6 {
                    monitor-oper-group {
                        name "VAwP2"
                        operation add
                        priority-delta 90
                    }
                }
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config>service# info
----------------------------------------------
    oper-group “VAwP1” create
    oper-group “VAwP2” create
    exit
    vprn 1 customer 1 create
        shutdown
        interface to-lan
            vrrp 1 create
                oper-group “VAwP1”
            exit
         exit
        interface to-lan2
            vrrp 1 create
                oper-group “VAwP2”
            exit
         exit
         pim
            interface to-lan
                monitor-oper-group “VAwP1” family ipv4 add 90
                monitor-oper-group “VAwP1” family ipv6 add 90
            exit
            interface to-lan2
                monitor-oper-group “VAwP2” family ipv4 add 90
                monitor-oper-group “VAwP2” family ipv6 set 90
            exit
      exit
exit
```

5.2 IPv6 PIM models

IPv6 multicast enables multicast applications over native IPv6 networks. There are two service models: Any Source Multicast (ASM) and Source Specific Multicast (SSM) which includes PIM-SSM and MLD (see MLD overview). SSM does not require source discovery and only supports single source for a specific multicast stream. As a result, SSM is easier to operate in a large scale deployment that uses the one-to-many service model.

5.2.1 PIM SSM

The IPv6 address family for SSM model is supported. This includes the ability to choose which RTM table to use (unicast RTM, multicast RTM, or both). OSPF3, IS-IS and static-route have extensions to support submission of routes into the IPv6 multicast RTM.

5.2.1.1 System PIM SSM scaling

PIM SSM scaling can be increased to 256k (S,G)s using the **pim-ssm-scaling** command. This command enables (S,G) scaling for PIM SSM in the global routing table only. The current scaling limitation of (S,G)s per complex (FP) still exist. However, the 256K (S,G)s can be configured over multiple complex to achieve this higher scaling.

When PIM SSM scaling is enabled, the following multicast features are disabled:

- DM
- MoFRR
- JP policy
- SSM groups
- (S,G) programming is a maximum of 32000 per complex
- InBand features (BIER and MLDP)
- Extranet
- ASM

This feature is only supported on CPM5s.

When the **pim-ssm-scaling** command is enabled and there is a mix of FP3, FP4, and FP5 cards in the system, Nokia recommends that you configure the following command with the **dynamic** option to ensure the system dynamically chooses the lowest denominator throughput card as multicast-plane throughput.

- **MD-CLI**

```
configure multicast-management chassis-level per-mcast-plane-capacity total-capacity
 dynamic
```

- **classic CLI**

```
configure mcast-management chassis-level per-mcast-plane-capacity total-capacity dynamic
```

Note: When PIM SSM scaling is enabled with IMPM, and different generations of FP are provisioned in the system, Nokia recommends that the multicast-management chassis per-plane total capacity is left at its default value of **dynamic**.

To achieve fast failover when PIM SSM scaling is enabled, the default MCID is used which results in the multicast traffic being sent to all line cards and silently discarded where there is no receiver for that traffic. Consequently, the maximum achievable plane capacity for this traffic is constrained to that of the lowest performance FP. When the maximum link capacity from the fabric to the lowest-performance FP is reached, the link to that FP is overloaded causing the fabric to back-pressure the ingress and resulting in packet loss for all FPs. By using the default MCID, this capacity constraint is independent of whether the lowest-performance FP has a receiver on it or not.

If the multicast management chassis per-plane total capacity is configured to an explicit value which is larger than that supported by the lowest-performance FP, IMPM believes there is more plane capacity available than there really is and the result is (S,G) packet loss instead of blackholing.

By setting the multicast management chassis per-plane total capacity to **dynamic**, the system automatically sets the switch fabric multicast plane capacity to the minimum value supported by the fabric and all line cards in the system. IMPM then has the correct view of the available plane capacity and correctly blackholes (S,G)s when insufficient plane capacity is available. The total maximum multicast capacity is still constrained by the lowest-performance FP.

5.2.2 PIM ASM

IPv6 PIM ASM is supported. All PIM ASM related functions such as bootstrap router, RP, and so on, support both IPv4 and IPv6 address-families. Use the following command to configure IPv6.

```
configure router pim ipv6
```

5.2.3 Embedded RP

The detailed protocol specification is defined in RFC 3956, *Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*. This RFC describes a multicast address allocation policy in which the address of the RP is encoded in the IPv6 multicast group address, and specifies a PIM-SM group-to-RP mapping to use the encoding, leveraging, and extending unicast-prefix-based addressing. This mechanism not only provides a simple solution for IPv6 inter-domain ASM but can be used as a simple solution for IPv6 intra-domain ASM with scoped multicast addresses as well. It can also be used as an automatic RP discovery mechanism in those deployment scenarios that would have previously used the Bootstrap Router protocol (BSR).

5.3 PIM signaling over BIER

PIM signaling over BIER provides a mechanism to signal PIM join and prune messages through a BIER domain with minimal disruption to the PIM domain routers. The Ingress BIER Boundary Routers (IBBRs) terminate PIM and forward the PIM joins and prunes through the BIER domain to egress at the Egress BIER Boundary Router (EBBR). The EBBR is the closest BIER router to the source.

The PIM signaling messages arriving at the IBBR are encapsulated in a BIER header and forwarded to the EBBR. The EBBR tracks every IBBR that is interested in a specific (S,G). The EBBR forwards the join and

prune messages to the PIM domain to which it is attached. When the source receives a join message, it starts the multicast flow for this (S,G). When these PDUs arrive on the EBBR, the EBBR notes all IBBRs that are interested in the (S,G) and adds the IBBRs' corresponding BIER bits to the BIER header. The EBBR (BFIR) forwards the BIER packets with the PDU as payload to the IBBRs (BFERs), where the BIER header is removed and the packet is forwarded on the interface where the join (S,G) was received.

The following restrictions apply:

- PIM signaling through a BIER domain is only supported on FP4 and FP5.
- PIM signaling through a BIER domain is only supported with SSM and not with ASM.
- PIM signaling through a BIER domain and MLDP-inband and GTM are mutually exclusive.

5.3.1 EBBR discovery

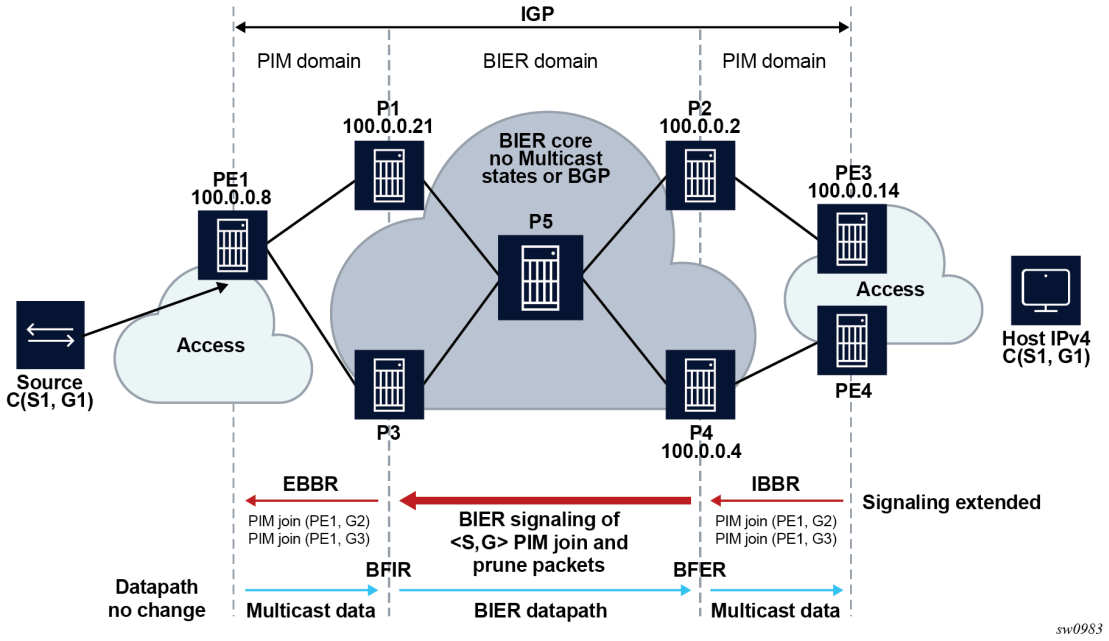
For the EBBR to be discovered, the EBBR must be the source of the route for the multicast source. This means that the EBBR has to generate the multicast source route and advertise it using IGP in one of the following ways:

- **single area**
  - The EBBR is directly connected to the multicast source and therefore is the source of the multicast source route.
  - The EBBR has a static route for the multicast source and redistributes this into IGP. Consequently, it becomes the source of the route for the multicast source.
- **multiple areas**
  - The EBBR is an ABR and re-advertises the multicast source, making the EBBR the source of that route. In this case, the IBBR must be in the same area.
  - The EBBR is an ASBR, and the IBBR has to be in the same AS and area as the EBBR.

5.3.1.1 Single area scenario

Figure 5: Single area PIM signaling over BIER scenario shows a scenario in which the PIM domains and the BIER domain (including the source and the hosts) are all in a single area. The source can be directly connected to the EBBR. In this case, the EBBR is the source of the route for multicast source, or on the EBBR a static route can be configured for the multicast source and this is re-distributed into the IGP.

Figure 5: Single area PIM signaling over BIER scenario



5.3.1.2 Multiple area scenarios

Figure 6: Multiple area PIM signaling over BIER scenario - core area shows a scenario in which the EBBR and IBBR are part of the core (backbone) area.

As in the single area scenario, the IBBR attempts to resolve the source IP address, and the IGP calculates the SPF tree to the source. IGP finds that the source IP address (S1) is generated using ABR routers P1 or P3 because these routers are the source routers for S1.

If the EBBR is not the ABR router, the rules of the single area scenario persists and a static route to the multicast source is required. If the EBBR is an ABR, a static route is not required.

Figure 6: Multiple area PIM signaling over BIER scenario - core area

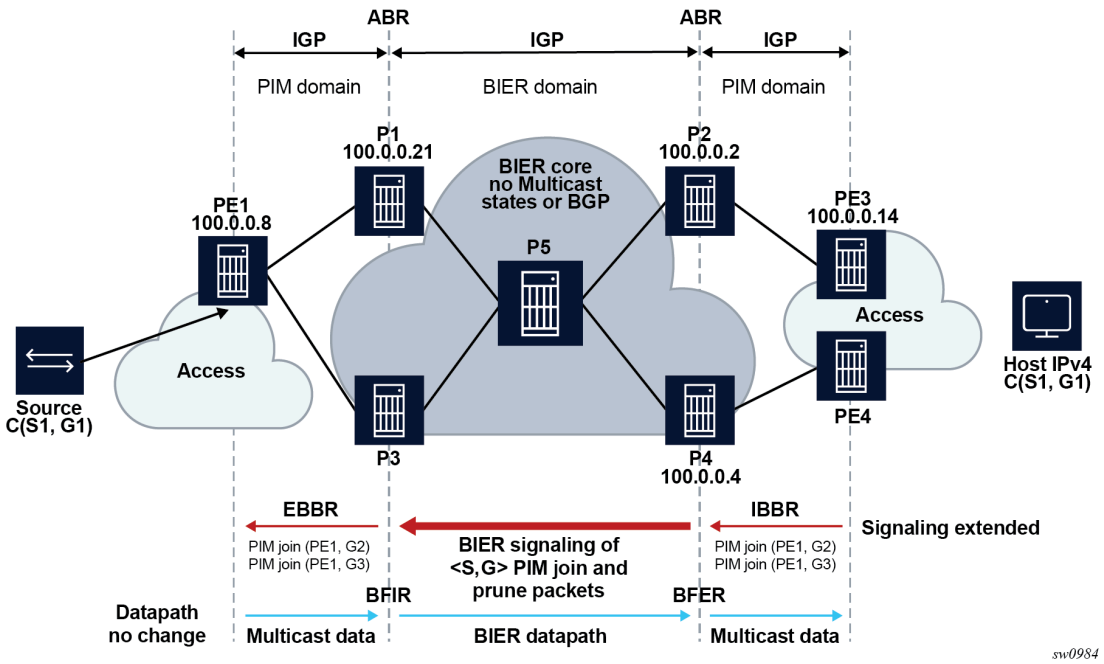
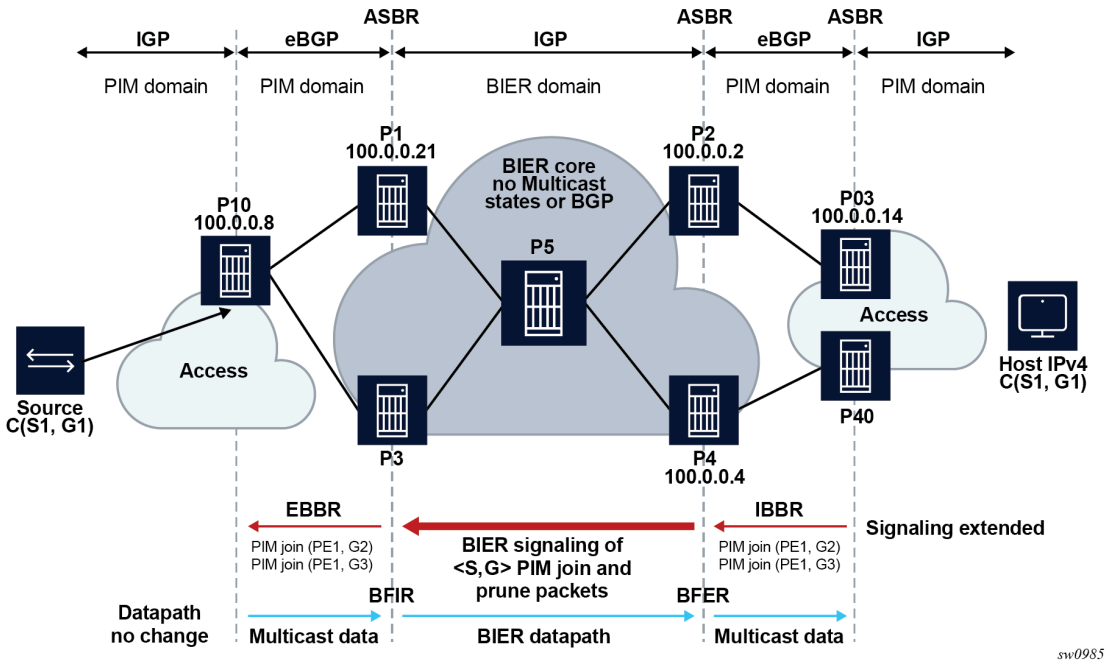


Figure 7: Multiple area PIM signaling over BIER scenario - core AS shows a scenario in which the BIER domain is in the core AS instead of the core area. The core AS consists of a single area.

The EBBR and IBBRs can be any router in the core AS/area. BGP redistributes its routes into the IGP routers P1, P2, P3, and P4.

Figure 7: Multiple area PIM signaling over BIER scenario - core AS



5.3.2 PIM signaling support

PIM signaling is supported only in the SD0.

When a PIM join or prune message arrives on the IBBR, the source is resolved only in the SD0 IGP Multi-instance. For example, if SD0 is assigned to MT IPv4-Unicast the source of the PIM message is resolved only in the MT0.

On the EBBR, only SD0 is supported, so if any PIM signaling messages arrive on the EBBR on any other SD other than SD0 they are not processed.

The PIM join attribute uses attribute type 7.

5.3.3 PIM signaling MTU considerations

The following considerations apply to signaling Rosen MVPN PIM packets over a BIER core.

The MTU of Rosen MVPN tunnels is hardcoded to 1468. For each MVPN, the PIM joins are packed into a single packet up to a size of 1468 bytes. MVPN signaling does not take into account the BIER header size when the Rosen tunnels are signaled over the BIER domain.

Use the following command to reduce the Rosen tunnel MTU to allow a BIER header to be added without exceeding the network MTU, and to reduce the tunnel MTU by the number of bytes specified for the MTU value.

```
configure service vprn pim mtu-over-head
```

5.4 PIM auto-RP full support

The auto-RP protocol consists of announcing, mapping, and discovery functions. The Internet Assigned Numbers Authority (IANA) has assigned two group addresses, 224.0.1.39 and 224.0.1.40, for auto-RP.

The following roles can be configured in auto-RP:

- **candidate RP (CRP)**

  The candidate RP is a router that aims to be an RP for a multicast group using RP-announce messages.

- **mapping agent**

  The mapping agent sees the RP-announce messages and selects the best RP based on the highest IP address for the advertised multicast group. Then, it floods the network with RP-mapping messages about which RPs to use for that group.

- **auto-RP listener**

  The auto-RP listener forwards traffic about the multicast group throughout the network so that the routers that are not directly attached to the mapping agent or to the RP can also learn the RP address. When the listener functionality is enabled, the router uses dense mode only for the two dedicated multicast group addresses.

SR OS supports all auto-RP functionality under GRT and MVPN (NG-MVPN and Rosen MVPN with BGP SAFI).

5.4.1 Candidate RP considerations

The candidate RP address must be set correctly to the system IP or a loopback IP address. The candidate RP **group-range** is optional. If it is not set, the default 224.0.0.0/4 group is advertised.

5.4.2 Timer considerations

By default, the SR OS sends the RP-announce messages every 60 seconds, with the configured **holdtime** for the candidate RP set to 150 seconds. The 60 seconds default time can be modified by changing the RP-candidate **holdtime** ((holdtime)*2)/5).

SR OS sends the RP-mapping messages by default every 60 seconds with the **holdtime** set to 181 seconds. The 60 seconds default time is fixed and cannot be modified.

5.5 Configurable source IP address for PIM register messages

When PIM messages are transmitted over IGP shortcuts, their source IP addresses are selected by choosing the smallest IP address from available interfaces on the node. This can be undesirable because of security measures within the network, such as ACLs, that cause packets to drop. To prevent the messages from being dropped, SR OS supports configuring the source IP address of the register messages to any IP address, regardless of whether it resides on the node.

Use the following commands to configure the source IP address for register messages:

- **MD-CLI**

```
configure router pim ipv4 source-address register-message
configure router pim ipv6 source-address register-message
configure service vprn pim ipv4 source-address register-message
configure service vprn pim ipv6 source-address register-message
```

- **classic CLI**

```
configure router pim source-address register-message
configure service vprn pim source-address register-message
```

5.6 Configuring PIM with CLI

This section provides information to configure PIM using the CLI.

5.6.1 PIM configuration overview

The PIM protocol is not operational until at least one interface is specified for it, at which time the interface is enabled for PIM and is called a PIM interface. When enabled, a PIM interface can be configured with PIM command options in addition to the standard options for the interface when it is created. When PIM is operational, data is forwarded to network segments with active host receivers that have explicitly requested the multicast group.

Note: Before an IP interface can be specified in the PIM context, it must be created. Use one of the following commands to create the IP interface.

```
configure router interface
```

```
configure service ies interface
```

5.6.2 Basic PIM configuration

Perform the following basic PIM configuration tasks:

1. Enable PIM (required)
2. Add interfaces so the protocol establishes adjacencies with the neighboring routers (required)
3. Configure a way to calculate group-to-RP mapping (required) by either:
   - static group-to-RP mapping
   - enabling Candidate RP/Bootstrap mechanism on some routers
4. Enable unicast routing protocols to learn routes toward the RP/source for reverse path forwarding (required)
5. Add SSM ranges (optional)
6. Enable Candidate BSR (optional)
7. Enable Candidate RP (optional)

8. Change hello interval (optional)
9. Configure route policies (bootstrap-export, bootstrap-import, import join and register)

5.6.3 PIM configuration

5.6.3.1 Configuring and enabling PIM

When configuring PIM, make sure to enable PIM on all interfaces for the routing instance, otherwise multicast routing errors can occur.

Use the commands in the following context to configure PIM.

```
configure router pim
```

The following example displays a basic configuration with PIM enabled.

Example: MD-CLI

```
[ex:/configure router "Base" pim]
A:admin@node-2# info
    admin-state enable
    apply-to none
    rp {
        ipv4 {
            bsr-candidate {
                admin-state disable
                address 10.10.10.2
                priority 0
                hash-mask-len 30
            }
            rp-candidate {
                admin-state disable
                holdtime 150
                priority 192
                address 10.10.10.1
            }
            static {
                address 10.10.10.10 {
                }
                address 198.51.100.254 {
                    group-prefix 239.24.24.24/32 {
                }
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config>router# info
#------------------------------------------
echo "PIM Configuration"
#------------------------------------------
        pim
            apply-to none
            rp
                static
                    address 198.51.100.254
```

```
                        group-prefix 239.24.24.24/32
                    exit
                    address 10.10.10.10
                    exit
                exit
                bsr-candidate
                    shutdown
                    address 10.10.10.2
                    priority 0
                    hash-mask-len 30
                exit
                rp-candidate
                     shutdown
                     address 10.10.10.1
                     holdtime 150
                     priority 192
                exit
            exit
            no shutdown
        exit
----------------------------------------
```

5.6.3.2 Configuring PIM interfaces

You can reference router interfaces in the PIM configuration. You must create the interfaces first in the router context. Use the commands in the following context to configure and enable PIM router interfaces.

```
configure router pim interface
```

The following example shows a PIM configuration with basic interfaces configured.

Example: MD-CLI

```
[ex:/configure router "base" pim]
A:admin@node-2# info
    admin-state enable
    apply-to all
    interface "lax-sjc" {
        admin-state enable
    }
    interface "lax-vls" {
        admin-state enable
    }
    interface "pl-ix" {
        admin-state enable
    }
    interface "system" {
        admin-state enable
    }
    rp {
        ipv4 {
            bsr-candidate {
                admin-state enable
                address 10.10.10.10
            }
            rp-candidate {
                admin-state enable
                address 10.10.10.1
            }
            static {
```

```
                address 10.10.10.1 {
                }
                address 198.51.100.254 {
                    group-prefix 239.24.24.24/32 { }
                }
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config>router>pim# info
----------------------------------------------
            interface "system"
                no shutdown
            exit
            interface "lax-sjc"
                no shutdown
            exit
            interface "lax-vls"
                no shutdown
            exit
            interface "pl-ix"
                no shutdown
            exit
            apply-to all
            rp
                static
                    address 10.10.10.1
                    exit
                    address 198.51.100.254
                        group-prefix 239.24.24.24/32
                    exit
                exit
                bsr-candidate
                    address 10.10.10.10
                    no shutdown
                exit
                rp-candidate
                    address 10.10.10.1
                    no shutdown
                exit
            exit
            no shutdown
----------------------------------------------
```

5.6.3.3 Configuring PIM join and register policies

Join policies are used in Protocol Independent Multicast (PIM) configurations to prevent the transportation of multicast traffic across a network and the dropping of packets at a scope at the edge of the network. PIM Join filters reduce the potential for denial of service (DoS) attacks and PIM state explosion—large numbers of Joins forwarded to each router on the RPT, resulting in memory consumption. See the Importing PIM Join/Register Policies section of the Multicast Routing Guide for more information.

(*,G) or (S,G) is the information used to forward unicast or multicast packets. The following options can be configured:

- **group-address**

  This matches the group address policy in join/prune messages group-address "group-address-policy".

- **source-address**

  This is the source-address (192.168.0.1) that matches the source address in join/prune messages.
- **interface**

  This matches any join message received on the specified interface, for example, interface port 1/1/1
- **neighbor**

  This matches any join message received from the specified neighbor; for example, neighbor 1.1.1.1

Use commands in the following context to configure policy options:

- **MD-CLI**

```
configure policy-options
```

- **classic CLI**

```
configure router policy-options
```

The following configuration example does not allow join messages for the specified group address prefix list and source 192.168.0.1 but allows other join messages.

Example: MD-CLI

```
[ex:/configure policy-options]
A:admin@cses-V208# info
    prefix-list "prefix-list-1" {
        prefix 192.0.2.0/24 type exact {
        }
    }
    policy-statement "Foo" {
        entry 10 {
            from {
                group-address "prefix-list-1"
                source-address {
                    ip-address 192.168.0.1
                }
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config>router>policy-options# info
----------------------------------------------
            prefix-list "prefix-list-1"
                prefix 192.0.2.0/24 exact
            exit
            policy-statement "Foo"
                entry 10
                    from
                        group-address "prefix-list-1"
                        source-address 192.168.0.1
                    exit
                exit
            exit
----------------------------------------------
```

5.6.3.4 Importing PIM join and register policies

An import mechanism is provided to control the (*,G) and (S,G) states that are created on the router.

Note: In the import policy, if an action is not specified in the entry then the default-action takes precedence. If no entry matches then the default-action also takes precedence. If no default-action is specified, then the default default-action is executed.

Use the following commands to configure PIM join or register import policies.

```
configure router pim import join-policy
configure router pim import register-policy
```

The following example shows a PIM configuration with an imported policy applied. The policy would not allow join messages for group 229.50.50.208/32 and source 192.168.0.0/16 but would allow join messages for 192.168.0.0/16, 229.50.50.208 (see the "Configuring Route Policy Components" section of the *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide*).

Example: MD-CLI

```
[ex:/configure router "base" pim]
A:admin@node-2# info
    ...
    apply-to-all true
    import join-policy "foo"
    interface "lax-sjc" {
        admin-state enable
    }
    interface "lax-vls" {
        admin-state enable
    }
    interface "pl-ix" {
        admin-state enable
    }
    interface "system" {
        admin-state enable
    }
    rp {
        ipv4 {
            bsr-candidate {
                admin-state enable
                address 10.10.10.10
            }
            rp-candidate {
                admin-state enable
                address 10.10.10.1
            }
            static {
                address 10.10.10.1 {
                }
                address 198.51.100.254 {
                    group-prefix 239.24.24.24/32 { }
                }
            }
        }
    }
    ...
```

Example: classic CLI

```
A:node-2>config>router>pim# info
----------------------------------------------
            ...
            import join-policy "foo"
            interface "system"
            exit
            interface "lax-sjc"
            exit
            interface "lax-vls"
            exit
            interface "pl-ix"
            exit
            apply-to all
            rp
                static
                    address 10.10.10.1
                    exit
                    address 198.51.100.254
                        group-prefix 239.24.24.24/32
                    exit
                exit
                bsr-candidate
                    address 10.10.10.10
                    no shutdown
                exit
                rp-candidate
                    address 10.10.10.1
                    no shutdown
                exit
            exit
            ...
----------------------------------------------
```

5.6.3.5 Configuring bootstrap message import and export policies

Bootstrap import and export policies are used to control the flow of bootstrap messages to and from the RP.

The following configuration example specifies that no BSR messages are received or sent out of interface port 1/1/1.

Example: Configuration of import and export policy statements (MD-CLI)

```
[ex:/configure policy-options]
A:admin@node-2# info
    ...
    prefix-list "pim-policy-1" {
        prefix 10.0.0.0/16 longer
        prefix 10.10.186.0/24 longer
    }
    prefix-list "pim-policy-2"  {
        prefix 10.1.0.0/16 longer
    }
    policy-statement "pim-export-policy" {
        entry 10 {
            to {
                prefix-list "pim-policy-1" "pim-policy-2"
            }
```

```
            action {
                action-type reject
            }
        }
    }
    policy-statement "pim-import-policy" {
        entry 10 {
            from {
                interface ["port1"]
            }
            action {
                action-type reject
            }
        }
    }
    ...
```

Example: Configuration of import and export policy statements (classic CLI)

```
A:node-2>config>router>policy-options# info
----------------------------------------------
        ...
        prefix-list "pim-policy-1"
            prefix 10.0.0.0/16 longer
            prefix 10.10.186.0/24 longer
        exit
        prefix-list "pim-policy-2"  {
            prefix 10.1.0.0/16 longer
        exit
        policy-statement "pim-import-policy"
            entry 10
                from
                    interface "port1"
                exit
                action drop
                exit
            exit
        exit
        policy-statement "pim-export-policy"
            entry 10
                to
                    prefix-list "pim-policy-1" "pim-policy-2"
                exit
                action accept
            exit
        exit
        ...
```

Example: PIM configuration with import and export policies (MD-CLI)

```
[ex:/configure router "Base" pim]
A:node-2# info
    admin-state enable
    apply-to all
    interface "lax-sjc" {
    }
    interface "lax-vls" {
    }
    interface "pl-ix" {
    }
    interface "system" {
    }
```

```
    rp {
        bootstrap {
            import ["pim-import"]
        }
        bootstrap {
            export ["pim-export"]
        }
        ipv4 {
            bsr-candidate {
                admin-state disable
                priority 0
                address 10.10.10.10
                hash-mask-len 30
            }
            rp-candidate {
                admin-state enable
                address 10.10.10.1
            }
            static {
                address 10.10.10.1 {
                }
                address 198.51.100.254 {
                    group-prefix 239.24.24.24/32 { }
                }
            }
        }
    }
```

Example: PIM configuration with import and export policies (classic CLI)

```
A:node-2>config>router>pim# info
----------------------------------------------
            interface "system"
            exit
            interface "lax-sjc"
            exit
            interface "lax-vls"
            exit
            interface "pl-ix"
            exit
            apply-to all
            rp
                bootstrap-import "pim-import"
                bootstrap-export "pim-export"
                static
                    address 10.10.10.1
                    exit
                    address 198.51.100.254
                        group-prefix 239.24.24.24/32
                    exit
                exit
                bsr-candidate
                    shutdown
                    address 10.10.10.10
                exit
                rp-candidate
                    address 10.10.10.1
                    no shutdown
                exit
            exit
            no shutdown
----------------------------------------------
```

5.6.4 Disabling PIM

Use the following commands to disable PIM:

- **MD-CLI**

