7750 Service Router
7950 Extensible Routing System
Release 24.7.R1

Segment Routing and PCE User Guide

3HE 20114 AAAB TQZZA 01
Edition: 01
July 2024

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

Table of contents

1 Getting started

1.1 About this guide

This guide describes the Nokia SR OS segment routing and PCE functionality.

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.

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

The topics and commands described in this document apply to the following SR OS products:

- 7750 SR
- 7950 XRS

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

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

Note: The Segment Routing and PCE supports configuration using classic CLI and MD-CLI. This guide provides configuration examples based on classic CLI syntax only.
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 MD-CLI Command Reference Guide*
- *7450 ESS, 7750 SR, 7950 XRS, and VSR Clear, Monitor, Show, and Tools CLI Command Reference Guide (for both MD-CLI and Classic CLI)*

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

1.2 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.

2 Segment routing with MPLS data plane (SR-MPLS)

This section describes:

- Segment Routing (SR) in shortest path forwarding
- SR with Traffic Engineering (SR-TE)
- SR policies

2.1 Segment routing in shortest path forwarding

Segment routing provides support for shortest path routing and source routing using abstract segments for IS-IS and OSPF protocols. A segment can represent a local prefix of a node, a specific adjacency of the node (interface or next hop), a service context, or a specific explicit path over the network. For each segment, the IGP advertises a Segment ID (SID).

When segment routing is used together with the MPLS data plane, the SID is a standard MPLS label. A router forwarding a packet using segment routing pushes one or more MPLS labels.

Segment routing using MPLS labels can be used in both shortest path routing applications and in traffic engineering (TE) applications.

When a received IPv4 or IPv6 prefix SID is resolved, the Segment Routing module programs the Incoming Label Map (ILM) with a swap operation and programs the LTN with a push operation, both of which point to the primary or Loop-Free Alternate (LFA) Next-Hop Label to Forwarding Entry (NHLFE). An IPv4 or IPv6 tunnel to the prefix destination is also added to the TTM and can be used by shortcut applications and Layer 2 and Layer 3 services.

Segment routing provides the remote LFA feature, which expands the coverage of LFA by computing and automatically programming SR tunnels that are used as backup next hops. The SR shortcut tunnels terminate on a remote alternate node that provides loop-free forwarding for packets with resolved prefixes. When the **loopfree-alternates** option is enabled in an IS-IS or OSPF instance, SR tunnels are protected with an LFA backup next hop. If the prefix of a specific SR tunnel is not protected by the base LFA, the remote LFA automatically computes a backup next hop using an SR tunnel if the **remote-lfa** option is also enabled in the IGP instance.

2.1.1 Configuring segment routing in shortest path

Segment routing in an IGP routing instance is enabled using the sequence of commands described in this section.

First, the user configures the global label block, known as the Segment Routing Global Block (SRGB), which is reserved for assigning labels to segment routing prefix SIDs originated by this router. The label range is derived from the system dynamic label range and is not instantiated by default. The range is configured as follows.

```
config>router>mpls-labels>sr-labels start start-value end end-value
```

Next, the user enables the context to configure segment routing parameters within an IGP instance.

```
config>router>isis>segment-routing
config>router>ospf>segment-routing
```