```
configure router pim admin-state disable
```

- **classic CLI**

```
configure router pim shutdown
```

6 MSDP

MSDP-speaking routers in a PIM-SM domain have MSDP peering relationship with MSDP peers in another domain. The peering relationship is made up of a TCP connection in which control information is exchanged. Each domain has one or more connections to this virtual topology.

When a PIM-SM rendezvous point (RP) learns about a new multicast source within its own domain from a standard PIM register mechanism, it encapsulates the first data packet in an MSDP source-active message and sends it to all MSDP peers.

The source-active message is flooded (after an RPF check) by each peer to its MSDP peers until the source-active message reaches every MSDP router in the interconnected networks. If the receiving MSDP peer is an RP, and the RP has a (*.G) entry (receiver) for the group, the RP creates a state for the source and joins to the shortest path tree for the source. The encapsulated data is de-encapsulated and forwarded down the shared tree of that RP. When the packet is received by the last hop router of the receiver, the last hop router also may join the shortest path tree to the source.

The MSDP speaker periodically sends source-active messages that include all sources.

6.1 Multicast Source Discovery Protocol

MSDP-speaking routers in a PIM-SM domain have MSDP peering relationship with MSDP peers in another domain. The peering relationship is made up of a TCP connection in which control information is exchanged. Each domain has one or more connections to this virtual topology.

When a PIM-SM rendezvous point (RP) learns about a new multicast source within its own domain from a standard PIM register mechanism, it encapsulates the first data packet in an MSDP source-active message and sends it to all MSDP peers.

The source-active message is flooded (after an RPF check) by each peer to its MSDP peers until the source-active message reaches every MSDP router in the interconnected networks. If the receiving MSDP peer is an RP, and the RP has a (*.G) entry (receiver) for the group, the RP creates a state for the source and joins to the shortest path tree for the source. The encapsulated data is de-encapsulated and forwarded down the shared tree of that RP. When the packet is received by the last hop router of the receiver, the last hop router also may join the shortest path tree to the source.

The MSDP speaker periodically sends source-active messages that include all sources.

6.1.1 Anycast RP for MSDP

MSDP is a mechanism that allows RPs to share information about active sources. When RPs in remote domains learn about the active sources, they can pass on that information to the local receivers and multicast data can be forwarded between the domains. MSDP allows each domain to maintain an independent RP that does not rely on other domains but enables RPs to forward traffic between domains. PIM-SM is used to forward the traffic between the multicast domains.

Using PIM-SM, multicast sources and receivers register with their local RP by the closest multicast router. The RP maintains information about the sources and receivers for a specific group. RPs in other domains do not have any knowledge about sources located in other domains.

MSDP is required to provide inter-domain multicast services using Any Source Multicast (ASM). Anycast RP for MSDP enables fast convergence when an MSDP/PIM PR router fails by allowing receivers and sources to rendezvous at the closest RP.

6.1.2 MSDP procedure

When an RP in a PIM-SM domain first learns of a new sender, for example, by PIM register messages, it constructs a source-active (SA) message and sends it to its MSDP peers. The SA message contains the following fields:

- source address of the data source
- group address the data source sends to
- IP address of the RP

Note: An RP that is not a designated router on a shared network does not originate SAs for directly-connected sources on that shared network. It only originates in response to receiving register messages from the designated router.

Each MSDP peer receives and forwards the message away from the RP address in a peer-RPF flooding fashion. The peer-RPF flooding applies to forwarding SA messages. The Multicast Routing Information Base (MRIB) is examined to determine which peer toward the originating RP of the SA message is selected. Such a peer is called an RPF peer.

If the MSDP peer receives the SA message from a non-RPF peer toward the originating RP, it drops the message. Otherwise, it forwards the message to all its MSDP peers (except the one from which it received the SA message).

When an MSDP peer that is also an RP for its own domain receives a new SA message, it determines if any group members within the domain are interested in any group described by an (S,G) entry within the SA message. That is, the RP checks for a (*,G) entry with a non-empty outgoing interface list. This implies that some system in the domain is interested in the group. In this case, the RP triggers an (S,G) join event toward the data source as if a join/prune message addressed to the RP was received. This sets up a branch of the source-tree to this domain. Subsequent data packets arrive at the RP by this tree branch and are forwarded down the shared-tree inside the domain. If leaf routers choose to join the source-tree, they have the option to do so according to existing PIM-SM conventions. If an RP in a domain receives a PIM join message for a new group G, the RP must trigger an (S,G) join event for each active (S,G) for that group in its SA cache.

This procedure is called flood-and-join because if any RP is not interested in the group, the SA message can be ignored; otherwise, they join a distribution tree.

6.1.2.1 MSDP peering scenarios

RFC 4611, *Multicast Source Discovery Protocol (MSDP) Deployment Scenarios*, describes how protocols work together to provide intra- and inter-domain ASM service.

Inter-domain peering:

- peering between PIM border routers (single-hop peering)
- peering between non-border routers (multi-hop peering)
- MSDP peering without BGP

- MSDP peering between mesh groups
- MSDP peering at a multicast exchange

Intra-domain peering:

- peering between routers configured for both MSDP and MBGP
- MSDP peer is not BGP peer (meaning, no BGP peer)

6.1.2.2 Peer-RPF check

Unlike the normal multicast RPF checks, the peer-RPF check stops SA messages from looping. An MSDP router validates SA messages originated from other routers in a deterministic fashion. When the router receives an SA message, it applies a set of rules to validate the SA message, and the first rule that applies determines the peer-RPF neighbor. All SA messages from other routers are rejected. The rules applied to SA messages originating at Router S received at Router R from Router N are as follows:

1. If Router N and Router S are one and the same, the message is originated by a direct peer-RPF neighbor and is accepted.
2. If Router N is a configured peer, or a member of the Router R mesh group, its SA messages are accepted.
3. If Router N is the Border Gateway Protocol (BGP) next hop of the active multicast RPF route toward Router S, Router N is the peer-RPF neighbor, and its SA messages are accepted.
4. If Router N is an external BGP peer of Router R and the last autonomous system (AS) number in the BGP AS-path to Router S is the same as the AS number of Router N, Router N is the peer-RPF neighbor, and its SA messages are accepted.
5. If Router N uses the same next hop as the next hop to Router S, Router N is the peer-RPF neighbor, and its SA messages are accepted.
6. If Router N fits none of the preceding rules, Router N is not a peer-RPF neighbor, and its SA messages are rejected.

When a peer is configured as a default peer, all SA messages received from the peer are accepted without performing the preceding peer-RPF check.

Use the following commands to configure a default peer.

```
configure router msdp peer default-peer
configure router msdp group peer default-peer
```

6.1.3 MSDP peer groups

MSDP peer groups are typically created when multiple peers have a set of common operational parameters. Group parameters not specifically configured are inherited from the global level.

6.1.4 MSDP mesh groups

MSDP mesh groups are used to reduce SA flooding primarily in intra-domain configurations. When a number of speakers in an MSDP domain are fully meshed, they can be configured as a mesh group. The

originator of the SA message forwards the message to all members of the mesh group. Because of this, forwarding the SA between non-originating members of the mesh group is not necessary.

6.1.5 MSDP routing policies

MSDP routing policies allow for filtering of inbound or outbound, or both, SA messages. Policies can be configured at different levels:

- **global level**

  Global level applies to all peers.

- **group level**

  Group level applies to all peers in the peer group.

- **neighbor level**

  Neighbor level applies only to a specified peer.

The most specific level is used. If multiple policy names are specified, the policies are evaluated in the order they are specified. The first policy that matches is applied. If no policy is applied, SA messages are passed.

Match conditions include the following:

- **neighbor** - matches on a neighbor address is the source address in the IP header of the SA message
- **route filter** - matches on a multicast group address embedded in the SA message
- **source address filter** - matches on a multicast source address embedded in the SA message

6.1.6 Multicast in virtual private networks

6.1.6.1 Draft Rosen

RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*, describes a method of providing a VPN service. A VPN provides secure connections to the network, allowing more efficient service to remote users without compromising the security of firewalls. The Rosen draft specifies the protocols and procedures which must be implemented for a service provider to provide a unicast VPN. The draft extends that specification by describing the protocols and procedures that a service provider must implement to support multicast traffic in a VPN, assuming that PIM [PIMv2] is the multicast routing protocol used within the VPN, and the SP network can provide PIM.

IGMP is not supported for receivers or senders directly attached to the PE.

For more information, see the "Virtual Private Routed Network Service" section of the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 3 Services Guide: IES and VPRN*.

6.2 Configuring MSDP with CLI

This section provides information to configure MSDP using the CLI.

6.2.1 Basic MSDP configuration

Perform the following basic MSDP configuration tasks:

- enable MSDP (required)
- configure peer
- configure local address

6.2.2 Configuring MSDP

Use commands in the following context to configure and enable basic MSDP.

```
configure router msdp
```

The following example shows a basic MSDP configuration with default values and the peer and peer local addresses specified.

Note: When you configure a peer for MDSP, the default state is enabled. Only the **info detail** command displays the default enabled state.

Example: MD-CLI

```
[ex:/configure router "Base" msdp]
A:admin@node-2# info
...
    peer 10.20.1.1 {
        local-address 10.20.1.6
    }
...
```

Example: classic CLI

```
A:node-2>config>router>msdp# info
----------------------------------------------
 ...
      peer 10.20.1.1
          local-address 10.20.1.6
      exit
 ...
----------------------------------------------
```

6.2.3 Disabling MSDP

MSDP is enabled by default. Use the following commands to disable MSDP:

- **MD-CLI**

```
configure router msdp admin-state disable
```

- **classic CLI**

```
configure router msdp shutdown
```

The following example shows an MSDP configuration that is disabled.

Example: MD-CLI

```
[ex:/configure router "Base" msdp]
A:admin@node-2# info
    admin-state disable
    group "test" {
        active-source-limit 50000
        export-policy ["LDP-export"]
        import-policy ["LDP-import"]
        local-address 10.10.10.103
        mode mesh-group
        receive-message-rate {
            rate 100
            time 300
            threshold 5000
        }
        peer 10.10.10.104 {
        }
    }
    peer 10.20.1.1 {
        local-address 10.20.1.6
    }
```

Example: classic CLI

```
A:node-2config>router# info
----------------------------------------------
...
#------------------------------------------
echo "MSDP Configuration"
#------------------------------------------------
        msdp
            shutdown
            peer 10.20.1.1
                local-address 10.20.1.6
            exit
            group "test"
                active-source-limit 50000
                receive-msdp-msg-rate 100 interval 300 threshold 5000
                export "LDP-export"
                import "LDP-import"
                local-address 10.10.10.103
                mode mesh-group
                peer 10.10.10.104
                exit
            exit
        exit
```

7 MLDP

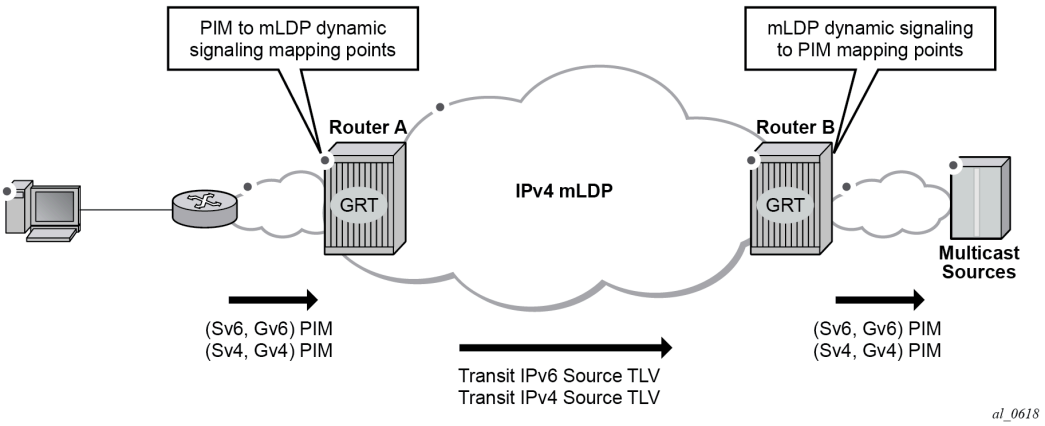
7.1 Dynamic multicast signaling over P2MP in GRT instance

This feature provides a flexible multicast signaling solution to connect native IP multicast source and receivers running PIM multicast protocol via an intermediate MPLS (P2MP LDP LSP) network. The feature allows each native IP multicast flow to be connected via an intermediate P2MP LSP by dynamically mapping each PIM multicast flow to a P2MP LDP LSP.

The feature uses procedures defined in RFC 6826: *Multipoint LDP In-Band Signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths*. On the leaf node of a P2MP LSP, PIM signaling is dynamically mapped to P2MP LDP tree setup. On the root node of P2MP LSP, P2MP LDP signaling is handed back to PIM. Because of dynamic mapping of multicast IP flow to P2MP LSP, provisioning and maintenance overhead is eliminated as multicast destination services are added and removed from the network. Per (S,G) IP multicast state is also removed from the network where P2MP LSPs are used to transport multicast flows.

Figure 8: Dynamic MLDP signaling for IP multicast in GRT illustrates dynamic MLDP signaling for IP multicast in GRT.

Figure 8: Dynamic MLDP signaling for IP multicast in GRT



As illustrated in Figure 8: Dynamic MLDP signaling for IP multicast in GRT, P2MP LDP LSP signaling is initiated from the router that receives PIM JOIN from a downstream router (Router A). To enable dynamic multicast signaling, **p2mp-ldp-tree-join** must be configured on PIM outgoing interface of Router A. This enables handover of multicast tree signaling from PIM to P2MP LDP LSP. Being a leaf node of P2MP LDP LSP, Router A selects the upstream-hop as the root node of P2MP LDP FEC based on routing table lookup. If an ECMP path is available for a specific route, then the number of trees are equally balanced toward multiple root nodes. The PIM Joins are carried in Transit IPv4 (IPv4 PIM-SSM) or IPv6 (IPv6 PIM-SSM) MLDP TLVs. On the root node of P2MP LDP LSP (Router B), multicast tree signaling is handed back to PIM and propagated upstream as native-IP PIM JOIN.

The feature is supported with IPv4 and IPv6 PIM-SSM and IPv4 MLDP. Directly connected IGMP/MLD receivers are also supported with PIM enabled on outgoing interfaces and SSM mapping configured if required.

If multiple criteria exist to set up a multicast flow, the priority is as follows:

1. Multicast (statically provisioned) over P2MP LSP (RSVP-TE or LDP)
2. Dynamic multicast signaling over P2MP LDP
3. PIM native-IP multicast

The following are feature restrictions:

- A single instance of P2MP LDP LSP is supported between the root and leaf nodes per multicast flow; there is no stitching of dynamic trees.
- Extranet functionality is not supported.
- The router LSA link ID or the advertising router ID must be a routable IPv4 address (including IPv6 into IPv4 MLDP use cases).
- IPv6 PIM with dynamic IPv4 MLDP signaling is not supported with e-BGP or i-BGP with IPv6 next-hop.
- Inter-AS and IGP inter-area scenarios where the originating router is altered at the ASBR and ABR respectively, (therefore PIM has no way to create the LDP LSP toward the source), are not supported.

7.2 Inter-AS non-segmented MLDP

This feature allows multicast services to use segmented protocols and span them over multiple autonomous systems (ASs), as done in unicast services. As IP VPN or GRT services span multiple IGP areas or multiple ASs, either for a network designed to deal with scale or as result of commercial acquisitions, users may require Inter-AS VPN (unicast) connectivity. For example, an Inter-AS VPN can break the IGP, MPLS and BGP protocols into access segments and core segments, allowing higher scaling of protocols by segmenting them into their own islands. SR OS also allows for similar provision of multicast services and for spanning these services over multiple IGP areas or multiple ASs.

For multicast VPN (MVPN), SR OS previously supported Inter-AS option A/B/C for Rosen MVPN; however, when MPLS was used, only option A was supported for Next Generation Multicast VPN (NG-MVPN) or d-MLDP signaling. MLDP now supports non-segmented MLDP trees for inter-AS solutions, applicable for multicast services in the GRT (Global Routing Table) where they need to ride over MLDP point-to-multipoint tunnels as well as NG-MVPN services.

See the "ECMP Support" subsection of the "Inter-AS Non-segmented MLDP" section in the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide* for more information.

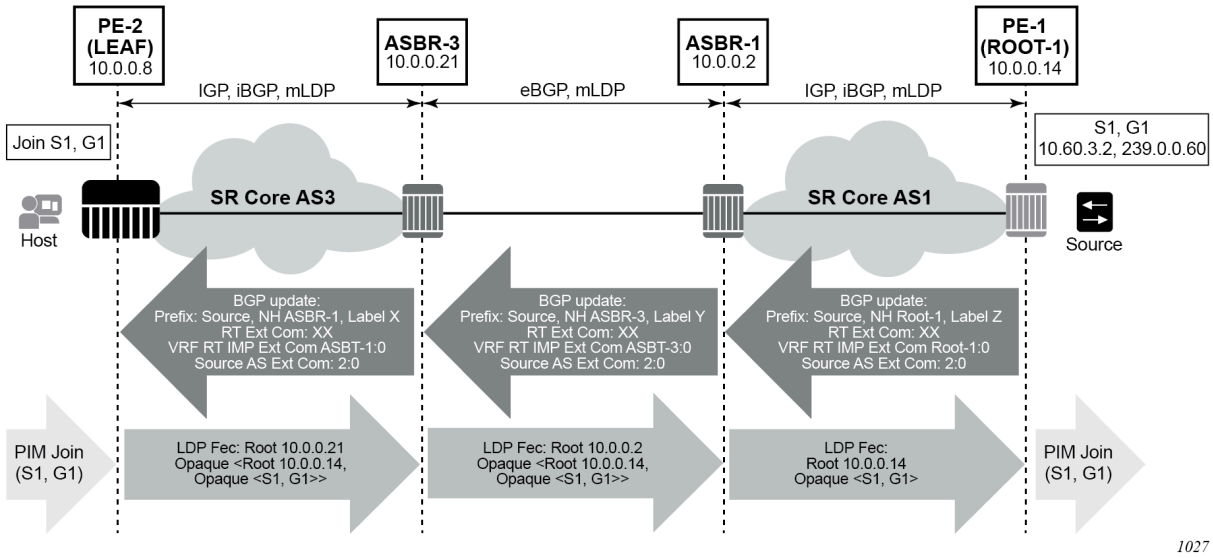
See the "Dynamic mLDP and Static mLDP Co-existing on Same Node" section in the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide* for more information.

7.2.1 d-MLDP inter-AS trees in GRT

Figure 9: In-band signaling with non-segmented inter-AS MLDP trees in GRT shows the processing required for d-mLDP with non-segmented mLDP Inter-AS trees in GRT (routers in AS3, including ASBR 1, have no route to ROOT-1 in IGP and must use BGP unicast routes to resolve route to ROOT-1 and to multicast source).

PE-1 (ROOT-1) is the root node of the MLDP tree, and PE-2 (LEAF) is the leaf node.

Figure 9: In-band signaling with non-segmented inter-AS MLDP trees in GRT



7.2.1.1 Routing

BGP unicast routes must advertise to the VRF Route Import Ext Community, identifying the root PE, for the feature to operate properly. Failure to do so results in PIM Inter-AS joins being dropped.

The community is an address-based community where the global administrator field is the address of the root PE and local administrator field is set to 0 (GRT). No new configuration is required; however, a user must enable inter-AS VPN and configure export policy to ensure the community is added to the BGP routes as required. The BGP unicast route is propagated across the AS, as shown in Figure 9: In-band signaling with non-segmented inter-AS MLDP trees in GRT (the same processing, not shown, applies to a BGP route specifying address used to build mLDP tree rooted at ROOT-1). The following configuration example shows an export policy configuration.

Static routes must be configured on inter-ASBR LDP-enabled links because the BGP peer uses a host address from the local subnet of the links (for GRT and VPN option C), or the BGP peer uses a system IP address that is not in the base routing table (for VPN option B).

- For system-IP to system-IP, static-routes are required for bringing up the EBGP/LDP session.
- If the link IP is used for creation of EBGP and ILD, then static-routes are not required; however, static-route (host-route) is mandatory on ASBR2 for the resolution of MLDP FEC, as the link LSR ID is not resolved by LDP using a /24 route; it needs a /32 route.

Example: MD-CLI

```
[ex:/configure]
A:admin@cses-V208# info
    ...
    policy-options {
        community "A" {
            member "target:1.1.1.1:0" { }
        }
        community "B" {
            member "ext:010b:0a1401060000" { }
```

```
        }
        policy-statement "accept_all" {
            default-action {
                action-type accept
            }
        }
        policy-statement "fromlocal" {
            entry 10 {
                from {
                    protocol {
                        name [direct]
                    }
                }
                to {
                    protocol {
                        name [bgp]
                    }
                }
                action {
                    action-type accept
                    community {
                        add ["A" "B"]
                    }
                }
            }
            default-action {
                action-type reject
            }
        }
    ...
        bgp {
            inter-as-vpn true
            export {
                policy ["fromlocal" "accept_all"]
            }
        }
    ...
```

Example: classic CLI

```
A:node-2>config>router# info
----------------------------------------------
    ...
#--------------------------------------------------
echo "Policy Configuration"
#--------------------------------------------------
        policy-options
            begin
            community "A"
                members "target:1.1.1.1:0"
            exit
            community "B"
                members "ext:010b:0a1401060000"
            exit
            policy-statement "fromlocal"
                entry 10
                    from
                        protocol direct
                    exit
                    to
                        protocol bgp
                    exit
                    action accept
```

```
                        community add "A" "B"
                    exit
                exit
                default-action drop
                exit
            exit
            policy-statement "accept_all"
                default-action accept
                exit
            exit
            policy-statement "pim-import"
                entry 10
                    from
                        interface "port1"
                    exit
                    action drop
                    exit
                exit
            exit
            commit
        exit

#--------------------------------------------------
echo "BGP Configuration"
#--------------------------------------------------
        bgp
            enable-inter-as-vpn
            export "fromlocal" "accept_all"
            no shutdown
        exit
-------------------------------------------
    ...
```

7.2.1.2 Join processing

To traverse an inter-AS domain, recursive FECs are required (see the "Inter-AS Non-segmented MLDP" section in the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide* for more information).

Use the following commands to enable dynamic signaling on interfaces where Inter-AS joins are expected to be received using existing configurations.

```
configure router pim interface p2mp-ldp-tree-join ipv4
configure router pim interface p2mp-ldp-tree-join ipv6
```

When enabled, the following describes the required processing of a PIM join, as shown in Figure 9: In-band signaling with non-segmented inter-AS MLDP trees in GRT.

When the leaf receives a PIM join for group (S1, G1) and, through configuration, knows dynamic signaling is required, the leaf fails to resolve the source S1 via IGP and attempts to resolve route via BGP. The leaf learns that source is reachable via Next-Hop ASBR3 and the route was advertised by PE1 (Root-1) (from VRF Import Ext Community). PE2 (leaf) sources a Recursive mLDP FEC with a root node of ASBR3, and an opaque value containing the MLDP in-band signaling information identifying the (S1, G1) group and the Root-1 (the root of the inter-AS non-segmented MLDP tree), as shown in the following.

```
LEAF FEC {Root = ASBR3, Opaque Value = {Root: ROOT-1, Opaque Value (S1, G1)}}
```

The FEC is forwarded using IGP to ASBR3. When the Recursive MLDP FEC arrives at ASBR3, it notes that it is the identified root node in the local AS, and that the opaque value is a Recursive Opaque Value.

Because ASBR3 fails to resolve the Recursive FEC's root (Root-1) in IGP, ASBR3 attempts to resolves the root via BGP. Similar to processing on LEAF, this yields a Next-Hop of ASBR1. ASBR3 creates a new mLDP FEC element with a root node of ASBR1, and the Opaque value as per received opaque value, as shown in the following.

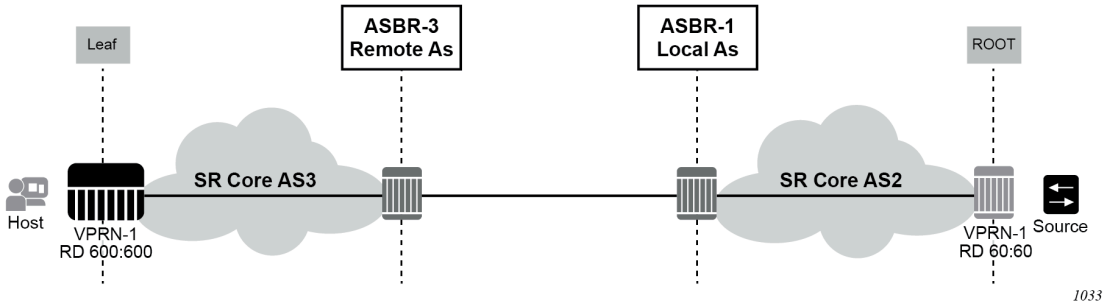
```
ASBR3 FEC {Root = ASBR1, Opaque Value = {Root: Root-1, Opaque Value (S1, G1)}}
```

ASBR 3 forwards the FEC using IGP or EBGP. When the MLDP FEC arrives at ASBR1, it notes that it is the identified root node, and that the opaque value is a Recursive Opaque value. Because ASBR1 can resolve the Recursive FEC's root (Root-1) via IGP, no further recursive processing is required. ASBR 1 forwards mLDP FEC containing in-band signaling using IGP toward ROOT-1.

7.2.2 ASBR support of PE functionality

Figure 10: Remote and local ASBRs displays remote and local ASBRs.

Figure 10: Remote and local ASBRs



While ASBRs can also act as PE nodes, SR OS does not support all PE functionalities in the ASBR node. Table 3: PE features on ASBRs lists supported PE features on ASBRs.

Table 3: PE features on ASBRs

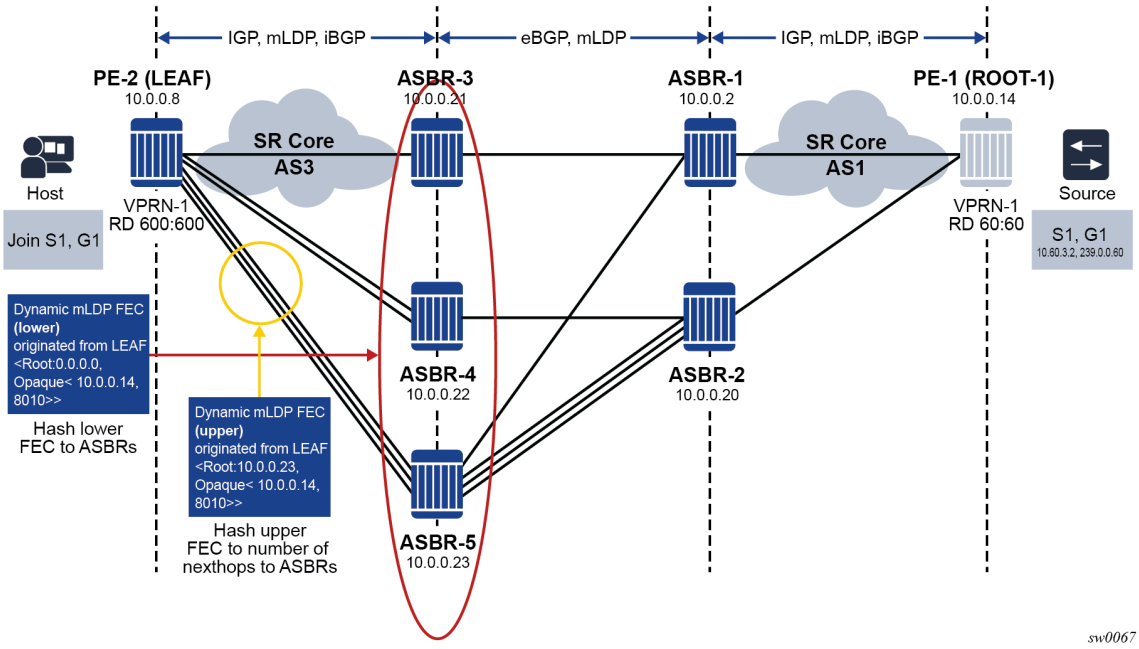
	ASBR node	
Inter-AS multicast context	**Leaf or bud**	**Root or source**
GRT	✓	
VPN	✓	✓

7.3 Hashing for inter-AS

At each leaf or ASBR, there are two FECs: a lower FEC and an upper FEC. The lower FEC is used for hashing to multiple ASBRs and the upper FEC is used to choose the next-hop that connects the leaf node to the ASBR. Hashing is performed based on the opaque value of the FEC. See the "Supported Recursive Opaque Values" section in the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide* for more information.

In Figure 11: Hashing for inter-AS, the leaf generates a lower FEC <0.0.0.0, opaque <10.0.0.14, 8010>>. The lower FEC's opaque <10.0.0.14, 8010> and number of ASBRs (three) are used to decide which ASBR is used based on hashing. After hashing produces ASBR-5 as the result, the upper FEC of <10.0.0.23, opaque <10.0.0.14, 8010>> is created. This upper FEC is used to resolve the ASBR-5 next-hop between the three interfaces that connect the leaf node to ASBR-5.

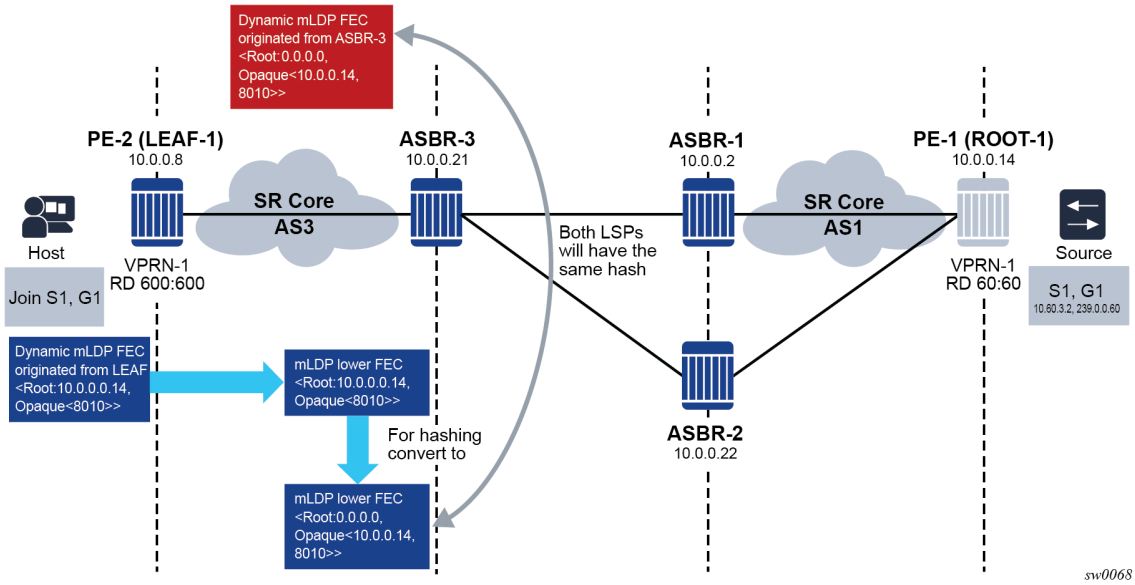
Figure 11: Hashing for inter-AS



7.4 Hashing at the ASBR

Figure 12: Hashing at the ASBR illustrates hashing at the ASBR.

Figure 12: Hashing at the ASBR



In Figure 12: Hashing at the ASBR, the leaf node has ROOT-1 in the RTM for optimized Option C; therefore, the leaf does not generate a recursive type 7 opaque, and only generates a type 1 opaque. When the FEC arrives at ASBR-3, it has a basic type 1 FEC of <ROOT: 10.0.0.14, opaque <8010>>.

If the ASBR also has a host that generates a mLDP LSP toward the root, this FEC looks up <ROOT: 0.0.0.0, opaque <10.0.0.14, 8010>>.

Hashing is performed based on the opaque value of the FEC. See "Supported Recursive Opaque Values" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide* for more information.

The opaque of the leaf node is not the same as the opaque of the ASBR bud node. In this scenario, the two LSPs generate a different ASBR as the next-hop, inefficiently duplicating multicast traffic.

To prevent this problem, SR OS converts the lower FEC of opaque type 1 that arrives from the leaf node into a recursive type 7 FEC, so that the bud FEC generated by the ASBR and the FEC arriving from the leaf node results in the same upper ASBR.

7.5 MLDP over RSVP P2P LSP

The following use cases are described in this section:

- tunneling MLDP over P2P RSVP on core nodes that do not have multicast protocols enabled and are protected through RSVP FRR and TE
- protecting MLDP P2MP over RSVP P2P on a network-segment basis; in networks that cannot afford duplication of multicast bandwidth (MoFRR), it is attractive to use MLDP over RSVP and reserve RSVP FRR for link and node protection.

7.5.1 Summary of procedures for MLDP over RSVP

Use the following commands to advertise an MLDP P2MP FEC to an upstream node using a T-LDP session.

```
configure router ldp prefer-mcast-tunnel-in-tunnel
configure router ldp targeted-session peer mcast-tunneling
```

The following apply when the **mcast-tunneling** command is configured for the LDP peer targeted session on the ELER (downstream) and the **prefer-mcast-tunnel-in-tunnel** and LDP **mcast-tunneling** commands are enabled on ILER (upstream):

- On the upstream router, LDP resolves the nexthop using TTM using a P2P RSVP-TE LSP.
- As shown in the following figure, when the LSR 100.0.0.102 and LSR 100.0.0.101 nodes are directly connected, the T-LDP and link-LDP adjacencies share a common LDP session. The P2MP FEC is advertised over the LDP session and is received by LSR 100.0.0.102, which resolves it over the RSVP LSP if the **mcast-tunneling** option is enabled. The **prefer-mcast-tunnel-in-tunnel** option in LDP dictates if the MLDP FEC is resolved in preference to the tunnel or to the link.

Figure 13: Configuring MLDP over RSVP P2P LSP (MD-CLI)

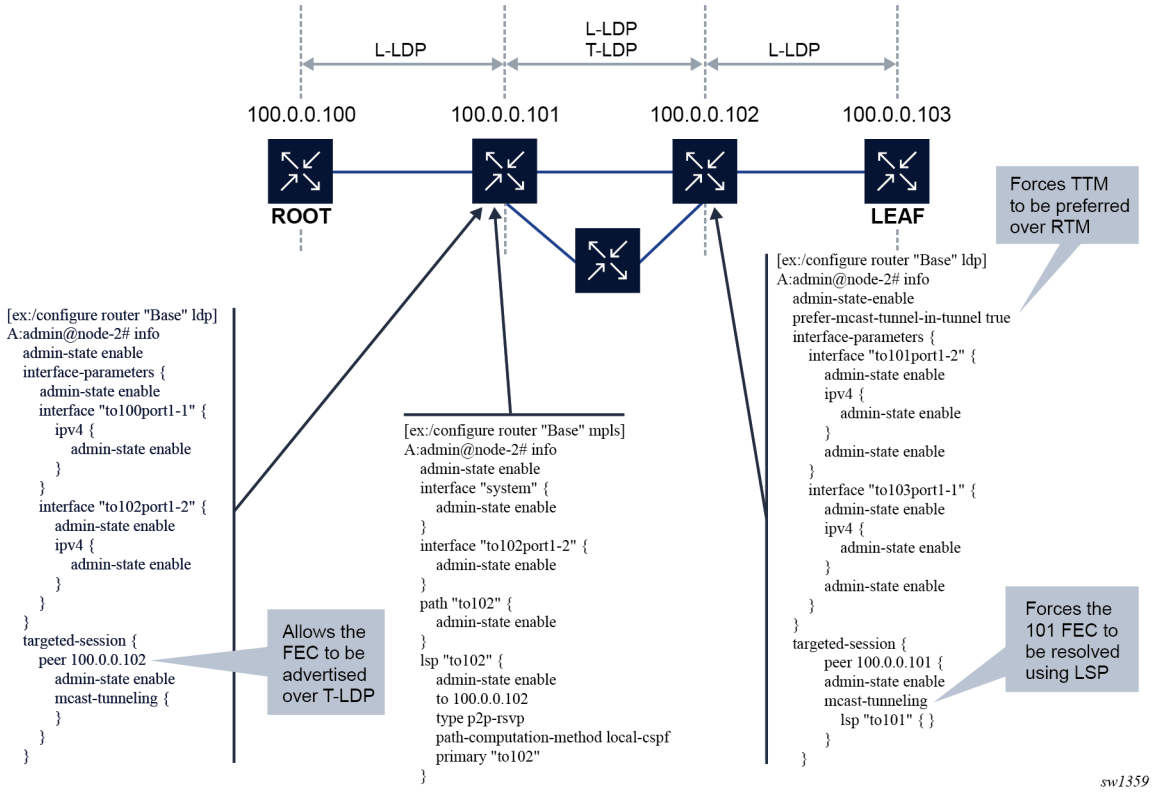
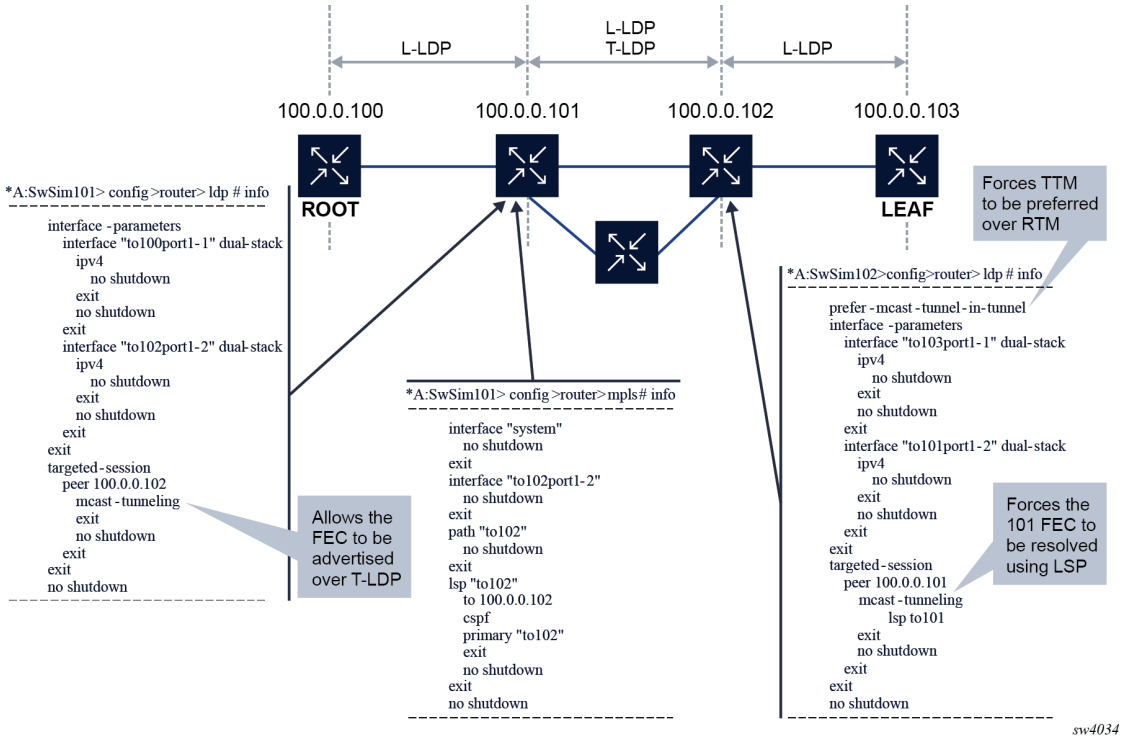


Figure 14: Configuring MLDP over RSVP P2P LSP (classic CLI)



7.5.2 Summary of requirements and procedures for IGP shortcut

The main difference between IGP shortcut and LDP over RSVP-TE is that the LSP is installed in the RIB with IGP shortcut. The T-LDP signaling messages are resolved through IGP shortcut and go over the P2P RSVP-TE. In the case of LDP over RSVP-TE, the P2P RSVP-TE is not installed in the RIB, so the T-LDP signaling is not over RSVP.

The **tunneling** command is required to ensure the label mapping is generated through T-LDP.

For upstream FEC resolution:

- **mcast-tunneling** must be enabled under the targeted peer for the mcast FEC to resolve over an IGP shortcut or using T-LDP.
- If **prefer-mcast-tunnel-in-tunnel** is disabled, the preference is given in the following order:
  1. non-tunneled NHs of IGP route
  2. IGP shortcut (tunneled NHs) of IGP route
  3. direct T-LDP to the root address
  4. indirect T-LDP to an intermediate node using the tunnel endpoints provided by routing; for IGP to compute the tunnel endpoints, the **ldp-over-rsvp** command must be enabled under the IGP context
- If **prefer-mcast-tunnel-in-tunnel** is enabled, the preference is given in the following order:
  1. direct T-LDP to root address

2. indirect T-LDP to an intermediate node using the tunnel endpoints
3. IGP shortcut (tunneled NHs) of IGP route
4. non-tunneled NHs of IGP route

For the downstream direction, a direct RSVP LSP to the downstream peer address is required for T-LDP sessions to be considered.

When the downstream peer is reachable through both a link LDP and a T-LDP, the mcast FEC next-hop is programmed as follows:

- The link LDP session is preferred if **prefer-mcast-tunnel-in-tunnel** is disabled.
- The T-LDP session is preferred if **prefer-mcast-tunnel-in-tunnel** is enabled.

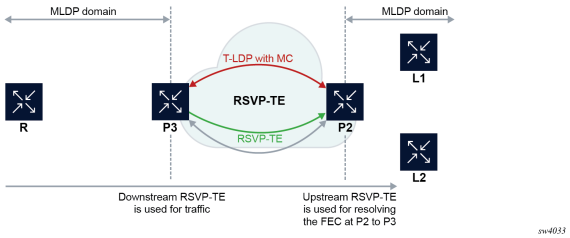
7.5.3 FEC T-LDP session selection

Figure 15: FEC T-LDP session selection shows an example of the FEC T-LDP session selection process.

T-LDP is used to signal MLDP over a core of RSVP-TE. The criteria for selecting the T-LDP session is as follows:

- A T-LDP session with multicast enabled is selected.
- An RSVP-TE LSP may be needed to terminate at the upstream node. The RSVP-TE far-end must be equal to the T-LDP peer. The FEC selects a T-LDP session that has an upstream direction RSVP-TE.

Figure 15: FEC T-LDP session selection



7.5.4 Basic FEC and recursive FEC

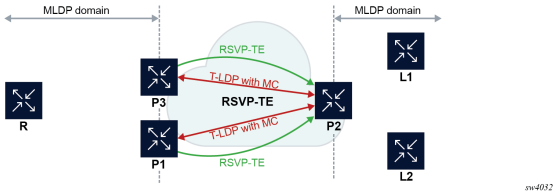
MLDP over RSVP P2P LSP supports both basic and recursive FEC.

The outer root address of the recursive FEC is used for the upstream FEC resolution. On ABRs or ASBRs, a new outer root address is generated and used for upstream FEC resolution.

7.5.5 Two nodes with ECMP upstream

In Figure 16: Two nodes with ECMP upstream, the FECs are hashed between P3 and P1. The hashing criteria are in accordance with RFC 6388 section 2.4.1.1. If P3 has a failure, the FEC is signaled through P1 and all tunnels go down and signal again.

Figure 16: Two nodes with ECMP upstream

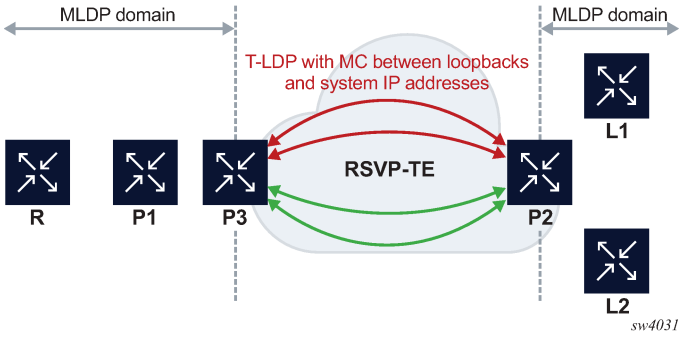


7.5.6 Single upstream node with multiple T-LDP to the upstream node

Figure 17: Single upstream node with multiple T-LDP to the upstream node shows an example of multiple T-LDP sessions to the upstream node.

The T-LDP peer IP that resolves the root is preferred first. Otherwise, the T-LDP peer with the smallest peer address is preferred. There is no ECMP between multiple T-LDP sessions to the same node.

Figure 17: Single upstream node with multiple T-LDP to the upstream node

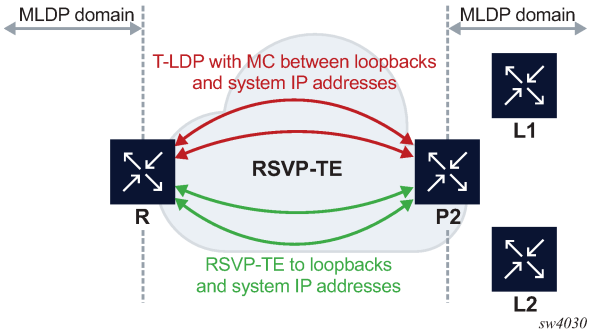


7.5.7 Root node with multiple T-LDP to the root node

Figure 18: Root node with multiple T-LDP to the root node shows an example of multiple T-LDP sessions to the root node.

If the FEC is to the system IP address, the T-LDP peer to the system IP is preferred. Otherwise, the behavior is as described in Single upstream node with multiple T-LDP to the upstream node.

Figure 18: Root node with multiple T-LDP to the root node



7.5.8 Root and leaf connectivity

The root can be colocated on the egress RSVP-TE router (egress from the signaling point of view). The root can be one hop or more away for the egress RSVP-TE router.

The leaf can be colocated on the ingress RSVP-TE router (ingress from the signaling point of view). The leaf can be one hop or more away downstream from the ingress RSVP-TE router.

7.5.9 IGP shortcut and ldp-over-rsvp knob

In the following example, a bidirectional LSPs is configured between Dut-A and Dut-C.

Dut- A ---------- Dut-B ---------- Dut-C

(leaf) (bud) (root)

The IGP shortcut behavior is as follows:

In the upstream direction, if LDP over RSVP on the LSP is disabled (**ldp-over-rsvp** command) from Dut-A to Dut-C, the FEC is signaled upwards using the LSP. This is because it is a shortcut-based LSP and the disabled **ldp-over-rsvp** configuration has no effect on it.

In the downstream direction, if the LDP over RSVP on the LSP is disabled from Dut-C to Dut-A, the bindings are removed on Dut-C even if the tunnel is a shortcut tunnel.

7.5.10 T-LDP peer and RSVP-TE far-end

In the upstream direction, the T-LDP peer and the RSVP endpoint can be different.

In the following example, Router A has Router ID 10.20.1.1 and loopback address 1.1.1.1. Router C has Router ID 10.20.1.3 and loopback address 3.3.3.3.

A---------- C

10.20.1.1 10.20.1.3

1.1.1.1 3.3.3.3

In the upstream direction, SR OS can configure the T-LDP peer between 1.1.1.1 and 3.3.3.3 and also configure the RSVP LSP between Router IDs 10.20.1.1 and 10.20.1.3.

In the downstream direction, the RSVP LSP must terminate on the targeted peer for the FEC to get resolved.

7.5.11 MoFRR considerations

MoFRR over RSVP-TE unicast domain is possible if there are multiple RSVP-TE tunnels.

For the direct case, that is, where there is a direct T-LDP session from the leaf node to the root node, LDP MoFRR is not set up. The user must set up RSVP FRR if redundancy is required.

For the indirect case, that is, where there are T-LDP sessions from the leaf node to the intermediate nodes, LDP sets up MoFRR if multiple eligible tunnel endpoints are provided by IGP and T-LDP sessions exist to these.

LDP uses the tunneled NH tunnel far-end provided in the RTM route to select a PHP. If multiple tunneled NH tunnel far-ends are provided, LDP can use one as primary and one as backup to select the PHPs. If the RTM route has non-tunneled next hops to multiple PHP nodes, it selects a primary and a backup.

For the MLDP-over-RSVP or MLDP-over-IGP shortcut, upstream MoFRR works only if ECMP is set to 2 or greater. There is currently no support for MoFRR leveraging an LFA route. Without ECMP set to 2 or greater, LDP cannot calculate the backup MoFRR path required for upstream traffic protection. The primary and backup ECMP selected path must be disjoint throughout the RSVP-TE cloud to protect against physical failures.

8 Multicast extensions to BGP

This section describes the implementation of extensions to MBGP to support multicast. Instead of assuming that all unicast routes are multicast-capable, some routed environments, in some cases, some ISPs do not support or have limited support for multicast throughout their AS.

BGP is capable of supporting two sets of routing information, one set for unicast routing and the other for multicast routing. The unicast and multicast routing sets either partially or fully overlay one another. To achieve this, BGP has added support for IPv4 and mcast-IPv4 address families. Routing policies can be imported or exported.

The multicast routing information can subsequently be used by the Protocol Independent Multicast (PIM) protocol to perform its Reverse Path Forwarding (RPF) lookups for multicast-capable sources. Thus, multicast traffic can only be routed across a multicast topology and not a unicast topology.

8.1 MBGP multicast topology support

This section describes the implementation of MBGP to support multicast topologies.

8.1.1 Recursive lookup for BGP next hops

The next hop for multicast RPF routes learned by MBGP is not always the address of a directly-connected neighbor. For unicast routing, a router resolves the directly-connected next-hop by repeating the IGP routes. For multicast RPF routes, there are different ways to find the real next-hops:

- Scanning to see if a route encompasses the BGP next hop. If one exists, this route is used. If not, the tables are scanned for the best matching route.
- Checking to see if the recursed next hop is taken from the protocol routing table with the lowest administrative distance (protocol preference). This means that the operating system algorithm must perform multiple lookups in the order of the lowest admin distance. Unlike recursion on the unicast routing table, the longest prefix match rule does not take effect; protocol preference is considered before prefix length. For example, the route 10.0.0.0/14 learned via MBGP is selected over the route 10.0.0.0/16 learned via BGP.

9 MCAC

9.1 MCAC overview

Multicast Connection Admission Control (MCAC) allows a router to limit bandwidth used by multicast channels, either on a router, on access links, or by an ESM subscriber, by controlling the number of channels that are accepted. When a pre-configured limit is reached, the router prevents receivers from joining any new channels not currently established. By rejecting new channel establishment during an overload condition, the degradation of the quality of the existing multicast service offering is avoided. However, as result, running the MCAC function may cause some channels to be temporarily unavailable to receivers under overload.

You can configure one or more MCAC bundle policies to specify multicast channel admission rules and then reference a required MCAC bundle policy on multicast-enabled IPv4 and IPv6 interfaces or group-interfaces. In addition, you can configure per-interface MCAC behavior.

Use the commands in the following context to configure a MCAC bundle policy:

- **MD-CLI**

```
configure mcac policy
```

- **classic CLI**

```
configure router mcac policy
```

MCAC is supported on ESM subscriber interfaces as well as multicast interfaces in the base router instance, VPLS, and in MVPNs. MCAC is supported for IGMP, IGMP-snooping, MLD, and PIM.

The amount of bandwidth multicast channels can consume is limited by user-configured unconstrained and mandatory bandwidth values. Those values can be configured on a per-MCAC bundle policy, per subscriber, per interface, and per MCAC interface policy. The bandwidth limits configured for a subscriber or interface limit multicast bandwidth for that particular subscriber or that interface only. The bandwidth limits configured for an MCAC interface policy limit multicast bandwidth across a set of interfaces that share the same interface policy. If bandwidth limits are defined on multiple levels, all level limits must be satisfied for a channel to be admitted. See MCAC algorithm for more information.

MCAC is not applicable to PIM snooping and MLD snooping.

9.1.1 MCAC bundle policy overview

MCAC bundle policy (shortened here to "MCAC policy" or "policy") is used to define MCAC rules to be applied on an MCAC interface when receivers are trying to join multicast channels. Within each policy, a user can define:

- **multicast channel**

  A channel can be defined using multicast group address only or both source and group addresses. Ranges can be used to group multiple multicast channels into a single MCAC channel. When ranges

are used, each multicast channel within range uses the same channel bandwidth (bandwidth), class, and priority configuration. By default, a channel is represented by the multicast group. For example, each multicast group is a channel. When source-specific accounting is used, each multicast source becomes a channel as well. For example, if there are three multicast groups and two multicast sources for each, there are six channels:

– **channel bandwidth**

  This is the bandwidth value to be used for a channel in MCAC.

– **channel type (mandatory or optional)**

  Mandatory channels have bandwidth pre-reserved on interfaces as soon as they are defined in MCAC policy, while optional channels consume bandwidth on-demand; only when there are active receivers for that channel and the remaining bandwidth allows for channels to be admitted.

– **channel class**

  High and low channel classes are supported. For LAG interfaces, the class parameter allows further prioritizing of the mandatory or optional channels. This brings the number of priority levels to four during reshuffles of the joined channels when LAG ports are changing state.

Note: Multicast channels not specified in an MCAC policy applicable on a specific interface are not subject to MCAC. Treatment of such channels is configurable as either accept or discard.

- **multicast channel bundle**

  The multicast bundle defines multicast channels as per the preceding description. A channel can only be part of one bundle. Source-specific accounting can also be specified at the bundle level. At this level, there is the ability to turn on source-specific accounting only at the bundle level, disable it for a particular bundle, or just follow the default configuration specified at the policy level.

  – **maximum bundle bandwidth**

    This is the maximum bandwidth the channels forming a bundle can consume on an interface.

  – **MCAC constraints**

    This set of rules governs available bandwidth for multicast channels over LAG as LAG ports are changing state.

9.1.2 MCAC algorithm

It is important to point out that the MCAC algorithm is based on configured BW values. The configured channel BW based on MCAC policy is CAC-ed against pre-configured maximum bundle BW and pre-configured subscriber, interface, or MCAC interface-policy multicast BW limits. A channel must pass all levels of CAC before it is accepted. The statements describe the CAC algorithm for a multicast channel defined in MCAC policy.

A join for a particular multicast channel is allowed under the following conditions:

- **mandatory channels**

  A sufficient bandwidth exists on the interface according to the policy settings for the interface (Interface-level MCAC and MCAC-interface-policy-level MCAC) and BW setting for a channel (Bundle-level MCAC). There is always sufficient BW available on the bundle level because mandatory channels get pre-reserved bandwidth.

- **optional channels**

  A sufficient BW exists on both interface (Interface-level MCAC and MCAC-interface-policy-level MCAC) and bundle level (Bundle-Level MCAC) based on channel configured BW and currently available BW on both interface and bundle.

When a policy is evaluated over a set of existing channels (applicable for MCAC on LAG when the number of ports in the LAG changes and applicable to subscribers when the submac policy is enabled on a subscriber), the channels are evaluated and admitted/dropped based on the following priority order: mandatory-high, mandatory-low, optional-high, optional-low.

This method does not guarantee that all bundles are fully allocated. However, this method does ensure that all mandatory-high channels are allocated before any mandatory-low channels are allocated.

When a new MCAC bundle policy is applied, the algorithm is forced to admit all currently joined channels to prevent any drops. This can result in an oversubscription until some of the joined channels disconnect. The same behavior applies when adding a new MCAC interface policy: all the joined channels are admitted, without dropping anything.

9.1.2.1 Interface-level MCAC details

Interface-level MCAC constraints are applied to the interface on which the join was received. Mandatory and optional channels are allowed under the following conditions:

- **mandatory channels**

  The bandwidth for the already-accepted mandatory channels plus the bandwidth of this channel cannot be greater than the configured mandatory bandwidth on this interface.

- **optional channels**

  The bandwidth for the already-accepted optional channels plus the bandwidth of this channel cannot be greater than the configured amount of unconstrained bandwidth less the configured amount of mandatory bandwidth on this interface.

9.1.2.1.1 MCAC-interface-policy-level MCAC details

MCAC interface policies are defined system wide and used on MCAC interfaces via assignment of the policy to one or more interfaces to, for example, limit multicast BW across a group of interfaces/ports, across a line card or across a system. If an MCAC interface policy is assigned to an interface with Interface-level constraints configured, then both Interface-level MCAC as described above and MCAC-interface-policy-level MCAC must be satisfied for a channel to be admitted.

Mandatory and optional channels are allowed under the following conditions:

- **mandatory channels**

  The bandwidth for the already-accepted mandatory channels on this and any other interface using this MCAC interface policy plus the bandwidth of this channel cannot be greater than the configured mandatory bandwidth for this MCAC interface policy.

- **optional channels**

  The bandwidth for the already-accepted optional channels on this and any other interface using this MCAC interface policy plus the bandwidth of this channel cannot be greater than the configured amount of unconstrained bandwidth less the configured amount of mandatory bandwidth for this MCAC interface policy.

Thus, when MCAC interface policy is used, admitting a channel on one interface affects all interfaces sharing the same MCAC interface policy.

9.1.2.2 Bundle-level MCAC details

Bundle-level CAC is applied to the bundle to which the channel that triggered the MCAC algorithm belongs.

Mandatory and optional channels are allowed under the following conditions:

- **mandatory channels**

  Mandatory channels are always allowed.

- **optional channels**

  The allocated bundle bandwidth cannot exceed the configured bandwidth. The allocated bandwidth equals the bandwidth of all the mandatory channels belonging to that bundle plus the bandwidth of the optional channels already accepted plus the bandwidth of this optional channel.

9.1.3 MCAC on Link Aggregation Group interfaces

When MCAC enabled interfaces reside on a LAG, SR OS allows users to change MCAC behavior when the number of active ports in a LAG changes. Both MCAC policy bundle and MCAC interface allows users to define multiple MCAC levels per LAG based on the number of active ports in the LAG. For each level, users can configure corresponding BW limits.

When MCAC LAG constraints are enabled, the level to use is selected automatically based on the configuration and a currently active number of LAG ports. In a case of the available bandwidth reduction (for example, a LAG link failure causes change to a level with smaller BW configured), MCAC attempts first to fit all mandatory channels (in an arbitrary order). If there is no sufficient capacity to carry all mandatory channels in the degraded mode, some channels are dropped and all optional channels are dropped. If after evaluation of mandatory channels, there remains available bandwidth, then all optional channels are re-evaluated (in an arbitrary order). Channel re-evaluation employs the above-described MCAC algorithm applied at the interface and bundle levels that use the constraints for the degraded mode of operation.

9.2 Configuring MCAC with CLI

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

9.2.1 Basic MCAC configuration

Perform the following basic MCAC policy configuration tasks:

- Configure the policy name.
- Configure the bundle for the policy.
- Specify the default action.

Use the commands in the following contexts to configure MCAC policies:

- **MD-CLI**

```
configure mcac policy
```

- **classic CLI**

```
configure router mcac policy
```

The following example shows an MCAC configuration:

Example: MD-CLI

```
[ex:/configure mcac]
A:admin@node-2# info
    policy "btv_fr" {
        description "foreign TV offering"
        bundle "FOR" {
            admin-state enable
            bandwidth 30000
            channel start 239.0.3.1 end 239.0.3.1 source 232.252.0.1/24  {
                bandwidth 4000
            }
            channel start 239.0.3.2 end 239.0.3.2 source 232.252.0.2/24  {
                bandwidth 4000
            }
            channel start 239.0.4.1 end 239.0.4.1 source 232.252.0.3/24 {
                bandwidth 3500
                priority-class high
                type mandatory
            }
            channel start 239.0.4.2 end 239.0.4.2 source 232.252.0.5/24 {
                bandwidth 3500
                priority-class high
            }
            channel start 239.0.4.3 end 239.0.4.3 source 232.252.0.100/24 {
                bandwidth 2800
                type mandatory
            }
            channel start 239.0.4.4 end 239.0.4.4 source 232.252.0.200/24 {
                bandwidth 2800
            }
            mc-constraints {
                level 1 {
                    bandwidth 20000
                }
                level 2 {
                    bandwidth 20000
                }
                level 3 {
                    bandwidth 20000
                }
                level 4 {
                    bandwidth 20000
                }
                level 5 {
                    bandwidth 20000
                }
                level 6 {
                    bandwidth 20000
                }
                lag-port-down "lag-1" number-down 1 {
                    level 1
```

```
                }
                lag-port-down "lag-1" number-down 2 {
                    level 3
                }
                lag-port-down "lag-1" number-down 3 {
                    level 5
                }
                lag-port-down "lag-2" number-down 1 {
                    level 1
                }
                lag-port-down "lag-2" number-down 2 {
                    level 3
                }
                lag-port-down "lag-2" number-down 3 {
                    level 5
                }
            }
        }
    }
    policy "btv_v1" {
        description "eastern TV offering"
        bundle "VRT" {
            admin-state enable
            bandwidth 120000
            channel start 239.1.2.0 end 239.1.2.4 source 232.252.0.1/24 {
                bandwidth 4000
                priority-class high
                type mandatory
            }
            channel start 239.1.2.5 end 239.1.2.5 source 232.252.0.2/24 {
                bandwidth 20000
                type mandatory
            }
            channel start 239.1.2.10 end 239.1.2.10 source 232.252.0.5/24 {
                bandwidth 8000
                type mandatory
            }
            channel start 239.2.2.0 end 239.2.2.4 source 232.252.0.10/24 {
                bandwidth 4000
            }
            channel start 239.2.2.5 end 239.2.2.5 source 232.252.0.20/24 {
                bandwidth 10000
                priority-class high
            }
            channel start 239.2.2.6 end 239.2.2.6 source 232.252.0.30/24 {
                bandwidth 10000
                priority-class high
            }
            channel start 239.2.2.7 end 239.2.2.7 source 232.252.0.100/24 {
                bandwidth 10000
            }
            channel start 239.2.2.8 end 239.2.2.8 source 232.252.0.200/24 {
                bandwidth 10000
            }
            mc-constraints {
                level 1 {
                    bandwidth 60000
                }
                level 2 {
                    bandwidth 50000
                }
                level 3 {
                    bandwidth 40000
                }
```

```
                level 4 {
                    bandwidth 30000
                }
                level 5 {
                    bandwidth 20000

                level 6 {
                    bandwidth 10000
                }
                lag-port-down "lag-1" number-down 1 {
                    level 1
                }
                lag-port-down "lag-1" number-down 2 {
                    level 3
                }
                lag-port-down "lag-1" number-down 3 {
                    level 5
                }
                lag-port-down "lag-2" number-down 1 {
                    level 1
                }
                lag-port-down "lag-2" number-down 2 {
                    level 3
                }
                lag-port-down "lag-2" number-down 3 {
                    level 5
                }
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config>router>mcac# info
----------------------------------------------
        policy "btv_fr"
            description "foreign TV offering"
            bundle "FOR" create
                bandwidth 30000
                channel 239.0.3.1 239.0.3.1 bw 4000
                channel 239.0.3.2 239.0.3.2 bw 4000
                channel 239.0.4.1 239.0.4.1 bw 3500 class high type mandatory
                channel 239.0.4.2 239.0.4.2 bw 3500 class high
                channel 239.0.4.3 239.0.4.3 bw 2800  type mandatory
                channel 239.0.4.4 239.0.4.4 bw 2800
                mc-constraints
                    level 1 bw 20000
                    level 2 bw 20000
                    level 3 bw 20000
                    level 4 bw 20000
                    level 5 bw 20000
                    level 6 bw 20000
                    lag-port-down 1 number-down 1 level 1
                    lag-port-down 1 number-down 2 level 3
                    lag-port-down 1 number-down 3 level 5
                    lag-port-down 2 number-down 1 level 1
                    lag-port-down 2 number-down 2 level 3
                    lag-port-down 2 number-down 3 level 5
                exit
                no shutdown
            exit
        exit
        policy "btv_vl"
```

```
            description "eastern TV offering"
            bundle "VRT" create
                bandwidth 120000
                channel 239.1.2.0 239.1.2.4 bw 4000 class high type mandatory
                channel 239.1.2.5 239.1.2.5 bw 20000  type mandatory
                channel 239.1.2.10 239.1.2.10 bw 8000  type mandatory
                channel 239.2.2.0 239.2.2.4 bw 4000
                channel 239.2.2.5 239.2.2.5 bw 10000 class high
                channel 239.2.2.6 239.2.2.6 bw 10000 class high
                channel 239.2.2.7 239.2.2.7 bw 10000
                channel 239.2.2.8 239.2.2.8 bw 10000
                mc-constraints
                    level 1 bw 60000
                    level 2 bw 50000
                    level 3 bw 40000
                    level 4 bw 30000
                    level 5 bw 20000
                    level 6 bw 10000
                    lag-port-down 1 number-down 1 level 1
                    lag-port-down 1 number-down 2 level 3
                    lag-port-down 1 number-down 3 level 5
                    lag-port-down 2 number-down 1 level 1
                    lag-port-down 2 number-down 2 level 3
                    lag-port-down 2 number-down 3 level 5
                exit
                no shutdown
            exit
        exit
-----------------------------------------------
```

9.2.2 Configuring MCAC

You can add MCAC policies to the following configurations:

- a SAP
- a spoke-SDP
- a mesh-SDP
- an IGMP interface
- a PIM interface

The following example displays an enabled IGMP and PIM configurations.

Example: MD-CLI

```
[ex:/configure router "2" igmp]
A:admin@node-2# info
    ...
    admin-state enable
    query-interval 125
    query-last-member-interval 1
    query-response-interval 10
    robust-count 2
    interface "lax-vls" {
        admin-state enable
        version 3
        mcac {
            policy "btv_fr"
        }
```

```
    }
    interface "p1-ix"
        admin-state enable
        version 3
    }
[ex:/configure router "Base" pim]
A:admin@node-2# info
    admin-state enable
    apply-to none
    interface "lax-sjc" {
        admin-state enable
        hello-interval 30
        priority 1
        tracking-support false
        bsm-check-rtr-alert true
        multicast-senders auto
    }
    interface "lax-vls" {
        admin-state enable
        hello-interval 30
        priority 1
        tracking-support false
        bsm-check-rtr-alert true
        multicast-senders auto
        mcac {
            policy "btv_fr"
        }
    }
    interface "pl-ix" {
        admin-state enable
        hello-interval 30
        priority 1
        tracking-support false
        bsm-check-rtr-alert true
        multicast-senders auto
    }
    interface "system" {
        admin-state enable
        hello-interval 30
        priority 1
        tracking-support false
        bsm-check-rtr-alert true
        multicast-senders auto
    }
    rp {
        bootstrap {
            import ["pim-import"]
        }
        ipv4 {
            bsr-candidate {
                admin-state disable
                priority 0
                hash-mask-len 30
            }
            rp-candidate {
                admin-state disable
                holdtime 150
                priority 192
            }
            static {
                address 10.22.187.237 {
                    group-prefix 239.24.24.24/32 { }
                }
            }
```

```
        }
```

Example: classic CLI

```
-----------------------------------------------
A:node-2>config>router>igmp# info
-----------------------------------------------
        interface "lax-vls"
            mcac
                policy "btv_fr"
            version 3
            no shutdown
        exit
        interface "p1-ix"
            version 3
            no shutdown
        exit
        query-interval 125
        query-last-member-interval 1
        query-response-interval 10
        robust-count 2
        no shutdown
-----------------------------------------------
A:node-2>config>router>igmp# exit
A:node-2>config>router# pim
-----------------------------------------------
A:node-2>config>router>pim# info
-----------------------------------------------
            no import join-policy
            no import register-policy
            interface "system"
                priority 1
                hello-interval 30
                multicast-senders auto
                no tracking-support
                bsm-check-rtr-alert
                no shutdown
            exit
            interface "lax-vls"
                priority 1
                hello-interval 30
                multicast-senders auto
                no tracking-support
                bsm-check-rtr-alert
                mcac
                    policy "btv_fr"
                exit
                no shutdown
            exit
            interface "lax-sjc"
                priority 1
                hello-interval 30
                multicast-senders auto
                no tracking-support
                bsm-check-rtr-alert
                no shutdown
            exit
            interface "p1-ix"
                priority 1
                hello-interval 30
                multicast-senders auto
                no tracking-support
                bsm-check-rtr-alert
```

```
                no shutdown
            exit
            apply-to none
            rp
                no bootstrap-import
                no bootstrap-export
                static
                    address 10.22.187.237
                        no override
                        group-prefix 239.24.24.24/32
                    exit
                exit
                bsr-candidate
                    shutdown
                    priority 0
                    hash-mask-len 30
                    no address
                exit
                rp-candidate
                    shutdown
                    no address
                    holdtime 150
                    priority 192
                exit
            exit
            no shutdown
-------------------------------------------------
```

10 GTM

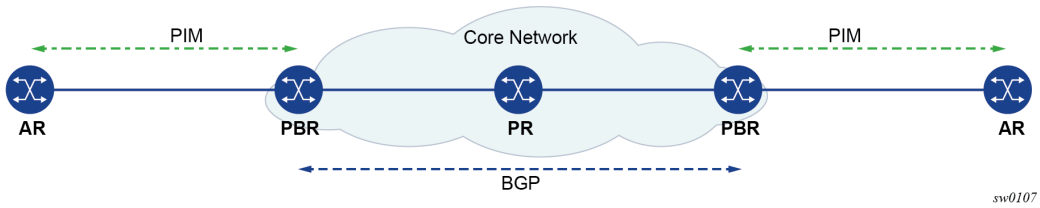
This section describes how to configure GTM.

10.1 GTM overview

GTM with BGP Multicast VPN (BGP-MVPN), as specified in RFC 7716, allows a Service Provider (SP) to use the same multicast architecture that was originally developed for VPNs to distribute multicast routing information that is not specific to VPNs. Instead of storing the routing information in VRFs, multicast routing information is maintained in a global table for the router.

The architecture can be logically divided into a core network and non-core (attachment) networks. The multicast routing protocol used in the core network may not be the same as the protocol used in the attachment networks. As there is a protocol boundary between the core and attachment networks, the term Protocol Boundary Router (PBR) refers to the core routers that are at the boundary. A PBR is not necessarily an edge router in the PE sense; however, a PBR in the SP network marks the border of any tunnels that are used to transport multicast traffic across the core network. Routers that are attached to the PBRs but that are not part of the core network are referred to as Attachment Routers (ARs). See Figure 19: GTM network topology example.

Figure 19: GTM network topology example



Multicast data traffic from an AR is tunneled through the core network from an ingress PBR to one or more egress PBRs, using multicast routing information stored in the PBR's global table. The global table learns the PBR's multicast routing information from the ARs attached to the PBR and distributes the information among the PBRs using BGP. PBRs use the same BGP-MVPN procedures used by PE routers to route multicast VPN traffic, with some adaptations to the procedures to use the global table instead of a VRF.

By using the BGP procedures designed for MVPN to support GTM, a single control plane is available to govern the use of both VPN and non-VPN multicast. The features and characteristics of MVPN carry over automatically to GTM, including, but not limited to:

- scaling
- aggregation
- transport over RSVP tunnels in the SP network
- support for non-segmented intra-autonomous systems (ASs) tunnels
- support for PIM-SSM outside of the core
- support for both IPv4 and IPv6 multicast flows over an IPv4 SP infrastructure

- support for unsolicited flooded data (including support for BSR as an RP-to-group mapping protocol

10.1.1 BGP-MVPN procedures in GTM

This section describes the BGP procedures designed for MVPN to support GTM.

10.1.1.1 Route distinguishers and route targets

The BGP routes used in the MVPN procedures have a Subsequent Address Family Identifier (SAFI) value of 5, or MCAST-VPN. The Network Layer Reachability Information (NLRI) format for MCAST-VPN routes consists of a Route Type (RT) field and depending on the RT, a Route Distinguisher (RD) Extended Community (EC) field.

Note: The ECs are automatically configured for GTM and are not visible in the configuration.

To distinguish MCAST-VPN routes originated for VPNs from MCAST-VPN routes in support of GTM, the RD field, if defined within that route's NLRI, must be set to zero (that is, 64 bits of zero). An RD of all zeros associates that route with GTM, as no VRF can have an RD of zero.

MVPN procedures use two types of RTs, one of which is carried only in the routes of C-multicast shared tree joins, C-multicast source tree joins, and leaf auto-discovery routes (A-D routes). This RT type identifies the PE router that has been selected by the route's originator as the Upstream PE or as the Upstream Multicast Hop (UMH) for a particular multicast flow or set of multicast flows. This RT must be an IPv4- or IPv6-address-specific EC, where the Global Administrator field identifies the Upstream PE or the UMH. If the Global Administrator field identifies the Upstream PE, the Local Administrator field identifies a particular VRF in that PE.

To support GTM, this type of RT is used in the same situations as in the MVPN specifications, with the modification that the Local Administrator field of this RT type must always be set to zero. This implicitly identifies the global table instead of identifying a VRF. This type of RT is referred to as an upstream-node-identifying RT.

10.1.1.2 UMH-eligible routes

For MVPN, routes of SAFI 128 or 129 are UMH-eligible routes. For GTM, routes of SAFI 1, SAFI 4, or SAFI 2 are UMH-eligible routes. Imported routes of SAFI 2 in the global table are UMH-eligible routes; otherwise, routes of SAFI 1 or SAFI 4 are considered UMH-eligible routes. For UMH determination, SAFI 1 and SAFI 4 routes containing the same IP prefix in their respective NLRI fields are considered by the BGP best-path selection process to be comparable.

UMH-eligible routes that have a SAFI of 1, 2, or 4 carry both the VRF Route Import EC and the Source AS EC. These ECs are automatically configured for GTM.

10.1.1.3 BGP route types supported

Table 4: BGP route types describes the BGP route types.

Table 4: BGP route types

BGP route type	Name	Description	Supported for GTM
1	Intra-AS I-PMSI AD route	Originated by all PBR routers. Used for advertising and learning intra-AS MVPN membership information.	Yes, always originated by SR OS
2	Inter-AS I-PMSI A-D route	Originated by ASBR routers. Used for advertising and learning inter-AS MVPN membership information.	No (no Inter-AS support)
3	S-PMSI A-D route	Originated by sender PBRs. Used for initiating a selective P-tunnel for a particular (C-S, C-G).	Yes
4	Leaf A-D route	Originated by receiver PBRs in response to receiving a Type 3 route. Used by sender PBR to discover the leaves of a selective P-tunnel.	Yes
5	Source Active A-D route	Originated by the PBR that discovers an active VPN multicast source. Used by PBRs to learn the identity of active VPN multicast sources.	Yes
6	Shared Tree Join route	Originated by receiver PBRs. Originated when a PE receives a shared tree C-join (C-*, C-G) through its PE-CE interface.	Yes
7	Source Tree Join route	Originated by receiver PBRs. Originated when a PBR receives a source tree C-join (C-S, C-G) or originated by the PBR that already has a Type 6 route and receives a Type 5 route.	Yes, for non-segmented trees

10.2 Configure GTM

This section describes how to configure GTM.

10.2.1 Configuration recommendations

When configuring GTM, consider the following recommendations:

- In a dual-homing configuration, ECMP to the upstream multicast hop (UMH) routers must be configured for the BGP routes to be used. If the UMH routers are unreachable via ECMP, it may cause duplicate traffic in the core. This behavior is consistent with RFC 7716, section 2.3.4., which states that the single forwarder selection (SFS) procedure cannot be applied to GTM.

- For IPv6 GTM, the IPv4 address must be configured as the IPv6 system address, as in the following example.

Note: You cannot configure GTM auto-discovery in the following contexts if IPv4 or IPv6 multicast fast failover (MoFRR) are enabled:

- **MD-CLI**

```
configure router pim ipv4 gtm auto-discovery
```

- **classic CLI**

```
configure router pim gtm auto-discovery
```

The following example shows a system interface configuration.

Example: MD-CLI

```
[ex:/configure router "Base"]
A:admin@node-2# info
    interface "system" {
        admin-state enable
        ipv4 {
            primary {
                address 10.20.1.4
                prefix-length 32
            }
        }
        ipv6
            primary {
                address ff0e::db8:104
                prefix-length 128
            }
        }
    }
```

Example: classic CLI

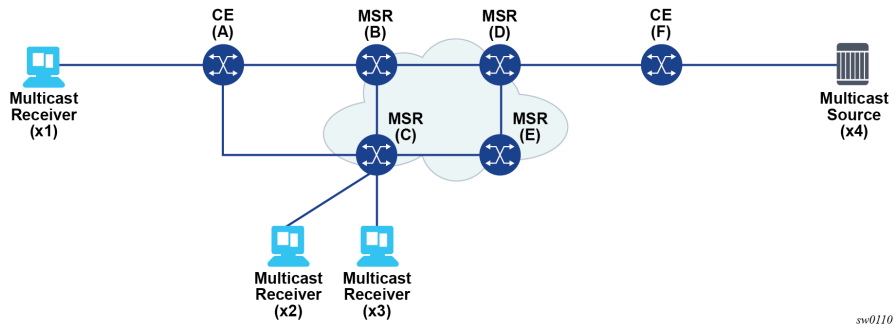
```
A:node-2>config>router# info
        interface "system"
            address 10.20.1.4/32
            ipv6
                address ff0e::db8:104/128
            exit
            no shutdown
        exit
```

10.2.2 Configuring GTM with CLI

Prerequisites

Figure 20: Example configuration shows an example GTM configuration.

Figure 20: Example configuration



In the example configuration, the following applies:

- Routers A and F are CE routers.
- Routers B, C, D, and E are MSR routers in the core network.
- The multicast source is at x4.
- The multicast receivers are at x1, x2, and x3.

Perform the following steps to configure GTM:

Procedure

Step 1. Configure PIM for GTM.

Example

PIM for GTM configuration (MD-CLI)

```
[ex:/configure router "base" pim]
A:admin@node-2# info
    admin-state enable
    apply-to all
    mc-ecmp-balance false
    ipv4 {
        gtm {
            auto-discovery bgp
        }
    }
    interface "intf_to_B" {
        admin-state disable
    }
     interface "intf_to_E" {
        admin-state disable
    }
    rp {
        ipv4 {
            static {
                address 10.100.1.1 {
                    group-prefix 224.0.0.0/4 {
                }
            }
        }
    }
```

Example

PIM for GTM configuration (classic CLI)

```
A:node-2>config>router>pim# info
----------------------------------------------
            interface "intf_to_B"
                shutdown
            exit
            interface "intf_to_E"
                shutdown
            exit
            apply-to all
            rp
                static
                    address 10.100.1.1
                        group-prefix 224.0.0.0/4
                    exit
                exit
                bsr-candidate
                    shutdown
                exit
                rp-candidate
                    shutdown
                exit
            exit
            gtm
                auto-discovery default
            exit
            no mc-ecmp-balance
            no shutdown
----------------------------------------------
```

Step 2. Configure GTM.

Example

GTM configuration (MD-CLI)

```
[ex:/configure router "Base" gtm]
A:admin@node-2# info
    mvpn true
    provider-tunnel {
        inclusive {
            rsvp {
                admin-state enable
                lsp-template "IpmsiTmp1"
            }
        }
        selective {
            maximum-p2mp-spmsi 4000
            data-threshold {
                group-prefix 224.0.0.0/4 {
                    threshold 1
                }
            }
            rsvp {
                admin-state enable
                lsp-template "SpmsiTmp1"
            }
        }
    }
```

Example

GTM configuration (classic CLI)

```
A:node-2>config>router>gtm# info
----------------------------------------------
            mvpn
            provider-tunnel
                inclusive
                    rsvp
                        lsp-template "IpmsiTmpl"
                        no shutdown
                    exit
                exit
                selective
                    rsvp
                        lsp-template "SpmsiTmpl"
                        no shutdown
                    exit
                    maximum-p2mp-spmsi 4000
                    data-threshold 224.0.0.0/4 1
                exit
            exit
----------------------------------------------
```

Step 3. Configure a route policy for BGP.

The following output displays the route policy configuration.

Example

BGP route-policy configuration (MD-CLI)

```
[ex:/configure policy-options]
A:admin@node-2# info
    prefix-list "sourceList" {
        prefix 10.10.0.0/16 type longer {
        }
        prefix 10.100.0.0/16 type longer {
        }
        prefix 10.114.0.0/16 type longer {
        }
        prefix 2001:db8:10:10:0:0/96 type longer {
        }
        prefix 2001:db8:100:0:0/96 type longer {
        }
        prefix 2001:db8:114:0:0/96 type longer {
        }
    }
    policy-statement "acceptAll" {
        default-action {
            action-type accept
        }
    }
    policy-statement "acceptAllBgp" {
        default-action {
            action-type drop
            }
        }
        entry 10 {
            from {
                protocol {
                    name rip
                }
```

```
            }
            action {
                action-type accept
            }
        entry 11 {
            from {
                protocol {
                    name ripng
                }
            }
            action {
                action-type accept
            }
        }
    }
    policy-statement "acceptAllPref8" {
        default-action {
            action-type accept
            preference 8
        }
    }
```

Example

BGP route-policy configuration (classic CLI)

```
A:node-2>config>router>policy-options# info
----------------------------------------------
             prefix-list "sourceList"
                 prefix 10.10.0.0/16 longer
                 prefix 10.100.0.0/16 longer
                 prefix 10.114.0.0/16 longer
                 prefix 2001:db8:10:10:0:0/96 longer
                 prefix 2001:db8:100:0:0/96 longer
                 prefix 2001:db8:114:0:0/96 longer
             exit
             policy-statement "acceptAll"
                 default-action accept
                 exit
             exit
             policy-statement "acceptAllBgp"
                 entry 10
                     from
                         protocol rip
                     exit
                     action accept
                     exit
                 exit
                 entry 11
                     from
                         protocol ripng
                     exit
                     action accept
                     exit
                 exit
                 default-action drop
                 exit
             exit
             policy-statement "acceptAllPref8"
                 default-action accept
                     preference 8
                 exit
             exit
```

```
-----------------------------------------------
```

Step 4. Configure BGP for GTM.

The following output displays the route policy configuration.

Example

BGP configuration for GTM (MD-CLI)

```
[ex:/configure router "Base" bgp]
A:admin@node-2# info
    admin-state enable
    connect-retry 1
    router-id 10.20.1.4
    inter-as-vpn true
    mvpn-vrf-import-subtype-new true
    rapid-withdrawal true
    best-path-selection {
        ignore-nh-metric true
    }
    rapid-update {
        mvpn-ipv4 true
        mdt-safi true
        mvpn-ipv6 true
    }
    export {
        policy ["acceptAlBgp"]
    }
    multipath {
        max-paths 16
    }
    group "none" {
        next-hop-self true
        local-address 10.20.1.4
        family {
            ipv4 true
            ipv6 true
            mvpn-ipv4 true
            mvpn-ipv6 true
            label-ipv4 true
            label-ipv6 true
        }
    }
    neighbor "10.20.1.2" {
        peer-as 200
    }
    neighbor "10.10.1.3" {
        med-out 100
        peer-as 200
    }
    neighbor "10.20.1.5" {
        med-out 100
        peer-as 200
    }
```

Example

BGP configuration for GTM (classic CLI)

```
A:node-2>config>router>bgp# info
-----------------------------------------------
            connect-retry 1
            multipath 16
```

```
            export "acceptAllBgp"
            router-id 10.20.1.4
            rapid-withdrawal
            rapid-update mvpn-ipv4 mdt-safi mvpn-ipv6
            mvpn-vrf-import-subtype-new
            best-path-selection
                ignore-nh-metric
            exit
            group "none"
                family ipv4 ipv6 mvpn-ipv4 mvpn-ipv6 label-ipv4 label-ipv6
                next-hop-self
                local-address 10.20.1.4
                neighbor 10.20.1.2
                    peer-as 200
                exit
                neighbor 10.20.1.3
                    med-out 100
                    peer-as 200
                exit
                neighbor 10.20.1.5
                    med-out 100
                    peer-as 200
                exit
            exit
            no shutdown
-------------------------------------------
```

Step 5. Use the following command to display MVPN configuration information.

```
show router mvpn
```

Example

Display MVPN configuration information

```
===============================================================================
MVPN Base configuration data
===============================================================================
signaling          : Bgp                  auto-discovery     : Default
UMH Selection      : Highest-Ip           SA withdrawn       : Disabled
intersite-shared   : Enabled              Persist SA         : Disabled
vrf-import         : N/A
vrf-export         : N/A
vrf-target         : unicast
C-Mcast Import RT  : target:10.20.1.4:0
ipmsi              : rsvp IpmsiTmpl
i-pmsi P2MP AdmSt  : Up
i-pmsi Tunnel Name : IpmsiTmpl-gtm-73881
enable-bfd-root    : false                enable-bfd-leaf    : false
Mdt-type           : sender-receiver
BSR signalling     : none
Wildcard s-pmsi    : Disabled
Multistream-SPMSI  : Disabled
spmsi              : rsvp SpmsiTmpl
s-pmsi P2MP AdmSt  : Up
max-p2mp-spmsi     : 4000
data-delay-interval: 3 seconds
enable-asm-mdt     : N/A
data-threshold     : 224.0.0.0/4 --> 1 kbps
===============================================================================
```

Step 6. Use the following command to display the PIM group configuration information.

```
show router pim group
```

Example

Display PIM group configuration information

```
===============================================================================
Legend:  A = Active   S = Standby
===============================================================================
PIM Groups ipv4
===============================================================================
Group Address               Type               Spt Bit  Inc Intf
No.Oifs
    Source Address              RP                  State    Inc Intf(S)
-------------------------------------------------------------------------------
239.100.0.0                 (S,G)              spt vprn_itf_D_2b* 1
    10.114.1.2                 10.100.1.1
239.100.0.1                 (S,G)              spt vprn_itf_D_2b* 1
    10.114.1.2                 10.100.1.1
239.100.0.2                 (S,G)              spt vprn_itf_D_2b* 1
    10.114.1.2                 10.100.1.1
239.100.0.3                 (S,G)              spt vprn_itf_D_2b* 1
    10.114.1.2                 10.100.1.1
-------------------------------------------------------------------------------
Groups : 4
===============================================================================
```

Step 7. Use the following commands to display the PIM group detail configuration information.

```
show router pim group detail
```

Example

Display PIM group detail configuration information

```
===============================================================================
PIM Source Group ipv4
===============================================================================
Group Address      : 239.100.0.0
Source Address     : 10.114.1.2
RP Address         : 10.100.1.1
Advt Router        :
Flags              : spt                Type              : (S,G)
Mode               : sparse
MRIB Next Hop      : 10.100.1.1
MRIB Src Flags     : remote
Keepalive Timer Exp: 0d 00:02:53
Up Time            : 0d 00:09:50        Resolved By       : rtable-u
Up JP State        : Joined             Up JP Expiry      : 0d 00:00:23
Up JP Rpt          : Not Joined StarG   Up JP Rpt Override : 0d 00:00:00
Register State     : No Info
Reg From Anycast RP: No
Rpf Neighbor       : 10.100.1.1
Incoming Intf      : vprn_itf_D_2base
Outgoing Intf List : mpls-if-73881 (mpls-if-73885)
Curr Fwding Rate   : 1.3 kbps
Forwarded Packets  : 132                Discarded Packets  : 0
Forwarded Octets   : 6072               RPF Mismatches     : 0
Spt threshold      : 0 kbps             ECMP opt threshold : 7
Admin bandwidth    : 1 kbps
```

```
===============================================================================
PIM Source Group ipv4
===============================================================================
Group Address      : 239.100.0.1
Source Address     : 10.114.1.2
RP Address         : 10.100.1.1
Advt Router        :
Flags              : spt                Type              : (S,G)
Mode               : sparse
MRIB Next Hop      : 10.100.1.1
MRIB Src Flags     : remote
Keepalive Timer Exp: 0d 00:02:53
Up Time            : 0d 00:09:50        Resolved By       : rtable-u
Up JP State        : Joined             Up JP Expiry      : 0d 00:00:23
Up JP Rpt          : Not Joined StarG   Up JP Rpt Override : 0d 00:00:00
Register State     : No Info
Reg From Anycast RP: No
Rpf Neighbor       : 10.100.1.1
Incoming Intf      : vprn_itf_D_2base
Outgoing Intf List : mpls-if-73881 (mpls-if-73886)
Curr Fwding Rate   : 1.3 kbps
Forwarded Packets  : 141                Discarded Packets  : 0
Forwarded Octets   : 6486               RPF Mismatches     : 0
Spt threshold      : 0 kbps             ECMP opt threshold : 7
Admin bandwidth    : 1 kbps
===============================================================================
PIM Source Group ipv4
===============================================================================
Group Address      : 239.100.0.2
Source Address     : 10.114.1.2
RP Address         : 10.100.1.1
Advt Router        :
Flags              : spt                Type              : (S,G)
Mode               : sparse
MRIB Next Hop      : 10.100.1.1
MRIB Src Flags     : remote
Keepalive Timer Exp: 0d 00:02:52
Up Time            : 0d 00:09:51        Resolved By       : rtable-u
Up JP State        : Joined             Up JP Expiry      : 0d 00:00:22
Up JP Rpt          : Not Joined StarG   Up JP Rpt Override : 0d 00:00:00
Register State     : No Info
Reg From Anycast RP: No
Rpf Neighbor       : 10.100.1.1
Incoming Intf      : vprn_itf_D_2base
Outgoing Intf List : mpls-if-73881 (mpls-if-73887)
Curr Fwding Rate   : 1.3 kbps
Forwarded Packets  : 140                Discarded Packets  : 0
Forwarded Octets   : 6440               RPF Mismatches     : 0
Spt threshold      : 0 kbps             ECMP opt threshold : 7
Admin bandwidth    : 1 kbps
===============================================================================
PIM Source Group ipv4
===============================================================================
Group Address      : 239.100.0.3
Source Address     : 10.114.1.2
RP Address         : 10.100.1.1
Advt Router        :
Flags              : spt                Type              : (S,G)
Mode               : sparse
MRIB Next Hop      : 10.100.1.1
MRIB Src Flags     : remote
Keepalive Timer Exp: 0d 00:02:52
Up Time            : 0d 00:09:51        Resolved By       : rtable-u
Up JP State        : Joined             Up JP Expiry      : 0d 00:00:22
```

```
Up JP Rpt          : Not Joined StarG   Up JP Rpt Override : 0d 00:00:00
Register State     : No Info
Reg From Anycast RP: No
Rpf Neighbor       : 10.100.1.1
Incoming Intf      : vprn_itf_D_2base
Outgoing Intf List : mpls-if-73881 (mpls-if-73888)
Curr Fwding Rate   : 1.3 kbps
Forwarded Packets  : 140                Discarded Packets  : 0
Forwarded Octets   : 6440               RPF Mismatches     : 0
Spt threshold      : 0 kbps             ECMP opt threshold : 7
Admin bandwidth    : 1 kbps
-------------------------------------------------------------------------------
Groups : 4
===============================================================================
```

Step 8. Use the following command to display the PIM S-PMSI configuration information.

```
show router pim s-pmsi
```

Example

Display PIM S-PMSI configuration information

```
===============================================================================
PIM RSVP Spmsi tunnels
===============================================================================
P2mp  Tunnel ID   Ext Tunnel Adrs       SPMSI Index   Num State   Multistre
ID                                                    VPN SGs         am-ID
-------------------------------------------------------------------------------
1     61444       10.20.1.4             73885         1 Up      0
1     61445       10.20.1.4             73886         1 Up      0
1     61446       10.20.1.4             73887         1 Up      0
1     61447       10.20.1.4             73888         1 Up      0
===============================================================================
PIM RSVP Spmsi Interfaces : 4
===============================================================================
```

Step 9. Use the following command to display the PIM S-PMSI detail configuration information.

```
show router pim s-pmsi detail
```

Example

Display PIM S-PMSI detail configuration information

```
===============================================================================
PIM RSVP Spmsi tunnels
===============================================================================
P2MP ID            : 1                  Tunnel ID          : 61444
Ext Tunnnel Addrs  : 10.20.1.4          Spmsi IfIndex      : 73885
Number of VPN SGs  : 1                  Up Time            : 0d 00:00:09
VPN Group Address  : 239.100.0.0
VPN Source Address : 10.114.1.2
Up Time            : 0d 00:00:09        Multistream-Id     : N/A
State              : TX Joined          Mdt Threshold      : 1
Join Timer         : N/A                Holddown Timer     : 0d 00:00:50
===============================================================================
PIM RSVP Spmsi tunnels
===============================================================================
P2MP ID            : 1                  Tunnel ID          : 61445
Ext Tunnnel Addrs  : 10.20.1.4          Spmsi IfIndex      : 73886
Number of VPN SGs  : 1                  Up Time            : 0d 00:00:09
```

```
VPN Group Address  : 239.100.0.1
VPN Source Address : 100.114.1.2
Up Time            : 0d 00:00:09        Multistream-Id     : N/A
State              : TX Joined          Mdt Threshold      : 1
Join Timer         : N/A                Holddown Timer     : 0d 00:00:50
===============================================================================
PIM RSVP Spmsi tunnels
===============================================================================
P2MP ID            : 1                  Tunnel ID          : 61446
Ext Tunnnel Addrs  : 10.20.1.4          Spmsi IfIndex      : 73887
Number of VPN SGs  : 1                  Up Time            : 0d 00:00:09
VPN Group Address  : 239.100.0.2
VPN Source Address : 10.114.1.2
Up Time            : 0d 00:00:09        Multistream-Id     : N/A
State              : TX Joined          Mdt Threshold      : 1
Join Timer         : N/A                Holddown Timer     : 0d 00:00:50
===============================================================================
PIM RSVP Spmsi tunnels
===============================================================================
P2MP ID            : 1                  Tunnel ID          : 61447
Ext Tunnnel Addrs  : 10.20.1.4          Spmsi IfIndex      : 73888
Number of VPN SGs  : 1                  Up Time            : 0d 00:00:10
VPN Group Address  : 239.100.0.3
VPN Source Address : 10.114.1.2
Up Time            : 0d 00:00:10        Multistream-Id     : N/A
State              : TX Joined          Mdt Threshold      : 1
Join Timer         : N/A                Holddown Timer     : 0d 00:00:49
===============================================================================
PIM RSVP Spmsi Interfaces : 4
===============================================================================
```

Step 10. Use the following command to display the PIM tunnel-interface configuration information.

```
show router pim tunnel-interface
```

Example

Display PIM tunnel-interface configuration information

```
===============================================================================
PIM Interfaces ipv4
===============================================================================
Interface                       Originator Address   Adm  Opr Transport
Type
-------------------------------------------------------------------------------
mpls-if-73881                   10.20.1.4            Up   Up Tx-IPMSI
mpls-if-73882                   10.20.1.3            Up   Up Rx-IPMSI
mpls-if-73883                   10.20.1.2            Up   Up Rx-IPMSI
mpls-if-73884                   10.20.1.5            Up   Up Rx-IPMSI
mpls-if-73885                   10.20.1.4            Up   Up Tx-SPMSI
mpls-if-73886                   10.20.1.4            Up   Up Tx-SPMSI
mpls-if-73887                   10.20.1.4            Up   Up Tx-SPMSI
mpls-if-73888                   10.20.1.4            Up   Up Tx-SPMSI
-------------------------------------------------------------------------------
Interfaces : 8
===============================================================================
```

11 BIER

11.1 BIER overview

Bit Indexed Explicit Replication (BIER) architecture allows optimal forwarding of multicast packets without requiring a legacy multicast protocol to build multicast trees or for intermediate routers to maintain any per-multicast flow state. This provides a simplified control plane because BIER information is distributed using underlay IGP.

The following terms are used in BIER:

- **Bit Forwarding Router (BFR)**

  A BFR is a router supporting BIER with a unique BFR prefix and optionally, a BIER ID assigned by the user. A BFR establishes BFR adjacencies (IGP or SDN-programmed), computes the BIER routing table, and forwards or replicates BIER packets.

- **BIER domain and sub-domain (SD)**

  A BIER domain is a connected set of BFRs, each with a unique BFR ID. A BIER domain can be divided into sub-domains for scalability without a linear increase in size of the BIER header. For example, in IS-IS, a BIER sub-domain is IS-IS multi topology, where ipv4-unicast is a single sub-domain and ipv4-multicast is another sub-domain.

  Sub-domains provide minimum traffic engineering and separation of services.

- **Bit Forwarding Ingress Router (BFIR)**

  A BFIR is the first PE in a BIER domain entered by a multicast packet. The BFIR adds a BIER header and forwards the packet using the BIER routing table.

- **Bit Forwarding Egress Router (BFER)**

  A BFER is the last PE that processes a BIER packet in a BIER domain. The BFER removes the BIER header before forwarding the packet. This is the only PE that requires a BIER ID as it is a PE with receiver connectivity.

- **Transit Bit Forwarding Router (transit BFR)**

  A transit BFT is a router in the BFR domain that is not a BFIR or a BFER that forwards the packet using the best path.

SR OS does not support multicast MPLS packets over an IGP shortcut. This includes BIER MPLS encapsulation. IGP shortcuts can be configured on SR OS for unicast and installed in the RIB or FIB, but BIER is not resolved over the IGP shortcut. If an IGP shortcut is used for unicast resolution, an IPv4-multicast MT can be used to create a separate MT for BIER without the IGP shortcut.

11.1.1 BIER hardware

BIER is only supported on FP4 and FP5 network interfaces. BIER is not supported on FP3 or earlier cards, or on access interfaces.

If a chassis has a mix of FP3, FP4, and FP5 network ports, BIER is signaled on all FP3, FP4, and FP5 interfaces that are part of the IS-IS. From a control plane perspective, BIER TLVs are advertised using IS-IS on FP3, FP4, and FP5 interfaces. The BIER forwarding table is not downloaded to FP3 cards. Therefore, there is no BIER packet forwarding or processing on these cards. If IGP chooses FP3 L3 interfaces, there are BIER forwarding issues. An event log is generated if an FP3 is part of the BIER sub-domain.

BIER-encapsulated multicast traffic can egress only on FP4 or FP5 interfaces. Non-FP4 or non-FP5 egress interfaces do not forward BIER-encapsulated multicast traffic.

11.1.2 BIER IMPM

BIER supports IMPM and all the IMPM rules apply to BIER also. Users can use IMPM to optimize and increase BIER throughput.

11.1.3 BIER ECMP

BIER supports ECMP/LAG. SR OS only uses the smallest IP address in the ECMP/LAG group to resolve the BFRs.

After an ECMP switch, IGP must download the BIER forwarding table to the new interface or card so BIER ECMP does not have a sub-millisecond recovery. The recovery time is in line with IGP convergence time.

11.1.4 BIER redundancy and resiliency

If BIER Fast Reroute (FRR) is not enabled, when there is a failure on the primary next-hop BIER does not switch to the protection next-hop (LFA) even if one exists. After the failure, BIER waits for IGP to converge and a new next-hop to be available, which means that traffic interruption is equal to the IGP convergence time. LFA is still configurable when BIER is enabled and can be used for other IP and MPLS functionality.

If BIER FRR is enabled, BIER uses the available underlay Loop-Free Alternates (LFA) to provide link protection to neighbors and the recovery time from link failure is less than the IGP convergence time.

11.1.4.1 BIER FRR

BIER FRR uses basic LFA only; it cannot use TI-LFA or remote-LFA. Also, currently BIER FRR only supports ISIS.

Use the commands in the following context to configure LFA:

- **MD-CLI**

```
configure router isis loopfree-alternate
```

- **classic CLI**

```
configure router isis loopfree-alternates
```

When FRR is enabled, link protection is enabled for all BIER neighbors that can find an underlay LFA. The neighbor BFR label is pushed first. Use the following command to enable BIER FRR.

```
configure router bier fast-reroute
```

Use the following command to display BIER routing information for the sub-domain.

```
show router bier routing sub-domain 0
```

In the following output example, next-hop 10.20.1.4 is protected through 10.20.1.2 using a basic LFA.

Output example: BIER routing information

```
===============================================================================
Destination Prefix                          Bfr-ID          Age
  Neighbor
    Nexthop
    Interface
-------------------------------------------------------------------------------
===============================================================================
BIER Routing Database Sub-Domain 0 BSL 256
===============================================================================
    10.20.1.1                                   1               0d 01:17:21
      10.20.1.2 (Backup)
        10.180.3.2
        ip-10.180.3.3
      10.20.1.4
        10.180.11.4
        ip-10.180.11.4

    10.20.1.2                                   2               0d 01:17:21
      10.20.1.2
        10.180.3.2
        ip-10.180.3.3
      10.20.1.4 (Backup)
        10.180.11.4
        ip-10.180.11.4

    10.20.1.4                                   0               0d 01:17:21
      10.20.1.4
        10.180.11.4
        ip-10.180.11.3
===============================================================================
Total (Sub-Domain 0): 3
===============================================================================
Total BIER Routing entries : 3
===============================================================================
```

Use the following command to display BIER forwarding information for the sub-domain.

```
show router bier forwarding sub-domain 0
```

Output example: BIER forwarding information

```
===============================================================================
  Neighbor
    Nexthop
    Interface
    [SI]: Label
       Forwarding Bit Mask
```

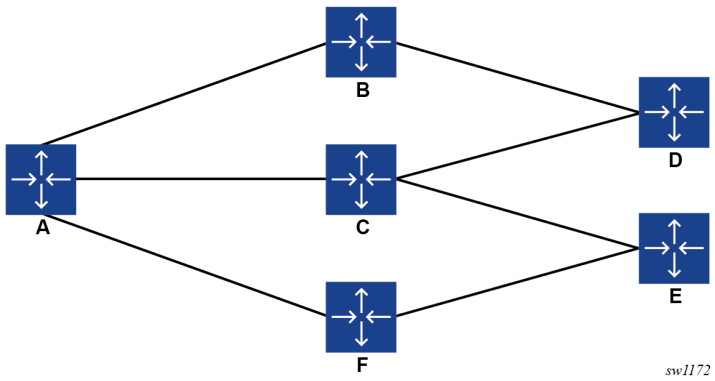
```
        BFR-ID : Prefix
-------------------------------------------------------------------------------
```

11.1.4.1.1 BIER FRR limitation

At the Point of Local Repair (PLR), when multiple leaves are reachable through the same next-hop, all the leaves must be protected through the same protection next-hop.

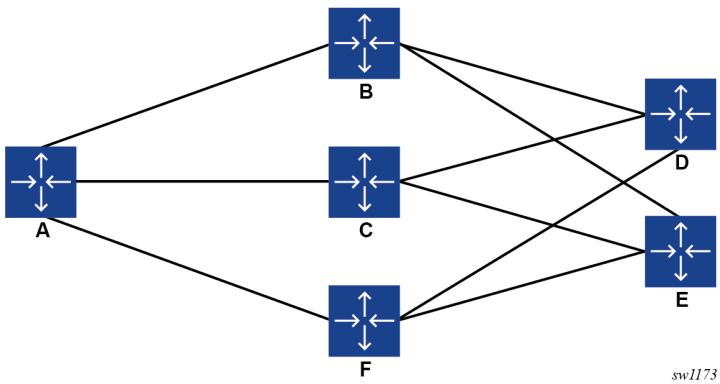
For example, in Figure 21: Topology example 1, from the point of view of A, both D and E are reachable through C. C is the next-hop for both D and E and can have a single protection next-hop, either B or F. If B is chosen as the protection next hop, for an LFA, the packet is forwarded to B for both D and E. B is not on the Shortest Path First (SPF) path to E and not the correct LFA, so an LFA for F is not possible in this network topology.

Figure 21: Topology example 1



If the network topology is changed as shown in Figure 22: Topology example 2, B can protect both D and E.

Figure 22: Topology example 2



11.1.4.1.2 BIER FRR with ECMP

When ECMP is enabled for two directly connected BIER routers, multiple next-hops are provided to BIER. BIER uses the next-hop with the smallest IP address for the primary path; all other next-hops can be used as protection paths. BIER uses as a protection next-hop for FRR all ECMP next-hops other than the primary ECMP next-hop. BIER does not hash the BIER packets over multiple ECMP next-hops.

BIER does not hash the BIER packets over multiple ECMP next-hops.

11.1.4.1.3 BIER BFD for next-hop failure detection

IGP Bi-directional Forwarding Detection (BFD) must be enabled on BIER neighbors that require FRR support. To register BIER with BFD, BFD must be enabled under BIER.

11.1.5 BIER layers

A multicast BIER network can be divided into three layers:

- routing underlay (IGP) has the following capabilities:
  - establishes BIER adjacencies based on BIER configuration
  - populates BIER routing table (best path reachability)
  - provides routing-underlay-based redundancy and convergence, ECMP
- BIER layer (BIER routing table, BIER header) has the following capabilities:
  - advertises and configures the BFR prefix and BIER ID (bitmask bit) for BIER routers
  - imposes a new BIER header (bitmask: "OR" for receiver PEs based on their BIER IDs as dictated by multicast flow overlay)
  - forwards multicast traffic using the BIER header and BIER routing table
  - prevents loops and duplication by using bitmask manipulation and removing the bits for PEs that are not reachable using the L3 interface next hop
- Multicast flow overlay (MVPN, BGP) uses MP-BGP to distribute and discover the endpoints (RFC-6513 and RFC-6514).

11.1.6 Implementation

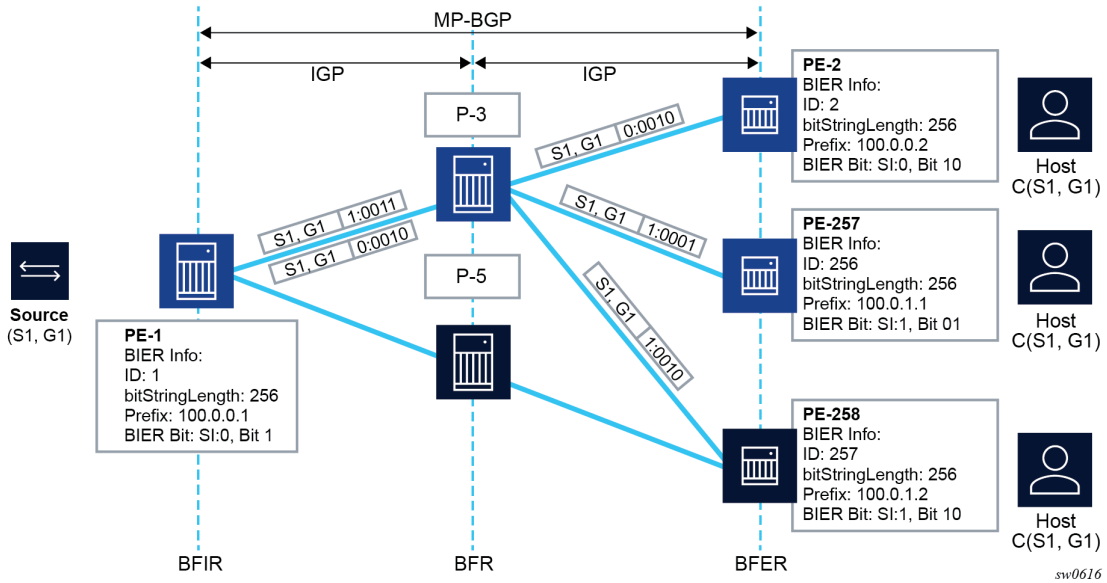
Figure 23: BIER high-level IGP and overlay shows multicast with BIER deployed. IGP is used as the routing underlay, and MP-BGP for NG-MVPN is used as the multicast flow overlay. The BFIR is the source PE-1, BFERs 2, 256, and 257 are receiver PEs, and the remaining routers are BFRs. All routers have their BIER prefix assigned and, additionally, the BFERs have BIER BFR-IDs assigned.

A BFR prefix is a unicast routable IP address (either IPv4 or IPv6) that is either a system loopback or a loopback interface. BFR prefixes are unique within a BIER domain.

A BFR ID is a unique number assigned to BFERs and BFIRs that is used to build the BIER bitmask used to forward packets. BIER IDs should be allocated as a continuous set of IDs starting at 1 to ensure a minimum number of sets are required to achieve multicast BIER connectivity. Sets allow scaling of BIER

beyond the bitmask length supported; however, sets require a separate copy of the multicast packet to be forwarded on the same link which may result in unwanted replication.

Figure 23: BIER high-level IGP and overlay



11.1.6.1 BIER Sub-domains

Each BIER domain can be divided into sub-domains. A BIER domain supports sub-domains numbered from 0 to 255. Each BIER domain must contain at least one sub-domain, and sub-domain 0 is the default. If a BIER domain contains more than one sub-domain, each BFR in the domain must be provisioned with the set of sub-domains to which it belongs.

A BIER domain is an IGP area, and sub-domains are the different topologies within that area. In IS-IS and OSPF, each topology must also have its own sub-domain ID. For example, in IS-IS a sub-domain is an IS-IS multitopology. SR OS supports two sub-domains in IS-IS: IPv4-multicast and IPv4-unicast MTs. A sub-domain creates the least traffic engineering in a BIER domain. A user can use separate L3 interfaces into IPv4-unicast MT and a set of disjointed interfaces into the IPv4-multicast MT. This creates separation and traffic engineering for different multicast streams.

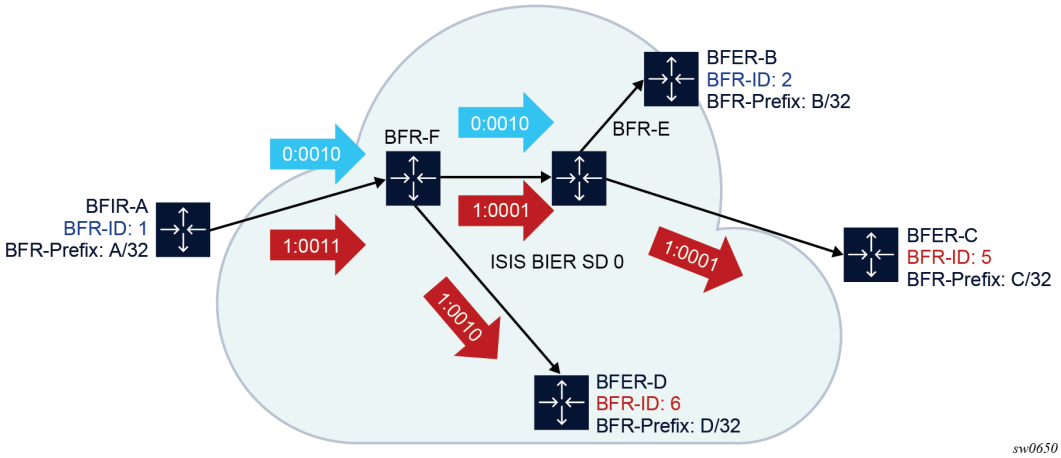
For each sub-domain to which a specific BFR belongs, if the BFR is capable of acting as a BFIR or a BFER, it must be provisioned with a BFR ID that is unique within the sub-domain. If a given BFR belongs to more than one sub-domain, it may have a different BFR ID for each sub-domain but this is not required.

11.1.6.2 BIER set IDs

To increase scalability of BIT String Length (BSL), routers can be grouped into BIER sets.

The BSL dictates how many BFRs can be represented in a BIER set. Each BIER set can contain as many routers as the length of BSL, and it is represented by a BIER Set ID (SI). The Set ID is part of the packet and represented as <SI:Bit Position>. Figure 24: BIER set shows an example set with a BSL of 4.

Figure 24: BIER set



The BFR ID is programmed into <SI, Bit Position> based on the network BSL.

SI = (BFR-ID – 1)/BSL

BP = ((BFR-ID – 1) mod BSL)+1

For example: BSL 4 and BFR-ID 6 = <SI=1, BP=2>.

BIER works well in an IP TV deployment where the network is in a spine and leaf deployment. SR OS supports 16 set IDs in this type of deployment where there is no packet duplication at the spine.

- The SHO can be connected to as many as 16 VHOs.
- Each tree can have 256 LEAFs without packet duplication.
- Each leaf can have as many hosts on it as the number of supported IGMP/MLD hosts.

11.1.6.3 BIER encapsulation

SR OS supports BIER MPLS encapsulation.

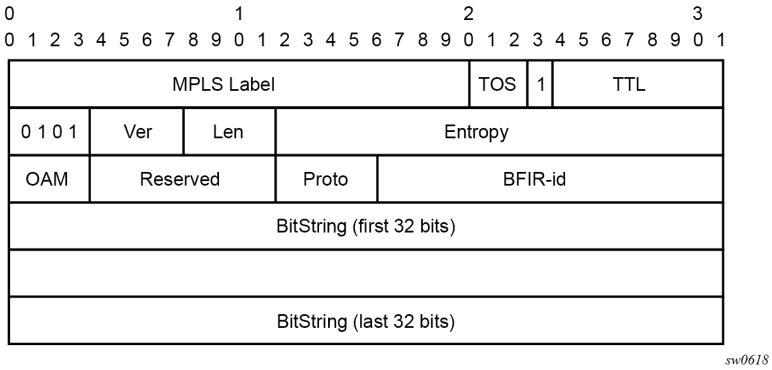
11.1.6.3.1 BIER MPLS encapsulation

The BIER MPLS labels are downstream-assigned MPLS labels that are unique only to the BFR that advertises them. BIER MPLS labels can be advertised using IGP (IS-IS) extension sub-TLVs or BGP extension sub-TLVs.

Penultimate Hop Popping (PHP) is not supported by BIER-MPLS labels as the labels are used to identify the BIER forwarding table that packets need to be looked up in.

Figure 25: BIER MPLS encapsulation shows the BIER MPLS encapsulation label.

Figure 25: BIER MPLS encapsulation



A BIER MPLS label is bound to the forwarding element class. A BIER label is assigned per BIER <SD, <BSL, SI>>. The SR OS supports only a BSL of 256.

Labels are chosen from the first available label in the label pool, and are only allocated locally when IGP advertises the BIER sub-TLVs.

When a packet arrives on a BFR the BIER forwarding table is identified using the MPLS label. BIER forwarding is then completed using the BIER header.

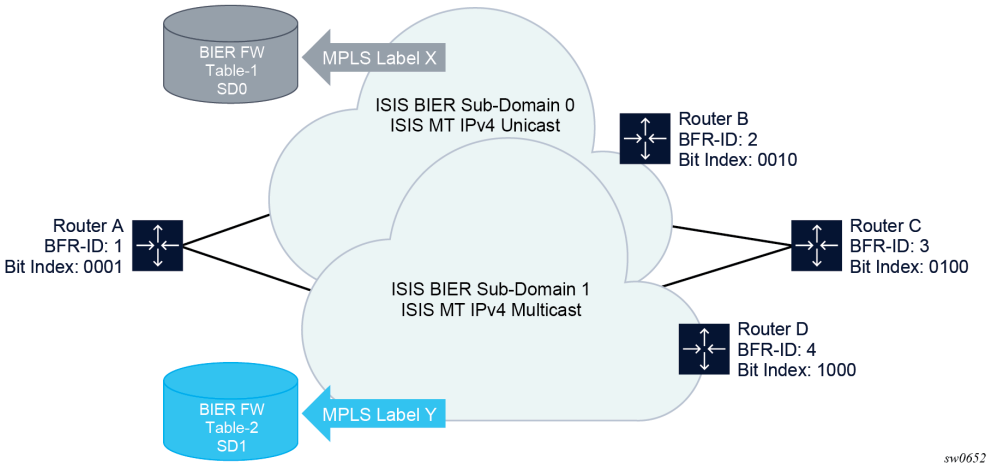
11.1.6.4 BIER forwarding tables

A BIER forwarding table is built based on the combination of:

- Set ID (SI)
- BIER String Length (BSL)
- Sub Domain (SD)

and saved in the format <SD, BSL, SI>. Figure 26: BIER forwarding tables shows an example of how forwarding tables are built.

Figure 26: BIER forwarding tables



For example, if there are 2 SDs and there are 256 PEs in each SD, there are two forwarding tables, one for each SD. One for <SD=0, BSL=256, SI=0> and the other for <SD=1, BSL=256, SI=0>.

Similarly, if there are 512 PEs in an SD and the BSL is 256, there are two forwarding tables, one for <SD=0, BSL=256, SI=0> and the other for <SD=0, BSL=256, SI=1>.

An MPLS label is assigned locally for each BIER routing table, and advertised.

11.1.6.5 BIER IS-IS sub-TLVs

BFRs establish BIER adjacencies through IS-IS and exchange their BFR prefixes and BIER IDs as well as transport-related information. IS-IS can be used to exchange the information. Figure 27: BIER IGP sub-TLV shows the IS-IS extensions for BIER and Figure 28: BIER MPLS sub-sub-TLV shows a BIER MPLS sub-sub-TLV.

Figure 27: BIER IGP sub-TLV

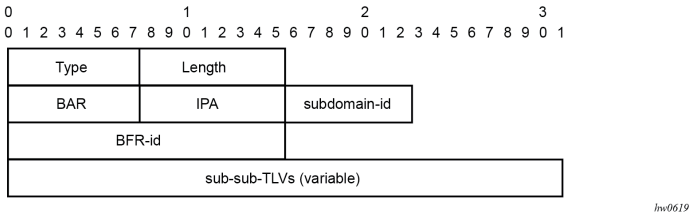
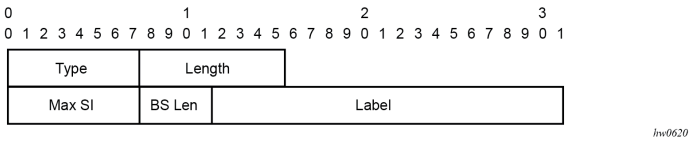


Figure 28: BIER MPLS sub-sub-TLV



In IS-IS, a new BIER sub-TLV is advertised as part of extended prefix opaque LSA carrying the BFR IP address (loopback) and supported BIER bitmask length for this BFR (multiple TLVs are used to convey support for multiple bitmask lengths). In addition, when MPLS encapsulation is used, a BIER MPLS encapsulation sub-TLV is included that contains the label range used for BIER. The label ranges advertised within the area are unique to a BFR and are used to identify the BIER forwarding context.

Based on the information exchanged, IGP creates a BIER routing table (unicast SPF) to reach each BFER that can be used to route BIER packets. The routing table specifies the shortest unicast path to reach each BFER through (BFERs bitmask, next-hop BFR)-tuples.

BIER sub-TLVs having the wrong length or illegal encoding are ignored and no error is raised. All other sub-TLV or sub-sub-TLV validation is done by the BIER module.

11.1.6.6 IS-IS BIER support

IS-IS supports multiple levels and BIER is supported under each level using the following rules:

- If the ABR is not a BFIR or BFER, the BIER sub-TLV must be leaked between different levels (areas) at the ABR. A BIER template without a BFR ID must be on both levels.
- The ABR can support BFIR and BFER functionality. ABR does not support BIER header stitching.

- A single area can have level 1 and level 2. In this case, the same template can be programmed on both levels.

11.1.6.7 IS-IS multitopologies

IS-IS supports multitopologies (MT), such as ipv4-unicast, ipv4-multicast, ipv6-unicast, and ipv6-multicast. SR OS supports ipv4-unicast and ipv4-multicast MTs for BIER.

A sub-domain is supported within only one topology. The mapping is indicated by the pair <MT, SD>.

For example, the following combination of <MT, SD>, where MT 0 is IPv4 unicast and MT 3 is IPv4 multicast, are valid:

<MT=0, SD=0>

<MT=0, SD =1>

<MT=0, SD =2>

However, the following combination, where MT 0 is IPv4 unicast and MT 3 is IPv4 multicast, is invalid because an SD belongs to more than one MT:

<MT=0, SD=0>

<MT=3, SD=0>

IPv4-multicast imports routes into the multicast RTM and ipv4-unicast imports routes into the unicast RTM.

A BIER forwarding table (BIFT) is identified using a label. A BIER label is assigned per (<MT, SD>, SI, BSL) and therefore, different MTs point to different BIFTs.

The MT can be used to engineer multicast and BIER routes separately from unicast routes.

11.1.6.8 BIER intra-AS solution

For intra-AS solutions, ensure that the ABR loopback interface used as the BIER prefix is included in both Level 1 and Level 2, otherwise the BIER prefix is not resolved at the level into which the route was leaked.

Nokia recommends having the loopback interface used as the BIER prefix in Level1/Level2 for intra-AS solutions, which is the default configuration for SR OS.

11.1.6.9 OSPF BIER support

SR OS supports the BIER TLV for OSPF. Each node uses the TLVs described in RFC 8444 to propagate the BIER information to the entire network. The peer nodes use this information to build their BIER Index forwarding table (BIFT).

OSPF does not support MT. Single BIER sub-domain per area is supported. Inside an area, BIER and virtual links are mutually exclusive.

Note: Although OSPF does not support MT, you can create a BIER template with an SD using ipv4-multicast MT in the **configure router bier** context. However, the node does not use this SD and does not generate any OSPF TLV for this <SD, MT>. If the same BIER template contains an SD with ipv4-unicast MT, that SD is used in OSPF. Both IPv4-unicast and IPv4-multicast MT are supported for IS-IS (see IS-IS multitopologies).

Use the following command to configure BIER under OSPF.

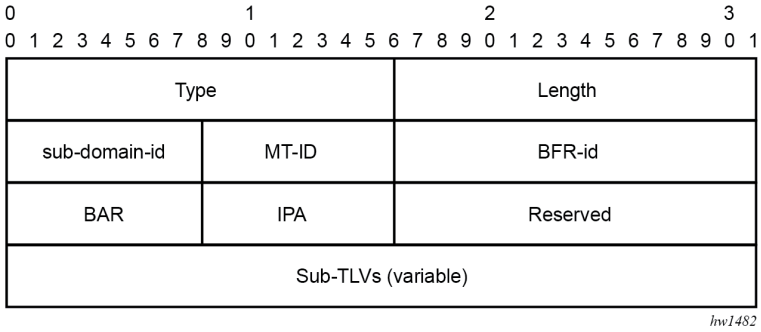
```
configure router ospf area bier template
```

11.1.6.9.1 BIER OSPF sub-TLVs

The BFRs establish the BIER adjacencies through OSPF and IS-IS, and exchange their BFR prefixes, BIER IDs, and transport-related information. OSPF can be used for this information exchange. The OSPF extensions for BIER and BIER MPLS sub-TLV are shown below. The peer OSPF routers use this information to build the BIER forwarding table.

The following figure shows the BIER sub-TLV format.

Figure 29: BIER sub-TLV format



Only the following values are supported for the BIER Algorithm (BAR) and, respectively, the IGP Algorithm (IPA): BAR=0 and IPA=0.

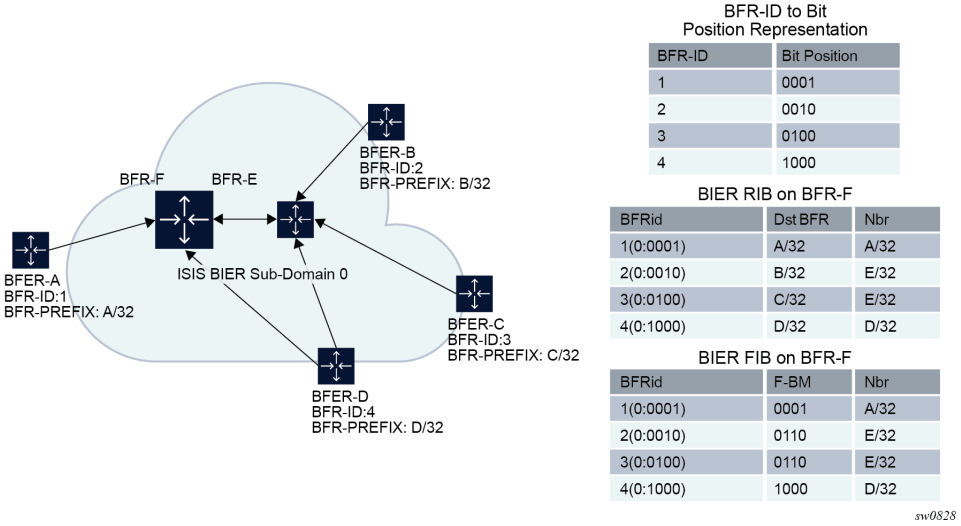
The following figure shows the BIER MPLS Encapsulation sub-TLV format.

Figure 30: BIER MPLS Encapsulation sub-TLV format

11.1.6.10 BIER forwarding

Figure 31: BIER forwarding shows that IGP builds the BIFT using IGP BIER TLVs. Each node also builds its LTN table based on IGP-advertised MPLS labels.

Figure 31: BIER forwarding



The BIER routing table is constructed based on the combination of:

- Set ID (SI)
- BIER String Length (BSL)
- Sub Domain (SD)

This information is presented as <SD, BSL, SI>. An MPLS label is assigned locally for each BIER routing table and is advertised using IGP to peers.

For example, if there are 512 PEs and the BSL is 256 there are two forwarding tables with each table having its own label as follows:

- <SD=0, BSL=256, SI=0> represented with a unique local label
- <SD=0, BSL = 256, SI=1> represented with a second unique local label

Each node is presented in the BIER header using its BFR-ID. It is recommended to assign the BFR-IDs sequentially and in a tight order for the PEs so that no bits in the BIER header remain unused.

After all BFRs forward their BIER information using IGP BIER TLV, each BFR builds its own BIER RIB and BIER FIB. In the Figure 31: BIER forwarding example, PE A is represented using BFR-ID 1 (0:0001) and PE B is represented using BFR-ID 2 (0:0010) and so on. Therefore, on BFR F, the routing table is built based on this information. In the Figure 31: BIER forwarding example, BFR A (0:0001) has a destination of A/32 and its neighbor also has a destination of A/32 because of it directly being attached to BFR F. BFR B (0:0010) has destination of B/32 but is reachable using BFR E.

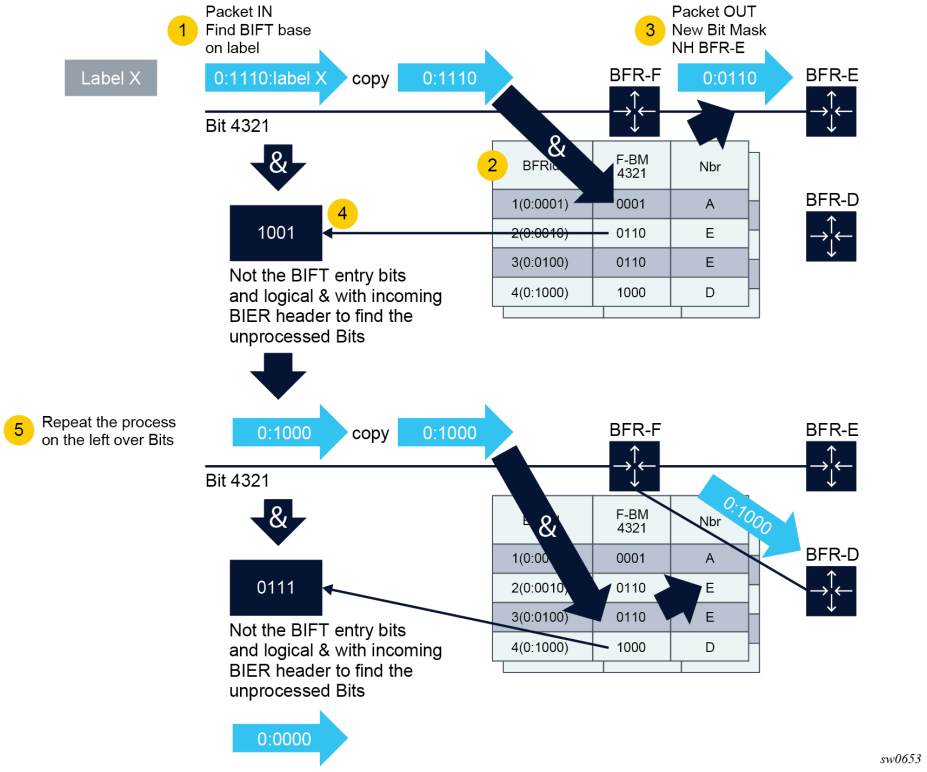
Therefore, the routing table for each PE is built based on its BFR-ID, destination IP address and the direct neighbor which it is attached.

The FIB is built based on the summation of the routes in the RIB that have the same neighbor. In the Figure 31: BIER forwarding example, BFER B and C have the same neighbor/peer BFR E. As a result, their FIB entry is identical BFR-ID 2 (0:0010) Forwarding bit mask 0110 neighbor/peer E/32; where the forwarding bit mask is a summation of BFR-ID nodes B and C (bit 2 and 3).

11.1.6.10.1 BIER FIB packet handling

Figure 32: BIER FIB packet handling shows how BIER packets are handled in the FIB.

Figure 32: BIER FIB packet handling



1. When a BIER packet arrives, the PE checks the label and finds the BIFT for that label and then pops the label.
2. For the incoming BIER header, it walks the bit index and finds the first entry in the BIFT for that bit position.
3. Using a logical "AND", the BIER header is combined with the BIFT bit-mask and forwarded to the neighbor. If there are multiple neighbors, the BIFT is programmed with a single entry, the neighbor with the smallest IP address.
4. Using a logical "NOT" on the BIFT bit-mask entry, the PE finds out which bits remain to be processed.
5. Repeat the process until all the BIER header's bits are processed.

11.1.6.11 BIER MVPN

Note: See "Intra-AS NG-MVPN over BIER" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 3 Services Advanced Configuration Guide for Classic CLI* for more information about advanced configurations.
See "Intra-AS NG-MVPN over BIER" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 3 Services Advanced Configuration Guide for MD CLI* for more information about advanced configurations.

BIER MVPN uses MP-BGP as an overlay to signal MVPN. It uses RFC 6514 the same way P2MP RSVP-TE.

BIER MVPN introduces a new tunnel type of BIER (0x0B).

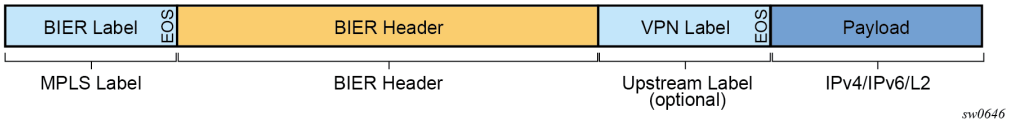
BIER PMSI replaces PIM, mLDP, and RSVP-TE P2MP in the core. There is no PMSI per (C-S, C-G), PMSI is used to reach all PE nodes interested in the C-Flow.

The VC label represents the VRF. Within the VRF, the payload IP header (C-S,C-G) finds the OIF based on PIM, IGP, and MLD states.

When a root PE (BFIR) receives a multicast packet and determines that the packet needs to be forwarded to the appropriate BFERs, the source PE encapsulates the multicast packet in a BIER header as described in BIER encapsulation. The root PE adds the appropriate VC label advertised by MP-BGP and PTA and forwards it to the BIER domain.

Figure 33: BIER MVPN packet format shows the BIER MVPN packet format.

Figure 33: BIER MVPN packet format



The original packet has a VC label identifying the VRF added first, then the packet (MPLS payload) is forwarded using BIER PMSI by adding a BIER header identifying the BFERs and BIER label learned using IGP from the next-hop router, as described in BIER forwarding. Finally, when the packet arrives on the BFER, the BIER label is stripped, the BIFT is used to identify whether the packet needs to be handled by a corresponding VRF (because the bit in the header corresponds to the BFER BFR-ID). The BFER strips the BIER header, uses the VC label to identify the VRF instance, strips the VC label, and forwards the packet according to the legacy multicast protocols configured on the SAPs of the MVPN (for example, PIM, IGMP, and MLD OIFs).

SR OS supports BIER as I-PMSI and S-PMSI. By default, all (C-S, C-G) are forwarded using I-PMSI. If a throughput threshold is configured in MVPN and that threshold is surpassed by a (C-S, C-G), then the traffic for that stream is switched from I-PMSI to S-PMSI. BIER uses standard NG-MVPN signaling for S-PMSI and uses leaf AD routes from the leaf PEs to set up S-PMSI to the corresponding leaf that is interested in a specific (C-S, C-G).

The following example shows a BIER MVPN configuration.

Example: MD-CLI

```
[ex:/configure service vprn "1" mvpn]
A:admin@node-2# info
    c-mcast-signaling bgp
```

```
    umh-selection hash-based
    auto-discovery {
        type bgp
    }
    vrf-target {
        unicast true
    }
    provider-tunnel {
        inclusive {
            bier {
                admin-state enable
                sub-domain 0
            }
        }
        selective {
            data-threshold {
                group-prefix 224.0.0.0/4 {
                    threshold 10
                }
            }
            bier {
                admin-state enable
                sub-domain 0
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config>service>vprn>mvpn# info
----------------------------------------------
        auto-discovery default
        c-mcast-signaling bgp
        umh-selection hash-based
        provider-tunnel
            inclusive
                bier
                    sub-domain 0
                    no shutdown
                exit
            exit
            selective
                bier
                    sub-domain 0
                    no shutdown
                exit
                data-threshold 224.0.0.0/4 10
            exit
        exit
        vrf-target unicast
        exit
----------------------------------------------
```

11.1.6.11.1 BIER MVPN IPv4 and IPv6

BIER MVPN only generates a single VC label and PMSI for IPv4 or IPv6 traffic belonging to the same VRF.

The BIER header protocol is set to "mpls packet with upstream-assigned label". This label is the VC label identifying the VRF that the packet belongs to. After finding the VRF and removing the VC label, a second

lookup on the IP header identifies the packet address family (IPv4 or IPv6). Based on the destination IP, which is the multicast group address, the packet is forwarded out the appropriate MVPN OIF SAPs.

11.1.6.11.2 BIER MVPN sub-domain

An MVPN belongs to a single sub-domain (SD). An SD is assigned to the PMSI of the MVPN, and forces the MVPN to resolve the BGP next-hop within that SD. Both I-PMSI and S-PMSI must be configured with the same SD. Different MVPNs can belong to different SDs. For example, mvpn-1 can belong to SD 0 which is an IPv4 unicast MT and mvpn-2 can belong to SD 1 which is an IPv4 multicast MT. This allows different MVPNs to be traffic engineered between different SDs or IS-IS MTs as needed.

11.1.6.11.3 BIER templates

A BIER template provides a centralized BIER configuration where the user can configure all the BIER parameters. The BIER template contains the sub-domain to multitopology mapping and other BIER configurations, such as the BFR ID and BIER prefix.

Each sub-domain can contain a single IGP Multitopology (MT). Currently, SR OS only supports MT for IS-IS but not for OSPF. A BIER template can contain many MTs and SDs. Each SD has its own BIER prefix and BFR ID and can belong to a different MT. The default MT is ipv4-unicast MT.

Use the commands in the following context to create a BIER template.

```
configure router bier template
```

After you configure a template you can assign it to a corresponding IGP protocol. The IGP protocol chooses the first <SD, MT> that matches its own configured MT. For example, if IS-IS has an MT of IPv4-multicast for the following example BIER template, it uses the sub-domain 1 configuration. It builds its BIER forwarding table base on SD 1 SI, BSL and uses the BIER prefix configured under sub-domain 1 for its IGP sub-TLVs.

Example: MD-CLI

```
[ex:/configure router "Base" bier template "bier1"]
A:admin@node-2# info
    admin-state enable
    sub-domain 0 {
        bfr-id 4096
        prefix 100.0.0.100
    }
    sub-domain 1 {
        bfr-id 1
        multi-topology ipv4-multicast
        prefix 100.0.0.101
    }
```

Example: classic CLI

```
A:node-2>config>router>bier>template# info
----------------------------------------------
       sub-domain 0
           prefix 100.0.0.100
           bfr-id 4096
       exit
       sub-domain 1
```

```
            prefix 100.0.0.101
            bfr-id 1
            mt ipv4-multicast
        exit
        no shutdown
    ----------------------------------------------
```

12 SR P2MP policy

The Segment Routing (SR) Point-to-Multipoint (P2MP) policy removes the need for the traditional underlay signaling layers like multipoint LDP (mLDP) and P2MP RSVP-TE. A P2MP policy can be instantiated statically using the CLI on the Path Computation Element Client (PCC), or instantiated dynamically using a Path Computation Element (PCE). The PCE uses the Path Computation Element Protocol (PCEP) to program the PCC.

The P2MP policy datapath and forwarding plane use MPLS instructions, similar to mLDP and P2MP RSVP-TE, and are programmed using a replication segment object. The replication segment is a forwarding entity with an incoming label and a set of Outgoing Interfaces (OIF) and labels. A P2MP policy can be used in a Next-Generation Multicast VPN (NG-MVPN) as a provider tunnel.

This functionality is described in the following IETF drafts:

- *draft-voyer-pim-sr-p2mp-policy*
- *draft-voyer-spring-sr-replication-policy*
- *draft-dhs-spring-sr-p2mp-policy-yang*
- *draft-hsd-pce-sr-p2mp-policy*
- *draft-hb-idr-sr-p2mp-policy*

12.1 SR P2MP policy details

A P2MP policy represents a multicast tree from the root node to a set of leaf nodes, and is a single provider tunnel. A P2MP policy can contain redundant trees from the root to leaf nodes, each with its own preference. This redundancy is implemented using Candidate Paths (CPs). Each CP represents a P2MP tree with its own Traffic Engineering (TE) constraints. The CPs can be optimized based on link failures or IGP optimizations. Each CP can contain multiple P2MP LSPs represented by path instance IDs. The CP can perform make-before-break between these path instances (P2MP LSPs).

A P2MP policy is relevant only on the root node where the P2MP tree is instantiated. The P2MP policy is identified by the tuple <root ID, tree ID>. A P2MP policy, does not include any forwarding information for the P2MP LSP. The policy only contains information about the root and leaf nodes and the TE, which is required to set up the tree from the root to the leaf nodes. The forwarding information is part of the replication segment. The root, transit, and leaf nodes contain replication segments.

12.2 Replication segment

A replication segment is the forwarding instruction for a P2MP LSP. It contains the incoming replication Segment Identifier (SID) and a set of OIFs and their corresponding SID or SID list. A replication segment is identified by a tree ID, root ID, and path instance ID (LSP ID) through the root, transit, and leaf nodes.

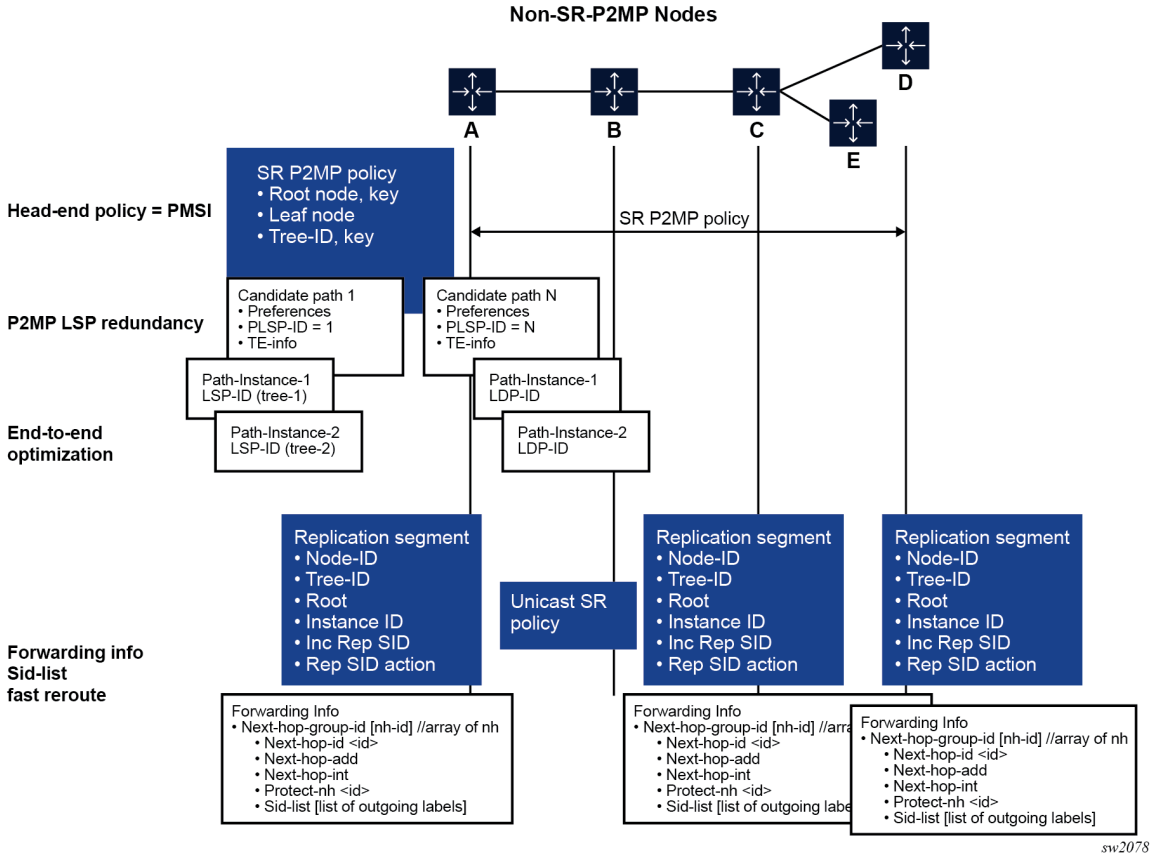
A second kind of replication segment is a shared replication segment. A shared replication segment is shared between multiple root nodes or P2MP LSPs. For this reason, it does not have a root ID but contains

a replication segment identifier, which is within the tree ID. A shared replication segment can be used for Fast Reroute (FRR). Currently, the P2MP policy supports a link protection facility bypass FRR.

12.3 P2MP and replication segment objects

In the following figure, nodes A, C, D, and E are replication-segment capable and B is unicast SR capable (that is, B is not replication-segment capable).

Figure 34: P2MP and replication segment objects



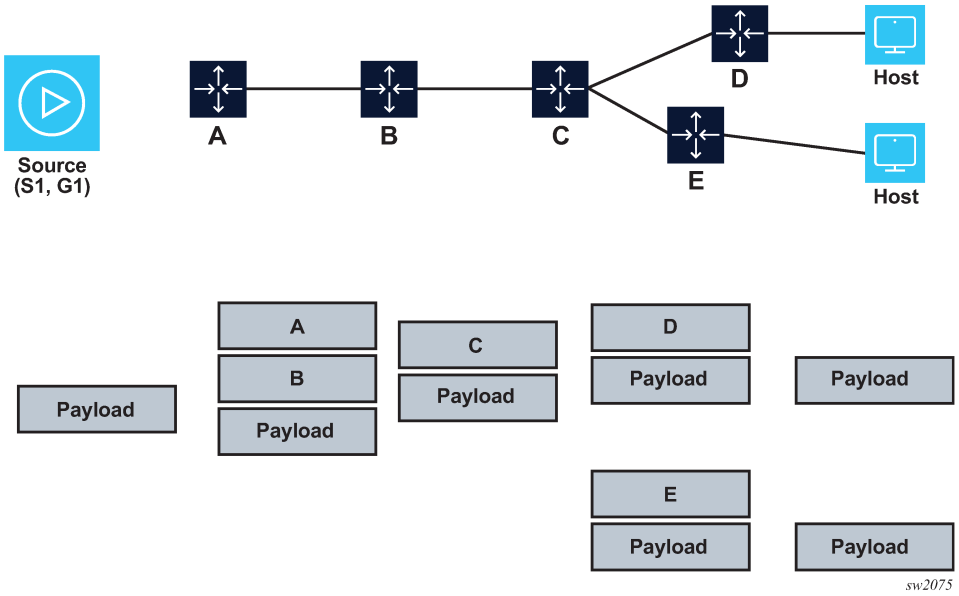
Multiple CPs can exist under a P2MP policy. However, the policy can have only one active CP, based on the CP preference. The highest CP preference is the active CP. The CPs act as tree redundancy.

Multiple path instances can exist under a CP. Each path instance is a P2MP LSP and each instance is presented with an instance ID. Path instances are used for global optimization of the active CP. Each path instance is built using replication segments, which forward P2MP tree information through the network at the root, transit, and leaf nodes. The P2MP policy is correlated to its replication segment by its root ID, tree ID, and instance ID.

The replication segments forward information with one or more OIFs to replicate and forward the PDUs. On the transit and leaf nodes, the incoming replication SID identifies the replication segment and its forwarding information. The replication segment can also contain FRR information for each outgoing interface.

In the following figure, the two replication segments on router A and C can be connected to each other using a unicast SR policy. To do so, the replication segment on router A is programmed with a SID list. The replication SID of router C is at the bottom of the stack and the SR labels connecting A to C are on top of the stack (that is, adjacency SIDs or node SIDs). Node B is not replication-policy capable, so node A pushes the SID list of B and C, where C is the replication SID at the bottom of the stack and B is the node SID.

Figure 35: Packet representation of a multicast stream



12.4 SR P2MP policy instantiation

The SR P2MP policy can be instantiated either statically on the PCC using the CLI, or dynamically using a PCE.

12.4.1 SR P2MP policy instantiation using the CLI

The CLI can be used to configure a P2MP policy, its CPs, and path instances on the root node. The CLI can also be used to create the replication segments on the root, transit, and leaf nodes. The P2MP policy can be assigned to NG-MVPN Inclusive P-Multicast Service Interfaces (IPMSIs) and Selective PMSIs (SPMSIs).

On each node, each replication segment represents a unique P2MP LSP with the following key: <tree ID, root ID, instance ID>. The instance ID and tree ID are unique to each root.

12.4.1.1 SPMSI for static P2MP policy

For a static P2MP policy, a single P2MP policy is assigned to an SPMSI. All the (S,G)s that are required to switch to the SPMSI and send an SPMSI AD route use this single P2MP policy. To assign (S,G)s to a different P2MP policy, use multi-stream SPMSIs and assign different (S,G)s to different SPMSIs.

12.4.1.2 PMSI tree ID advertised by BGP

The tree ID used in MP-BGP to advertise the AD routes is inherited from the P2MP policy assignment to the provider tunnels; it is not generated automatically.

12.4.2 SR P2MP policy instantiation using PCE

NG-MVPN can be configured using the CLI or SNMP on the PCCs.

The root node discovers all leaf nodes through NG-MVPN. The root and leaf nodes information is updated to the PCE using PCEP.

The PCE calculates the shortest path from the root to the leaf nodes and takes into account any programmed constraints. The PCE has an end-to-end view of the network through BGP LS.

After calculating the tree, the PCE downloads the P2MP policy to the root and the replication policies to the root, transit, and leaf nodes.

Updates to the P2MP policy or the replication paths are calculated by the PCE and downloaded accordingly.

12.4.2.1 SPMSI for PCE P2MP policy

When the root node listens to the MP-BGP SPMSI AD routes and determines that a set of (S,G)s are interested in an SPMSI, it sends an update message to the PCE with the <tree ID, root ID> of the SPMSI AD route and the leaf that is interested in joining this SPMSI. The PCE uses this information to build a P2MP policy for that specific tree ID and downloads it to be used for an SPMSI.

12.4.3 Configuration examples

This section provides examples to configure an MVPN service with SR P2MP policies for SPMSIs and IPMSIs. The configuration examples use the following network.

```
node-2-(100.101.1.x/24)---P-(101.102.1.x/24)----Leaf-1
|--(101.103.1.x/24)----Leaf-2
```

The following are the network settings:

- Node-2 is 100.0.0.100.
- P is 100.0.0.101.
- Leaf-1 is 100.0.0.102.
- Leaf-2 is 100.0.0.103.

The following example shows a reserved label block configured (in this example, "treeSID"), from which the Tree Segment ID (tree-SID) labels are allocated.

Example: Configure a reserved label block for MPLS labels (MD-CLI)

```
[ex:/configure router "Base" mpls-labels]
A:admin@node-2# info
    reserved-label-block "treeSID" {
        start-label 30000
        end-label 30999
    }
```

Example: Configure a reserved label block for MPLS labels (classic CLI)

```
A:node-2>config>router>mpls-labels# info
----------------------------------------------
            reserved-label-block "treeSID"
                start-label 30000 end-label 30999
            exit
```

The following example shows the reserved label block enabled for SR P2MP.

Example: Enable a reserved label block for SR P2MP (MD-CLI)

```
[ex:/configure router "Base" p2mp-sr-tree]
A:admin@node-2# info
    reserved-label-block "treeSID"
```

Example: Enable a reserved label block for SR P2MP (classic CLI)

```
A:node-2>configure>router>p2mp-sr-tree# info
----------------------------------------------
        reserved-lbl-block "treeSID"
```

The following example shows a P2MP policy configured with a tree on the node-2 PE. Two policies are configured, one for an IPMSI and one for an SPMSI.

Example: Configure a P2MP policy on the node with associated CP and path instance (MD-CLI)

```
[ex:/configure router "Base" p2mp-sr-tree]
A:admin@node-2# info
...
    admin-state enable
    p2mp-policy "IPMSI-VPRN1" {
        root-address 100.0.0.100
        tree-id 9000
        candidate-path "Primary-path" {
            active-instance 1
            preference 1000
            path-instances 1 {
                instance-id 1000
            }
        }
    }
    p2mp-policy "SPMSI-VPRN1" {
        root-address 100.0.0.100
        tree-id 9001
        candidate-path "Primary-path" {
```

```
            active-instance 1
            preference 1000
            path-instances 1 {
                instance-id 1000
            }
        }
    }
```

Example: Configure a P2MP policy on the node with associated CP and path instance (classic CLI)

```
A:node-2>config>router>p2mp-sr-tree# info
----------------------------------------------
...
            p2mp-policy "IPMSI-VPRN1"
                root-address 100.0.0.100
                tree-id 9000
                candidate-path "Primary-path"
                    preference 1000
                    path-instances
                        index 1 instance-id 1000
                    exit
                    active-instance 1
                    shutdown
                exit
                shutdown
            exit
            p2mp-policy "SPMSI-VPRN1"
                root-address 100.0.0.100
                tree-id 9001
                candidate-path "Primary-path"
                    preference 1000
                    path-instances
                        index 1 instance-id 1000
                    exit
                    active-instance 1
                    shutdown
                exit
                shutdown
            exit
            no shutdown
```

The following examples shows the P2MP policies assigned to NG-MVPN on the node. BGP must be established between the node-2 and leaf routers using IPv4/IPv6 MVPN Assured Forwarding (AF). The tree ID configured in the P2MP policies is used to advertise in the BGP Provider Tunnel Attribute (PTA) field.

Example: Assign P2MP policies to NG-MVPN (MD-CLI)

```
[ex:/configure service vprn "1"]
A:admin@node-2# info
    admin-state enable
    customer "1"
    pim {
        rp {
            ipv4 {
                bsr-candidate {
                    admin-state enable
                }
                rp-candidate {
                   admin-state enable
```

```
                }
            }
        }
    }
    mvpn {
        c-mcast-signaling bgp
        auto-discovery {
            type bgp
        }
        vrf-target {
            unicast true
        }
        provider-tunnel {
            inclusive {
                p2mp-sr {
                    admin-state enable
                    static-policy-mpls "IPMSI-VPRN1"
                }
            }
            selective {
                data-threshold {
                    group-prefix 231.0.0.0/24 {
                        threshold 10
                    }
                }
                p2mp-sr {
                    admin-state enable
                    static-policy-mpls "SPMSI-VPRN1"
                }
                data-threshold 231.0.0.0/24 10

            }
        }
    }
    bgp-ipvpn {
        mpls {
            admin-state enable
            route-distinguisher "70:70"
            vrf-target {
                community "target:70:70"
            }
            auto-bind-tunnel {
                resolution filter
                resolution-filter {
                    sr-isis true
                }
            }
        }
    }
    interface "to14" {
    }
```

Example: Assign P2MP policies to NG-MVPN (classic CLI)

```
A:node-2>config>service>vprn# info
----------------------------------------------
...
            interface "to14" create
            exit
            bgp-ipvpn
                mpls
                    auto-bind-tunnel
                        resolution-filter
```

```
                        sr-isis
                    exit
                    resolution filter
                exit
                route-distinguisher 70:70
                vrf-target target:70:70
                no shutdown
            exit
        exit            pim
            rp
                static
                exit
                bsr-candidate
                    no shutdown
                exit
                rp-candidate
                    no shutdown
                exit
            exit
            no shutdown
        exit
        mvpn
            auto-discovery default
            c-mcast-signaling bgp
            provider-tunnel
                inclusive
                    p2mp-sr
                        static-policy-mpls " IPMSI-VPRN1"
                        no shutdown
                    exit
                exit
                selective
                    p2mp-sr
                        static-policy-mpls " SPMSI-VPRN1"
                        no shutdown
                    exit
                    data-threshold 231.0.0.0/24 10
                exit
            exit
            vrf-target unicast
            exit
        exit
        no shutdown
```

You must configure the corresponding replication segment and forwarding instructions on the node, transit, and leaf nodes for the P2MP policies. The following example shows the configuration on the node.

Example: Configure replication segment and forwarding instructions on the node (MD-CLI)

```
[ex:/configure router "Base" p2mp-sr-tree]
A:admin@node-2# info
...
    replication-segment "IPMSI-VPRN1-NODE-2" {
        admin-state enable
        instance-id 1000
        root-address 100.0.0.100
        tree-id 9000
        segment-routing-mpls {
            sid-action push
            downstream-nodes 1 {
                admin-state enable
                next-hop-address "100.101.1.2"
                label {
```

```
                    sid-list 1 {
                        replication-sid 30000
                    }
                }
            }
        }
    }
    replication-segment "SPMSI-VPRN1-node-2" {
        admin-state enable
        instance-id 1000
        }        root-address 100.0.0.100
        tree-id 9001
        segment-routing-mpls {
            sid-action push
            downstream-nodes 1 {
                admin-state enable
                next-hop-address "100.101.1.2"
                label {
                    sid-list 1 {
                        replication-sid 30001
                    }
                }
            }
        }
    }
```

Example: Configure the replication segment and forwarding instructions on the node (classic CLI)

```
A:node-2>configure>router>p2mp-sr-tree# info
----------------------------------------------
...
        replication-segment "IPMSI-VPRN1-NODE-2"
            root-address 100.0.0.100
            tree-id 9000
            instance-id 1000
            segment-routing-mpls
                sid-action push
                downstream-nodes "1"
                    next-hop-address 100.101.1.2
                    replication-sid 30000
                    shutdown
                exit
            exit
            shutdown
        exit
        replication-segment "SPMSI-VPRN1-NODE-2"
            root-address 100.0.0.100
            tree-id 9001
            instance-id 1000
            segment-routing-mpls
                sid-action push
                downstream-nodes "1"
                    next-hop-address 100.101.1.2
                    replication-sid 30001
                    shutdown
                exit
            exit
            shutdown
        exit
        shutdown
```

The following example shows the configuration of the replication segment for IPMSI on the P router . The SPMSI replication router on the P router is configured in the same way, but with a tree ID of 9001 and an incoming SID of 30001.

Example: Configure a replication segment for IPMSI on the P router (MD-CLI)

```
[ex:/configure router "Base" p2mp-sr-tree]
A:admin@node-2# info
    admin-state enable
    reserved-label-block "treeSID"
    replication-segment "rs-IPMSI-VPRN1-100.0.0.100" {
        admin-state enable
        instance-id 1000
        root-address 100.0.0.100
        tree-id 9000
        segment-routing-mpls {
            incoming-sid 30000
            sid-action swap
            downstream-nodes 1 {
                admin-state enable
                # toleaf1
                next-hop-address "101.102.1.2"
                label {
                    sid-list 1 {
                        replication-sid 30000
                    }
                }
            }
            downstream-nodes 2 {
                admin-state enable
                # toleaf2
                next-hop-address "101.103.1.2"
                label {
                    sid-list 1 {
                        replication-sid 30000
                    }
                }
            }
        }
    }
```

Example: Configure a replication segment for IPMSI on the P router (classic CLI)

```
A:node-2>configure>router>p2mp-sr-tree# info
----------------------------------------------
...
        reserved-lbl-block "treeSID"
        replication-segment "rs-IPMSI-VPRN1-100.0.0.100"
            root-address 100.0.0.100
            tree-id 9000
            instance-id 1000
            segment-routing-mpls
                sid-action swap
                incoming-sid 30000
                downstream-nodes "1"
                    next-hop-address 101.102.1.2 //toleaf-1
                    replication-sid 30000
                    no shutdown
                exit
                downstream-nodes "2"
                    next-hop-address 101.103.1.2 //toleaf-2
                    replication-sid 30000
```

```
                    no shutdown
                exit
            exit
            no shutdown
        exit
        no shutdown
```

The following example shows the replication segment on the Leaf-1 router.

Example: Configure a replication segment for the Leaf-1 router (MD-CLI)

```
[ex:/configure router "Base" p2mp-sr-tree]
A:admin@node-2# info
    admin-state enable
    reserved-label-block "treeSID"
    replication-segment "rs-IPMSI-VPRN1-100.0.0.100" {
        admin-state enable
        instance-id 1000
        root-address 100.0.0.100
        tree-id 9000
        segment-routing-mpls {
            incoming-sid 30000
            sid-action pop
        }
    }
```

Example: Configure a replication segment for the Leaf-1 router (classic CLI)

```
A:node-2>configure>router>p2mp-sr-tree# info
----------------------------------------------
        reserved-lbl-block "treeSID"
        replication-segment "rs-IPMSI-VPRN1-100.0.0.100"
            root-address 100.0.0.100
            tree-id 9000
            instance-id 1000
            segment-routing-mpls
                sid-action pop
                incoming-sid 30000
            exit
            no shutdown
        exit
```

The following example shows the replication segment on the Leaf-2 router.

Example: Configure a replication segment for the Leaf-2 router (MD-CLI)

```
[ex:/configure router "Base" p2mp-sr-tree]
A:admin@node-2# info
    admin-state enable
    reserved-label-block "treeSID"
    replication-segment "rs-IPMSI-VPRN1-100.0.0.100" {
        admin-state enable
        instance-id 1000
        root-address 100.0.0.100
        tree-id 9000
        segment-routing-mpls {
            incoming-sid 30000
            sid-action pop
        }
    }
```

Example: Configure a replication segment for the Leaf-2 router (classic CLI)

```
A:node-2>configure>router>p2mp-sr-tree# info
----------------------------------------------
...
        reserved-lbl-block "treeSID"
        replication-segment "rs-IPMSI-VPRN1-100.0.0.100"
            root-address 100.0.0.100
            tree-id 9000
            instance-id 1000
            segment-routing-mpls
                sid-action pop
                incoming-sid 30000
            exit
            no shutdown
        exit
```

12.4.4 Administrative behavior of tree-SID

This section describes the administrative behavior of the SR P2MP policy and the CP.

12.4.4.1 Candidate path selection criteria

The active CP is chosen as follows.

1. The higher value protocol origin is selected.
2. The existing installed path is preferred.
3. The lower value of originator is selected.
4. The higher value of discriminator is selected.

The CLI static configuration has a protocol origin value of 30. This value is not configurable.

The CLI originator value is <0, 0.0.0.0>.

From the CLI perspective, the CP selection is as follows.

1. If the P2MP policy is operational and has an operational CP, the following handling applies.
   a. If a CP with a higher preference is configured and becomes operational, the users should switch to it.
   b. If the current CP is the highest preference and another CP with the same preference is configured, the users should stay on the current CP.
2. If the P2MP policy is administratively disabled, then administratively enabled, and there are multiple CPs with the same preference, the CP that was created last should be chosen. The **show** command for the CP displays the creation time.

12.4.4.2 Candidate path operational status

The CP operational status changes to down if at least one of the following conditions becomes true:

- The CP has no active instance.
- The CP is administratively disabled.

- The replication segment correlating to the CP on the root is operationally down or does not exist.

12.4.4.3 P2MP policy operational status

If at least one of the following conditions becomes true, the operational status of the P2MP policy and the status of the PMSI corresponding to the P2MP policy change to down:

- All the CPs in the P2MP policy are operationally down.
- The P2MP policy is administratively disabled.

12.4.4.4 Replication segment operational status

A replication segment is operationally down when one of the following is true:

- all its next-hops (NHs) are operationally down, including the FRR NHs
- no NHs are configured

The replication segment is operationally up if at least one NH is operationally up on the replication segment.

12.4.5 FRR behavior

Only facility bypass and link protection are supported for a P2MP policy. Node protection is not supported. The facility bypass can be created using shared replication segments. Shared replication segments do not have a tree ID. They are identified using a replication segment identifier within the tree ID.

At the Point of Local Repair (PLR), the primary replication segment has a protection next hop. This protection next-hop has a second OIF with its own outgoing label, which is used for the facility protection tunnel.

The facility protection tunnel can consist of multiple transit nodes until the tunnel reaches the Merge Point (MP). Replication segments are configured on these transit nodes to complete the facility protection tunnel. Multiple P2MP trees can share the facility protection tunnel at the PLR. The facility protection tunnel uses the implicit null label (see Implicit null case) or an actual label at the Penultimate Hop Popping (PHP) node (see Non-implicit null case).

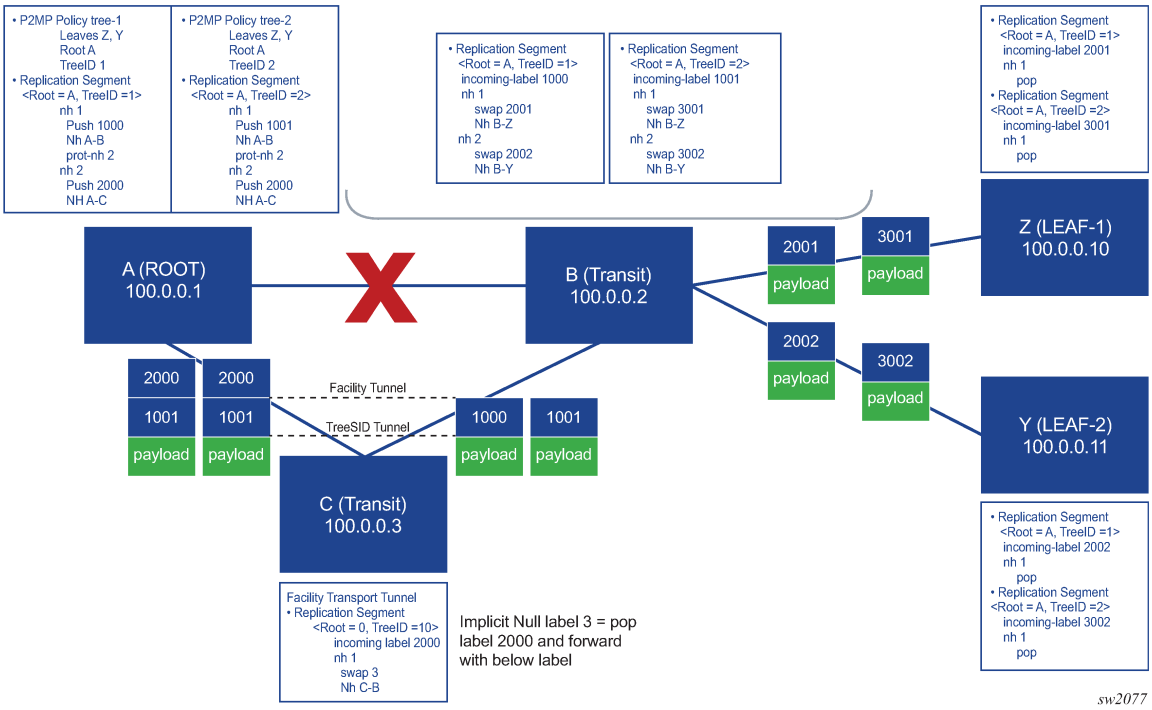
12.4.5.1 Implicit null case

If the implicit null label is used, the protection tunnel label is popped and the tree-SID P2MP LSPs are forwarded to the MP with the tree SID label.

In the following figure, node A is protecting link A-B through node C, so the facility protection tunnel is set up through node C. Node C is a PHP node and is programmed to swap the facility protection label with implicit null (label 3).

After a failure on the A-B link, node A pushes label 2000 as indicated in the protection next hop programmed in the replication segment for trees 1 and 2. Trees 1 and 2 share the same facility protection tunnel (label 2000). The packet is forwarded to node C, where node C pops the protection next hop and forwards the packet to node B with a replication SID on top of the packet.

Figure 36: Protection using the implicit null label



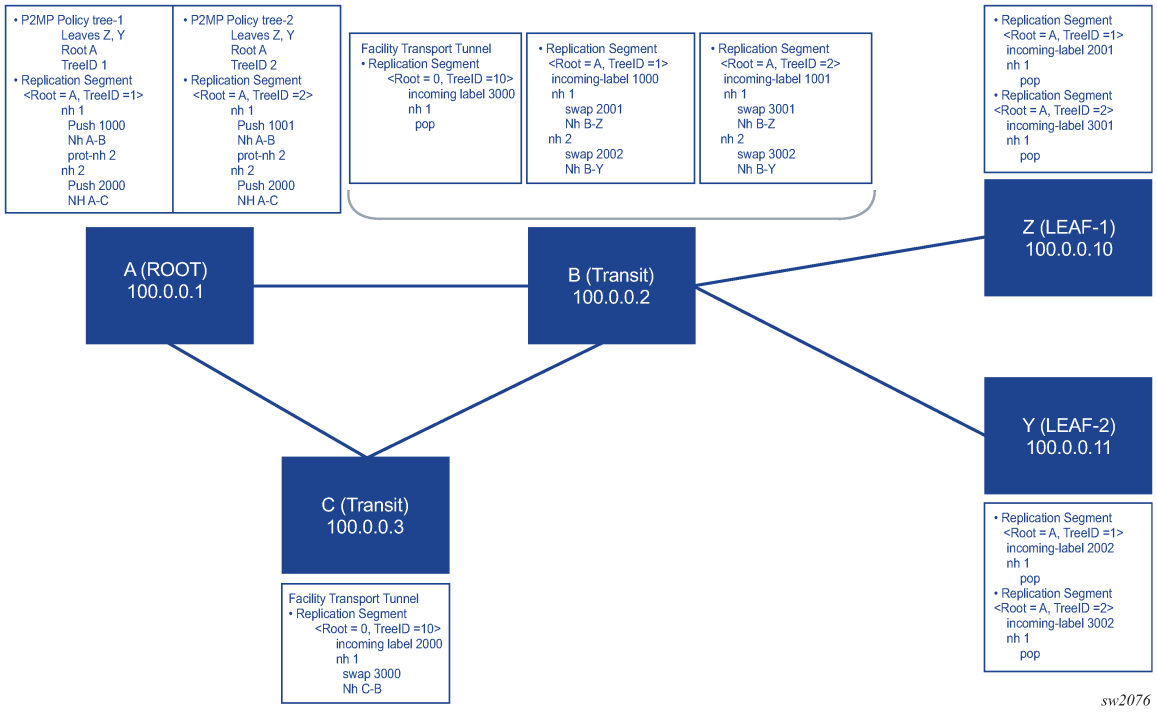
12.4.5.2 Non-implicit null case

If an implicit null label on the PHP node is not used, a replication segment is needed on the MP to pop the facility protection label and forward the underlying traffic based on the tree-SID label.

In the following figure, node A is protecting the A-B link through node C, so the facility protection tunnel is set up through node C. Node C is a PHP node and is programmed to swap the facility protection label with label 3000.

After a failure on link A-B, node A pushes label 2000 as indicated by the protection next-hop programmed in the replication segment for trees 1 and 2. The packet is forwarded to node C, where node C swaps the protection label with label 3000 for both tree 1 and tree 2, and forwards the packet to node B. Node B has a replication segment for the facility protection tunnel, which has an action of pop 3000. After popping 3000 on node B, the tree-SID label for label 1 and label 2 is exposed and the corresponding replication segment is found and executed.

Figure 37: Protection using the actual label at the PHP node



12.4.6 FRR recovery behavior

After the primary path is recovered, the P2MP LSP switches back to the primary path and away from the protection next-hop (FRR). This switch back to primary may cause a brief traffic outage.

The replication SID next-hops send optimistic ARP to populate the ARP table. If the next-hop MAC address is not found in the ARP table (that is, the address is not populated by BFD, IGP, or any other packet), the optimistic ARP populates the ARP table. The reversion from FRR to the primary next-hop happens only if the primary path ARP entry is found in the ARP table.

12.4.7 BFD behavior

Use the commands in the following context to enable BFD for the replication segment next hops.

```
configure router p2mp-sr-tree
```

BFD is enabled at the P2MP SR tree level. The P2MP SR tree registers itself with the current available BFD session. The BFD sessions need to be enabled using other protocols. For example, static route, OSPF, or IS-IS can enable the BFD session on an interface. The replication segments are registered with these BFD sessions. The replication segments cannot initiate a BFD session and rely on other protocols to initiate the BFD session because a replication segment is a unidirectional entity, while BFD is a bidirectional protocol.

When BFD is enabled under the **p2mp-sr-tree** context, all replication next hops that are using an L3 interface with BFD enabled on that interface register with the BFD module. If the BFD status on the L3 interface goes down, any replication segment next hop that is using that L3 interface goes operationally down. This operationally down status of the next hop within a replication segment can cause an FRR.

Only single-hop BFD is supported. BFD for unnumbered interfaces is not supported.

For IPv6, protocols such as OSPF or LDP create a BFD session to the link local interface. A static route can create a BFD session to link local or global IPv6 addresses. To use BFD for IPv6 next-hops within a replication segment, the replication segment needs to be configured with a link local next-hop for protocols that create the BFD session to the link local address. This way, the replication segment next hop finds a BFD session created by one of these protocols.

12.4.8 Maximum SPMSI behavior

The maximum P2MP SPMSI value configured under an MVPN selective provider tunnel does not affect any established SPMSIs. It only affects new spawning SPMSI counts.

If any existing SPMSIs are above the maximum P2MP SPMSI threshold, no new SPMSIs are spawned until the number of SPMSIs goes below the threshold.

12.4.9 Global optimization of P2MP policy and MBB behavior

Global optimization of the P2MP policy CP is supported, in addition to local FRR, where the protection next-hop is downloaded by the replication segment and the FRR action is triggered by a port failure or BFD failure.

To use the global optimization behavior, the user creates another instance under the P2MP policy. Appropriate replication segments must also be created for this optimized instance. After the entire tree is created, the active instance under the CP is set to this new, optimized instance and a switch from the previous instance to this optimized instance is performed. While the switch is in progress, the MVPN on the leaf is accepting traffic from both instances of the CP. After the switch is complete, the old instance can be deleted from the candidate path and its replication segments can be removed.

12.4.10 Global optimization of PCEP behavior

Global optimization is supported on the PCE. Local optimization of a replication segment using PCEP is not supported. If the PCE calculates an optimized path for a candidate path, that path instance is different from the current path instance. For this reason, a candidate path contains two path instances. The PCE must download a new path instance with an LSP ID of 0 and the PLSP ID of the current CP. This behavior applies to replication segments only.

When the current path instance is modified from the PCE to the PCC, the PCC-assigned LSP ID and PLSP ID are sent from the PCE to the PCC. This behavior ensures that the LSP ID of the replication segment for the existing path instances does not change.

12.4.11 PCEP behavior

You can clear all states on the PCC, including replication segments and P2MP policies.

The state of the P2MP LSP on the PCC is operationally up as long as there is one valid OIF and up for that LSP.

Use the following command to clear all states on the PCC:

- **MD-CLI**

```
configure router p2mp-sr-tree admin-state disable
```

- **classic CLI**

```
configure router p2mp-sr-tree shutdown
```

12.4.12 PCE pop with next-hop 127.0.0.0/8 or ::1

For a PCE, to program the datapath with a pop action, the next hops must be programmed as 127.0.0.0/8 or IPv6 ::1. If the replication segment next-hop has no information, the next hop is reported to the PCE with status down.

For CLI-initiated replication segments, the next-hop label action can be set to pop, and the next hop does not need to be programmed as 127.0.0.0/8 or ::1.

12.4.13 P2MP policy special considerations

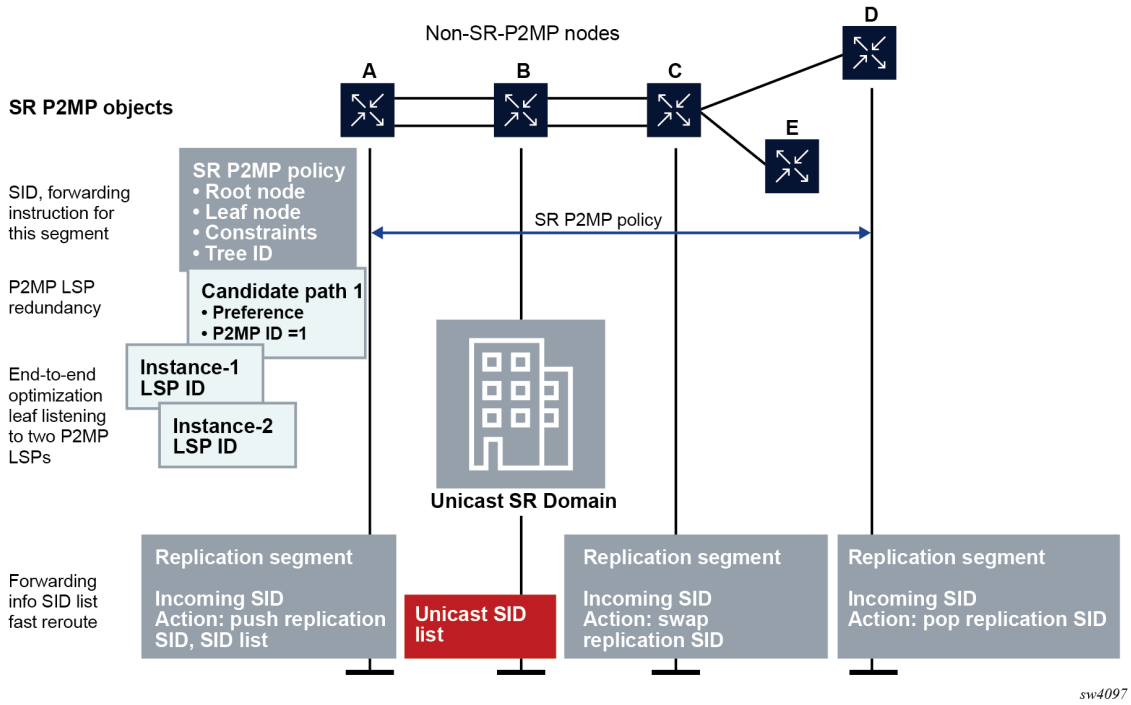
For interfaces that are not configured as unnumbered, the next-hop used in the replication segment must be a direct (local) next-hop. The replication segment cannot resolve indirect next-hops to a downstream router loopback or system IP address.

The FRR outage time can exceed 50 ms when a node has a large number of replication segments that are using a protection tunnel for FRR.

12.5 Replication segment steering through a unicast SR network

When a unicast SR network is present between two replication SIDs, it is possible to connect the two replication SIDs through a unicast SID list, as shown in the following figure. The SID list can be a list of adjacency or node SIDs that provides a traffic-engineered path though the unicast domain to connect the two replication SIDs.

Figure 38: Replication segment steering through a unicast SR network

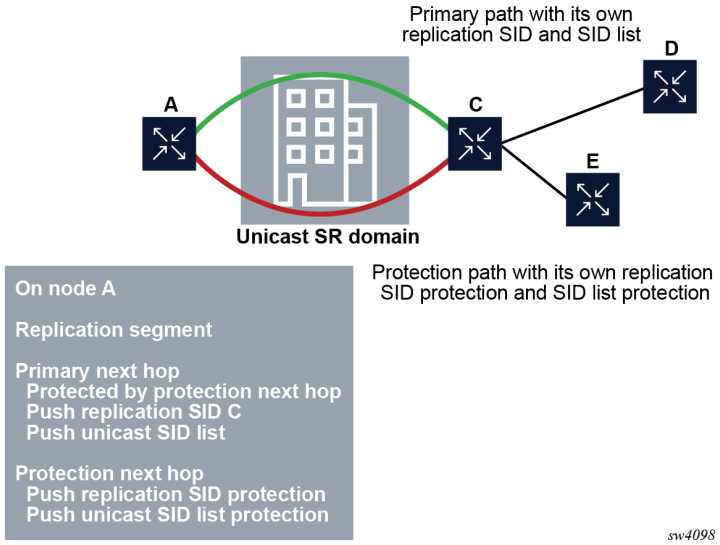


The unicast SID list can be configured by listing the node or adjacency SIDs under a replication segment. Even if two replication segments are connected directly, the egress interface or the next hop can be programmed using a SID list. For example, an adjacency SID can be used as a mechanism to steer the packet out of a local interface.

The unicast fast reroute functionality and Equal Cost Multipath (ECMP) functionality are not available on the egress replication segment node on which the SID list is configured. However, the next downstream node in the unicast SR domain can take advantage of all unicast SR TE and resiliency features, for example, Loop-Free Alternate (LFA), Remote Loop-Free Alternate (RLFA), or Topology-Independent Loop-Free Alternate (TI-LFA).

On the egress replication segment node, the protection next hop can be configured using a replication SID. In addition, a SID list can be used in any next-hop object under the replication segment, including the protection next-hop object, as shown in the following figure.

Figure 39: Primary and protection paths with SID list



12.6 Tree-SID OAM ping

Multiple candidate paths (CPs) can exist under a P2MP policy acting as tree redundancy, but only one CP can be active in the P2MP policy based on its CP preference. Multiple path instances can exist under a CP, with each path instance being a P2MP LSP and with each instance having an instance ID. SR OS can send ICMP ping packets over each of these path instances for both active and non-active instances on all CPs.

SR OS implementation supports version 3 of the *draft-ietf-pim-p2mp-policy-ping*.

Example: Tree-SID configuration (classic CLI)

```
A:node-2>config>router# info
...
#--------------------------------------------------
echo "TREE SID Configuration"
#--------------------------------------------------
            p2mp-sr-tree
                reserved-lbl-block "treeSID"
                p2mp-policy "ipmsi-1"
                    root-address 100.0.0.100
                    tree-id 9000
                    candidate-path "Primary-path-1"
                        preference 1000
                        path-instances
                            index 1 instance-id 1000
                        exit
                        active-instance 1
                        no shutdown
                    exit
                    candidate-path "Primary-path-2"
                        preference 500
                        path-instances
                            index 1 instance-id 1010
```

```
                        exit
                        path-instances
                            index 2 instance-id 1011
                        exit
                        active-instance 1
                        no shutdown
                    exit
                    no shutdown
                exit
            exit
...
```

Note: The following information for the **oam p2mp-lsp-ping** command only applies for the classic CLI.

Using the preceding configuration as an example, the **oam p2mp-lsp-ping** command can be used in the following ways.

Use the following command to test the "Primary-path-1" **instance-id** 1000.

```
A:node-2# oam p2mp-lsp-ping p2mp-policy root-address 10.0.0.100 root-tree-id 9000 instance-id
 1000
```

Use the following command to test the "Primary-path-2" **instance-id** 1010.

```
A:node-2# oam p2mp-lsp-ping p2mp-policy root-address 10.0.0.100 root-tree-id 9000 instance-id
 1010
```

Use the following command to test the "Primary-path-2" **instance-id** 1011.

```
A:node-2# oam p2mp-lsp-ping p2mp-policy root-address 10.0.0.100 root-tree-id 9000 instance-id
 1011
```

Example: P2MP SR policy ping output

```
A:node-2# oam p2mp-lsp-ping p2mp-policy root-address 10.0.0.100 tree-id 9000 instance-id
 1011 detail
88 bytes MPLS payload
================================================
LEAF Information
================================================
From            RTT                 Return Code
------------------------------------------------
10.20.1.2                       =2.59ms                                     EgressRtr(3)
10.20.1.1                       =2.68ms                                     EgressRtr(3)
10.20.1.6                       =3.03ms                                     EgressRtr(3)
10.20.1.5                       =4.89ms                                     EgressRtr(3)
================================================================================

Total Leafs responded = 4
          round-trip min/avg/max  = 2.59 / 3.29 / 4.89 ms

Responses based on return code:
      EgressRtr(3)=4
```

SID list support

The Tree-SID OAM feature can support the following:

- Testing a tree, with each replication segment directly connected to each other, and the outgoing label on each outgoing interface (OIF) being the replication SID.
- SID list or steering through an Adjacency Segment Identifier (Adj-SID). In this case, there is a SID list above the replication SID. This SID list can be a single Adj-SID used for steering the replication SID out of an interface or it can be a SID list with multiple node and Adj-SIDs used to connect two replication segments through a unicast domain.

12.7 Tree-SID SRv6

Implementation Overview

The SR OS supports tree-SID with encapsulation MPLS and SRv6 in the tree mode as follows:

- All the routers in a P2MP tree must support tree-SID SRv6.
- The replication routers cannot be connected via a unicast domain and SID list.

In addition, tree-SID SRv6 currently only supports basic SID and no other SRv6 SID formats.

Similar to unicast routing, the SRv6 tree-SID is composed of a locator and a function, where the function identifies the multicast state on the root, transit, and leaf router. The multicast state is presented by the replication segment via an incoming SID. The incoming SID does the following:

- behaves as an end.x, where the multicast payload is extracted and forwarded to the local host, in the case of bud and leaf routers
- identifies the instructions for a transit router, where the outgoing interface constructs the new outgoing SRv6 header

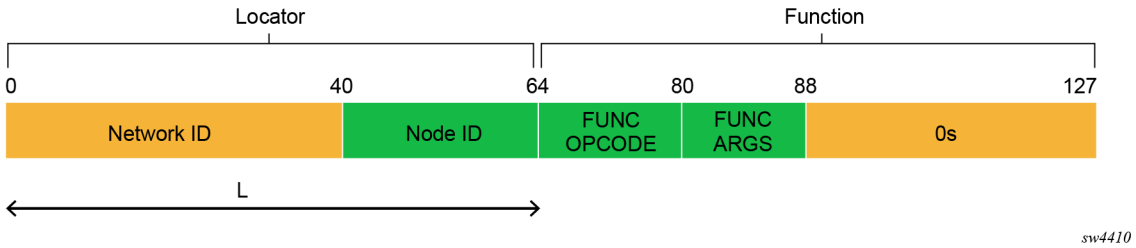
12.7.1 Tree-SID SRv6 header

Similarly to other SRv6 headers, an SRv6 tree-SID header uses a locator and a function to do the following:

- A locator from prefix /1 to /96 bits is used to forward the packet to the next replication segment.
- A function of length 16 bits or 20 bits is used to identify a replication segment and the multicast state.

The following figure shows an example of the SRv6 tree-SID header. The SRv6 tree-SID uses a different locator on the same node to deconflict the SID for multicast as compared to unicast. The function length is 16-bit, but SR OS supports up to 20 bits, using some of the ARGS bits to scale higher.

Figure 40: SRv6 tree-SID header



12.7.2 Tree-SID SRv6 implementation details

The SR OS SRv6 tree-SID is implemented as follows:

- supported only as selective PMSI and inclusive PMSI in NG-MVPN
- supports manual configuration using CLI only and does not support PCEP
- supports only BGP IPv4 neighboring
- supports only multicast source resolution via BGP-IPVN MPLS in a VPRN, not via BGP-IPVN segment-routing-6

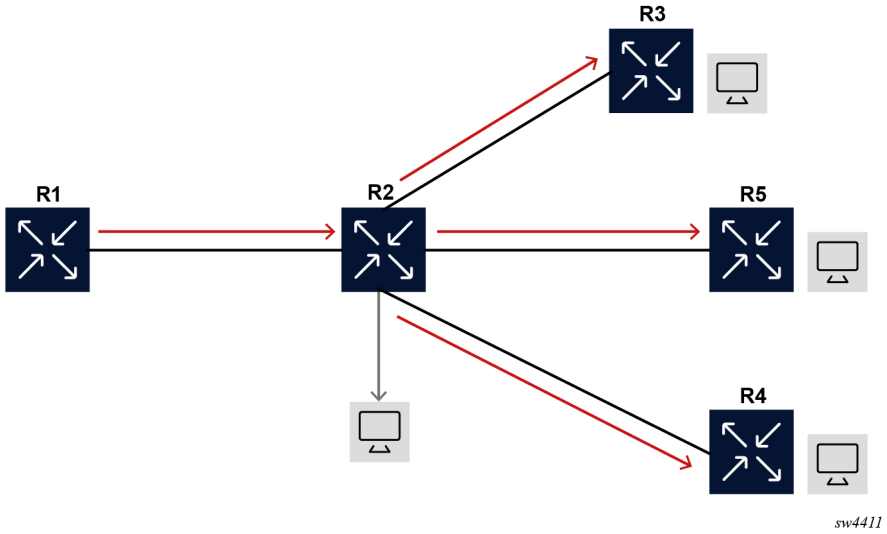
Note: BGP-IPVN segment-routing-6 cannot resolve the source IP currently for tree-SID SRv6.

- no support for SID-LIST, which means replication segments must be directly connected
- no support for tree-SID protection nexthop of SRv6 LFA, which means tree-SID recovery is based on IGP convergence if the primary outgoing interface is down
- no support for BFD
- UMH redundancy support only when UMH route selection is configured to use the unicast route preference (**configure service vprn mvpn umh-selection** command)
- for simpler debugging, recommend the function value be constant for the tree, to identify the tree end-to-end from the root to all the leaves
- distributes tree-SID SRv6 locator via IGP and is pingable

Implementation use case 1

The following figure shows replication segments between the Replication segment 1 (R1) node to the R2 node, and the R2 node to R3, R4, and R5 nodes.

Figure 41: Implementation use case 1



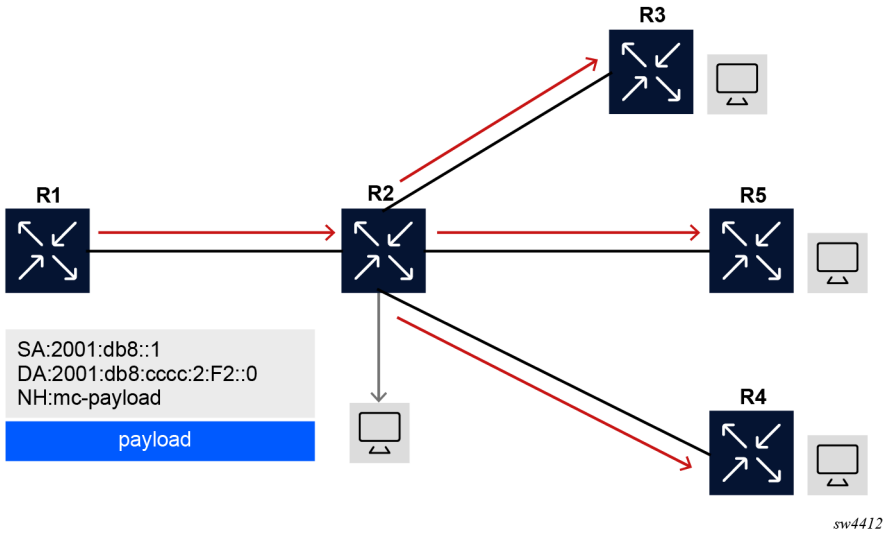
Implementation use case 2

- Node k has a classic IPv6 loopback address 2001:db8::k/128, which is advertised in the IGP.
- The address 2001:db8:cccc::/48 is dedicated to the internal SRv6 SID space.
- Node k has 2001:db8:cccc:k::/64 for its local SID space.
- Node k advertises 2001:db8:cccc:k::/64 in its IGP.
- Function :Fn:: is for End.Replication.

In the following figure, the R1 node encapsulates the multicast stream from the source into an SRv6 tree-SID, and builds the SRv6 SID with the R2 node as locator and F2 as the function (DA:2001:db8:cccc:2:F2::0). The R1 node forwards the packet toward the R2 node based on the locator.

Note: Nokia recommends the function to be the same for the entire tree from the root to transit to leaf routers. This helps identify the tree end-to-end and makes debugging and tree tracking easier.

Figure 42: Implementation use case 2

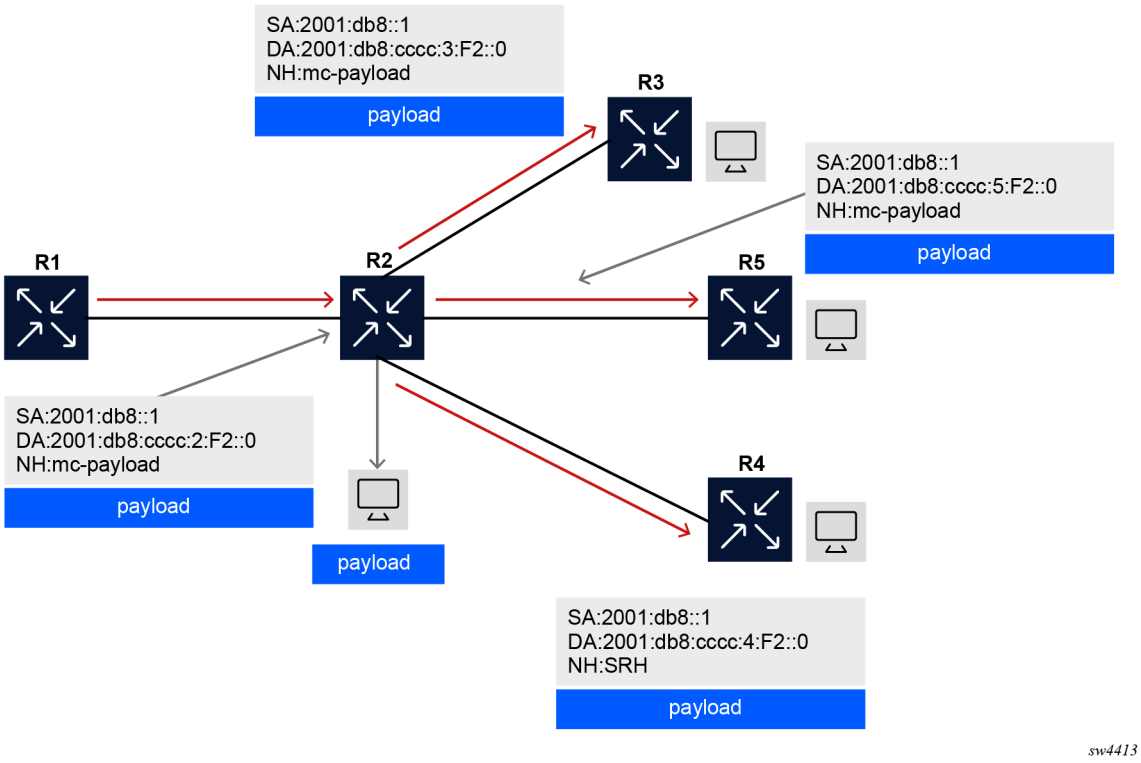


Implementation use case 3

In the following figure, the R2 node (replication SID: 2001:db8:cccc:2:F2::0, state: leaf) receives the packet and based on the locator, it uses the function value to find the multicast state for function F2. Based on the multicast state (replication segment), the R2 node forwards the packet to the R3 node (2001:db8:cccc:3:F2::0), the R4 node (2001:db8:cccc:4:F2::0), and the R5 node (2001:db8:cccc:5:F2::0). The R2 node also builds a new IPv6 destination using the corresponding locator of each router. The R2 node keeps the function as F2.

The R2 node also acts as end-x for the locally connected host. In this capacity it removes the SRv6 header and forwards the packet to the locally connected host.

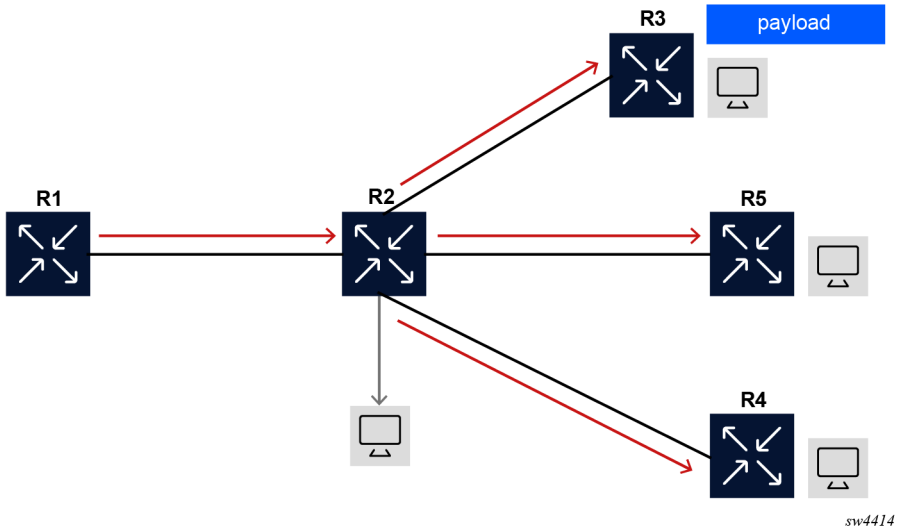
Figure 43: Implementation use case 3



Implementation use case 4

In the following figure, when the R3 node (Replication SID: 2001:db8:cccc:3:F2::0, state: leaf) receives the packet with its own locator and function F2, it realizes that it is the end.x for the locally connected host. The R3 node then removes the SRv6 header and forwards the payload to the locally connected host. The R4 node (Replication SID: 2001:db8:cccc:4:F2::0, state leaf) and the R5 node (Replication SID: 2001:db8:cccc:5:F2::0, state: leaf) follow the same procedure as the R3 node.

Figure 44: Implementation use case 4



12.7.3 Tree-SID SRv6 configuration guidelines

The following guidelines apply when configuring replication segments in the SR OS system:

- The system supports one replication segment with the same root IP address, tree ID and instance ID, of either type MPLS or SRv6 depending on its configuration. Use the commands in the following contexts to configure the replication-segment settings for SRv6 or MPLS.

```
configure router p2mp-sr-tree replication-segment segment-routing-v6
configure router p2mp-sr-tree replication-segment segment-routing-mpls
```

- P2MP policies can only associate with replication segments of the same type, either MPLS or SRv6. When an MPLS or SRv6 P2MP policy is assigned to the MVPN provider tunnel in the following contexts, the MVPN tunnel becomes the assigned type, either SRv6 or MPLS.

```
configure service vprn mvpn provider-tunnel inclusive p2mp-sr static-policy-srv6
configure service vprn mvpn provider-tunnel inclusive p2mp-sr static-policy-mpls
```

  – If there is a mismatch between the replication segment type and the PMSI type, the PMSI is not operational and the router does not advertise the AD routes in BGP.

  Note: The P2MP policy is accepted under the PMSI even if there is a mismatch; the system does not check for mismatch errors.

  – If there is no mismatch between RS and PMSI type, the router advertises the BGP route PTA based on this type.

The following example shows P2MP SR tree configuration with SRv6 and MPLS replication segments, P2MP policies, and MVPN provider tunnels.

Example: P2MP-SR-tree configuration (MD-CLI)

```
[ex:/configure router "Base" p2mp-sr-tree]
A:admin@node-2# info
    admin-state enable
    reserved-label-block "tree_sid_block"
    p2mp-policy "p2mp_policy_8193_10.20.1.3" {
        admin-state enable
        root-address 10.20.1.3
        tree-id 8193
        candidate-path "cp_1" {
            admin-state enable
            active-instance 1
            preference 0
            path-instances 1 {
                instance-id 1
            }
        }
    }
    p2mp-policy "p2mp_policy_8194_10.20.1.3" {
        admin-state enable
        root-address 10.20.1.3
        tree-id 8194
        candidate-path "cp_1" {
            admin-state enable
            active-instance 1
            preference 0
            path-instances 1 {
                instance-id 1
            }
        }
    }
    replication-segment "rs_8193_10.20.1.3_inst_1_mpls" {
        admin-state enable
        instance-id 1
        root-address 10.20.1.3
        tree-id 8193
        segment-routing-mpls {
            sid-action push
            downstream-nodes 1 {
                admin-state enable
                next-hop-address "10.180.3.2"
                label {
                    sid-list 1 {
                        replication-sid 78432
                    }
                }
            }
        }
    }
    replication-segment "rs_8194_10.20.1.3_ins_1_srv6_leaf" {
        admin-state enable
        instance-id 1
        root-address 10.20.1.1
        tree-id 8194
        segment-routing-v6 {
            role leaf
            incoming-sid {
                locator {
                    locator-name "p2mp"
                    function end-replicate
                    function-value 1001
                }
            }
```

```
        }
    }
    replication-segment "rs_8194_10.20.1.2_ins_1_srv6_bud" {
        admin-state enable
        instance-id 1
        root-address 10.20.1.2
        tree-id 8194
        segment-routing-v6 {
            role transit
            incoming-sid {
                locator {
                    locator-name "p2mp"
                    function end-replicate
                    function-value 1002
                }
            }
            downstream-nodes 1 {
                admin-state enable
                replication-sid 2001:db8:cccc:202::1002
            }
            downstream-nodes 2 {
                admin-state enable
                replication-sid 2001:db8:cccc:404::1002
            }
        }
    }
    replication-segment "rs_8194_10.20.1.2_ins_srv6_root" {
        admin-state enable
        instance-id 1
        root-address 10.20.1.3
        tree-id 8194
        segment-routing-v6 {
            role root
            downstream-nodes 1 {
                admin-state enable
                replication-sid 2001:db8:cccc:202::1003
            }
        }
    }
    replication-segment "rs_8193_10.20.1.3_ins_1_mpls" {
        admin-state enable
        instance-id 1
        root-address 10.20.1.3
        tree-id 8193
        segment-routing-mpls {
            sid-action push
            downstream-nodes 1 {
                admin-state enable
                next-hop-address "10.180.3.2"
                label {
                    sid-list 1 {
                        replication-sid 78432
                    }
                }
            }
        }
    }
[ex:/configure service]
A:admin@node-2# info
    vprn "1" {
        mvpn {
            provider-tunnel {
                inclusive {
                    p2mp-sr {
```

```
                        admin-state enable
                        static-policy-mpls "p2mp_policy_8192_10.20.1.3"
                    }
                }
                selective {
                    data-threshold {
                        group-prefix 224.0.0.0/4 {
                            threshold 1
                        }
                    }
                    p2mp-sr {
                        admin-state enable
                        static-policy-srv6 "mcastTreeSIDPolS"
                        maximum-p2mp-spmsi 4000
                    }
                }
            }
        }
    vprn "2" {
        mvpn {
            provider-tunnel {
                inclusive {
                    p2mp-sr {
                        admin-state enable
                        static-policy-srv6 "mcastTreeSIDPolI"
                    }
                }
                selective {
                    data-threshold {
                        group-prefix 224.0.0.0/4 {
                            threshold 1
                        }
                    }
                    p2mp-sr {
                        admin-state enable
                        static-policy-srv6 "mcastTreeSIDPolS"
                        maximum-p2mp-spmsi 4000
                    }
                }
            }
        }
```

Example: P2MP-SR-Tree configuration

```
A:node-2>config>router>p2mp-sr-tree# info
----------------------------------------------
            bfd-enable ipv4
            reserved-lbl-block "tree_sid_block"
            p2mp-policy "p2mp_policy_8193_10.20.1.3"
                root-address 10.20.1.3
                tree-id 8193
                candidate-path "cp_1"
                    preference 0
                    path-instances
                        index 1 instance-id 1
                    exit
                    active-instance 1
                    no shutdown
                exit
                no shutdown
            exit

            p2mp-policy "p2mp_policy_8194_10.20.1.3"
```

```
                root-address 10.20.1.3
                tree-id 8194
                candidate-path "cp_1"
                    preference 0
                    path-instances
                        index 1 instance-id 1
                    exit
                    active-instance 1
                    no shutdown
                exit
                no shutdown
            exit
            replication-segment "rs_8193_10.20.1.3_ins_1_mpls"
                root-address 10.20.1.3
                tree-id 8193
                instance-id 1
                segment-routing-mpls
                    sid-action push
                    downstream-nodes "1"
                        next-hop-address 10.180.3.2
                        replication-sid 78432
                        shutdown
                    exit
                exit
                shutdown
            exit
            replication-segment "rs_8194_10.20.1.3_ins_1_srv6_root"
                root-address 10.20.1.3
                tree-id 8194
                instance-id 1
                segment-routing-v6
                    role transit
                    downstream-nodes "1"
                        replication-sid 2001:db8:cccc:202::1003
                        shutdown
                    exit
                exit
                shutdown
            exit
            replication-segment "rs_8194_10.20.1.1_inst_1_srv6_leaf"
                root-address 10.20.1.1
                tree-id 8194
                instance-id 1
                segment-routing-v6
                    role leaf
                    incoming-sid
                        locator "locatorSRv6" function end-replicate function-value 3
                    exit
                exit
                no shutdown
            exit
            replication-segment "8194_10.20.1.2_inst_1_srv6_bud"
                root-address 10.20.1.2
                tree-id 8194
                instance-id 1
                segment-routing-v6
                    role leaf
                    incoming-sid
                        locator "p2mp" function end-replicate function-value 1002
                    exit
                    next-hop-id 1
                        replication-sid 2001:db8:cccc:202::1002
                        no shutdown
                    exit
```

```
                    next-hop-id 2
                        replication-sid 2001:db8:cccc:404::1002
                        no shutdown
                    exit
                exit
                no shutdown
            exit
            no shutdown
        exit
A:node-2>config>service# info
----------------------------------------------
        vprn 1 name "1" customer 1 create
            no shutdown
            mvpn
                provider-tunnel
                    inclusive
                        p2mp-sr
                            static-policy-mpls "p2mp_policy_8192_10.20.1.3"
                            no shutdown
                        exit
                    exit
                    selective
                        p2mp-sr
                            static-policy-mpls "mcastTreeSIDPolS"
                            no shutdown
                        exit
                        maximum-p2mp-spmsi 4000
                        data-threshold 224.0.0.0/4 1
                    exit
                exit
                vrf-target unicast
                exit
            exit
        vprn 2 name "2" customer 1 create
            no shutdown
            mvpn
                provider-tunnel
                    inclusive
                        p2mp-sr
                            static-policy-srv6 "mcastTreeSIDPolI"
                            no shutdown
                        exit
                    exit
                    selective
                        p2mp-sr
                            static-policy-srv6 "mcastTreeSIDPolS"
                            no shutdown
                        exit
                        maximum-p2mp-spmsi 4000
                        data-threshold 224.0.0.0/4 1
                    exit
                exit
            exit
        exot
----------------------------------------------
```

13 Troubleshooting tools

This chapter provides information about troubleshooting tools.

13.1 Mtrace

To help assess problems in the distribution of IP multicast traffic, the **mtrace** feature uses a traceroute feature implemented in multicast routers that is accessed via an extension to the IGMP protocol. The **mtrace** feature is used to print the path from the source to a receiver; it does this by passing a trace query hop-by-hop along the reverse path from the receiver to the source. At each hop, information such as the hop address, routing error conditions, and packet statistics should be gathered and returned to the requester.

Data added by each hop includes:

- query arrival time
- incoming interface
- outgoing interface
- previous hop router address
- input packet count
- output packet count
- total packets for this source/group
- routing protocol
- Time To Live (TTL) threshold
- forwarding/error code

The information enables the network administrator to determine:

- where multicast flows stop
- the flow of the multicast stream

When the trace response packet reaches the FHR (the router that is directly connected to the net of the source), that router sends the completed response to the response destination (receiver) address specified in the trace query.

If some multicast router along the path does not implement the multicast traceroute feature, or if there is some outage, no response is returned. To solve this problem, the trace query includes a maximum hop count field to limit the number of hops traced before the response is returned. This allows a partial path to be traced.

The reports inserted by each router contain not only the address of the hop, but also the TTL required to forward, and some flags to indicate routing errors, plus counts of the total number of packets on the incoming and outgoing interfaces, and those forwarded for the specified group. Taking differences in these counts for two traces separated in time, and comparing the output packet counts from one hop with the input packet counts of the next-hop, allows the calculation of packet rate and packet loss statistics for each hop, to isolate congestion problems.

13.1.1 Finding the last-hop router

The trace query must be sent to the multicast router which is the last hop on the path from the source to the receiver. If the receiver is on the local subnet (as determined using the subnet mask), then the default method is to multicast the trace query to all-routers.mcast.net (224.0.0.2) with a TTL of 1. Otherwise, the trace query is multicast to the group address because the last hop router is a member of that group if the receiver is. Therefore, it is necessary to specify a group that the intended receiver has joined. This multicast is sent with a default TTL of 64, which may not be sufficient for all cases.

When tracing from a multihomed host or router, the default receiver address may not be the wanted interface for the path from the source. In that case, the wanted interface should be specified explicitly as the receiver.

13.1.2 Directing the response

By default, **mtrace** first attempts to trace the full reverse path, unless the number of hops to trace is explicitly set with the hop option. If there is no response within a 3 second timeout interval, an asterisk (*) is printed, and the probing switches to hop-by-hop mode. Trace queries are issued starting with a maximum hop count of one and increasing by one until the full path is traced or no response is received. At each hop, multiple probes are sent. The first attempt is made with the unicast address of the host running **mtrace** as the destination for the response. As the unicast route may be blocked, the remainder of attempts request that the response be multicast to mtrace.mcast.net (224.0.1.32) with the TTL set to 32 more than what is needed to pass the thresholds seen so far along the path to the receiver. For the last attempts, the TTL is increased by another 32.

Alternatively, the TTL may be set explicitly with the TTL option.

For each attempt, if no response is received within the timeout, an asterisk (*) is printed. After the specified number of attempts have failed, **mtrace** tries to query the next-hop router with a DVMRP_ASK_NEIGHBORS2 request (as used by the **mrinfo** program) to determine the router type.

The output of **mtrace** is a short listing of the hops in the order they are queried, that is, in the reverse of the order from the source to the receiver. For each hop, a line is printed showing:

- hop number (counted negatively to indicate that this is the reverse path)
- multicast protocol
- threshold required to forward data (to the previous hop in the listing as indicated by the up-arrow character)
- cumulative delay for the query to reach that hop (valid only if the clocks are synchronized)

The response ends with a line showing the round-trip time, which measures the interval from when the query was issued until the response was received, both derived from the local system clock.

Mtrace packets use special IGMP packets with IGMP type codes of 0x1E and 0x1F.

13.2 Mstat

The **mstat** command adds the capability to show the multicast path in a limited graphic display and provide drops, duplicates, TTLs and delays at each node. This information is useful to the network operator

because it identifies nodes with high drop and duplicate counts. Duplicate counts are shown as negative drops.

The output of **mstat** provides a limited pictorial view of the path in the forward direction with data flow indicated by arrows pointing downward and the query path indicated by arrows pointing upward. For each hop, both the entry and exit addresses of the router are shown if different, along with the initial TTL required on the packet to be forwarded at this hop and the propagation delay across the hop assuming that the routers at both ends have synchronized clocks. The output consists of two columns, one for the overall multicast packet rate that does not contain lost/sent packets and a column for the (S,G)-specific case. The S,G statistics do not contain lost/sent packets.

13.3 Mrinfo

Mrinfo is a mechanism based on the **ask_neighbors**>**igmp** command to display the configuration information from the target multicast router. The type of information displayed includes the multicast capabilities of the router, code version, metrics, TTL-thresholds, protocols, and status. This information, for instance, can be used by network operators to verify whether bidirectional adjacencies exist. When the specified multicast router responds, the configuration is displayed.

14 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.

14.1 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*

14.2 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)*

14.3 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-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*

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 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*

RFC 9294, *Application-Specific Link Attributes Advertisement Using the Border Gateway Protocol - Link State (BGP LS)*

RFC 9351, *Border Gateway Protocol - Link State (BGP-LS) Extensions for Flexible Algorithm Advertisement*

RFC 9494, *Long-Lived Graceful Restart for BGP*

RFC 9552, *Distribution of Link-State and Traffic Engineering Information Using BGP*

14.4 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*

14.5 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)*

14.6 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*

14.7 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*

14.8 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*

14.9 Ethernet VPN (EVPN)

draft-ietf-bess-evpn-ip-aliasing-03, *EVPN Support for L3 Fast Convergence and Aliasing/Backup Path*

draft-ietf-bess-evpn-ipvpn-interworking-15, *EVPN Interworking with IPVPN*

draft-ietf-bess-evpn-l3mh-proto-00, *EVPN Multi-Homing support for L3 services*

draft-ietf-bess-evpn-unequal-lb-16, *Weighted Multi-Path Procedures for EVPN Multi-Homing* – section 9

draft-ietf-bess-evpn-vpws-gateway-01, *Ethernet VPN Virtual Private Wire Services Gateway Solution*

draft-rbickhart-evpn-ip-mac-proxy-adv-04, *Proxy MAC-IP Advertisement in EVPN*

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 9014, *Interconnect Solution for Ethernet VPN (EVPN) Overlay Networks*

RFC 9047, *Propagation of ARP/ND Flags in an Ethernet Virtual Private Network (EVPN)*

RFC 9135, *Integrated Routing and Bridging in Ethernet VPN (EVPN)*

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)*

RFC 9541, *Flush Mechanism for Customer MAC Addresses Based on Service Instance Identifier (I-SID) in Provider Backbone Bridging EVPN (PBB-EVPN)*

RFC 9625, *EVPN Optimized Inter-Subnet Multicast (OISM) Forwarding* – ingress replication and mLDP

RFC 9784, *Virtual Ethernet Segments for EVPN and Provider Backbone Bridge EVPN*

RFC 9785, *Preference-Based EVPN Designated Forwarder (DF) Election*

RFC 9819, *Argument Signaling for BGP Services in Segment Routing over IPv6 (SRv6)*

14.10 gRPC Remote Procedure Calls (gRPC)

cert.proto version 0.1.0, *gNOI Certificate Management Service*

file.proto version 0.1.0, *gNOI File Service*

gnmi.proto version 0.8.0, *gNMI Service Specification*

gnmi_ext.proto, *gNMI Commit Confirmed Extension*

gnmi_ext.proto, *gNMI Config Subscription Extension*

gnmi_ext.proto, *gNMI Depth Extension*

system.proto version 1.0.0, *gNOI System Service*

tunnel.proto version 0.2, *gRPC Tunnel Service*

PROTOCOL-HTTP2, *gRPC over HTTP2*

14.11 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*

RFC 9885, *Multi-Part TLVs in IS-IS*

14.12 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*

14.13 Internet Protocol (IP) general

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 5289, *TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)*

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*

RFC 8907, *The Terminal Access Controller Access-Control System Plus (TACACS+) Protocol*

14.14 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)*

RFC 9573, *MVPN/EVPN Tunnel Aggregation with Common Labels* – DCB and static service labels

14.15 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

14.16 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*

14.17 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)*

RFC 8784, *Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security*

14.18 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*

14.19 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"*

14.20 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*

14.21 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*

14.22 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*

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*

14.23 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 8071, *NETCONF Call Home and RESTCONF Call Home* – 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

14.24 Media sanitization

NIST Special Publication 800-88 Revision 1, *Guidelines for Media Sanitization* – Clear action for CF, MMC, SSD, SD, USB

14.25 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 4811, *OSPF Out-of-Band Link State Database (LSDB) Resynchronization* – OSPFv2

RFC 4812, *OSPF Restart Signaling* – OSPFv2

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*

14.26 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-multipath-18, *Path Computation Element Communication Protocol (PCEP) Extensions for Signaling Multipath Information*

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 8233, *Extensions to the Path Computation Element Communication Protocol (PCEP) to Compute Service-Aware Label Switched Paths (LSPs)* – Path Delay Metric

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*

RFC 9862, *Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing (SR) Policy Candidate Paths*

14.27 Point-to-Point Protocol (PPP)

RFC 1332, *The PPP Internet Protocol Control Protocol (IPCP)*

RFC 1661, *The Point-to-Point Protocol (PPP)*

RFC 1877, *PPP Internet Protocol Control Protocol Extensions for Name Server Addresses*

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*

14.28 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*

14.29 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*

14.30 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)*

14.31 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*

14.32 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*

14.33 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*

14.34 Segment Routing (SR)

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-bess-mvpn-evpn-sr-p2mp-07, *Multicast and Ethernet VPN with Segment Routing P2MP and Ingress Replication* – MVPN

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-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*

RFC 9514, *Border Gateway Protocol - Link State (BGP-LS) Extensions for Segment Routing over IPv6 (SRv6)*

RFC 9800, *Compressed SRv6 Segment List Encoding*

14.35 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 3430, *Simple Network Management Protocol (SNMP) over Transmission Control Protocol (TCP) Transport Mapping*

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 3434, *Remote Monitoring MIB Extensions for High Capacity Alarms*

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 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 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*

14.36 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*

14.37 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

RFC 9503, *Simple Two-Way Active Measurement Protocol (STAMP) Extensions for Segment Routing Networks* – excluding Sections 3, 4.1.2 and 4.1.3

RFC 9534, *Simple Two-Way Active Measurement Protocol Extensions for Performance Measurement on a Link Aggregation Group*

14.38 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)*

14.39 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*

14.40 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*

14.41 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 version 0.3.2, *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 2.12.2, *OpenConfig Interfaces Ethernet Model*

openconfig-if-ip.yang version 3.9.0, *OpenConfig Interfaces IP Model*

openconfig-if-ip-ext.yang version 2.3.1, *OpenConfig Interfaces IP Extensions Model*

openconfig-igmp.yang version 0.3.1, *OpenConfig IGMP Model*

openconfig-interfaces.yang version 3.8.0, *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 2.1.0, *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 1.1.0, *OpenConfig Packet Match Model*

openconfig-pim.yang version 0.4.3, *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-qos.yang version 0.11.2, *OpenConfig QoS Model*

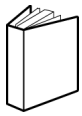
openconfig-qos-elements.yang version 0.11.2, *OpenConfig QoS Elements Model*

openconfig-qos-interfaces.yang version 0.11.2, *OpenConfig QoS Interfaces Model*

openconfig-qos-mem-mgmt.yang version 0.11.2, *OpenConfig QoS Memory Management 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 Device Model*
openconfig-vlan.yang version 3.2.2, *OpenConfig VLAN Model*

Customer document and product support



Customer documentation

Customer documentation welcome page

Technical support

Product support portal

Documentation feedback

Customer documentation feedback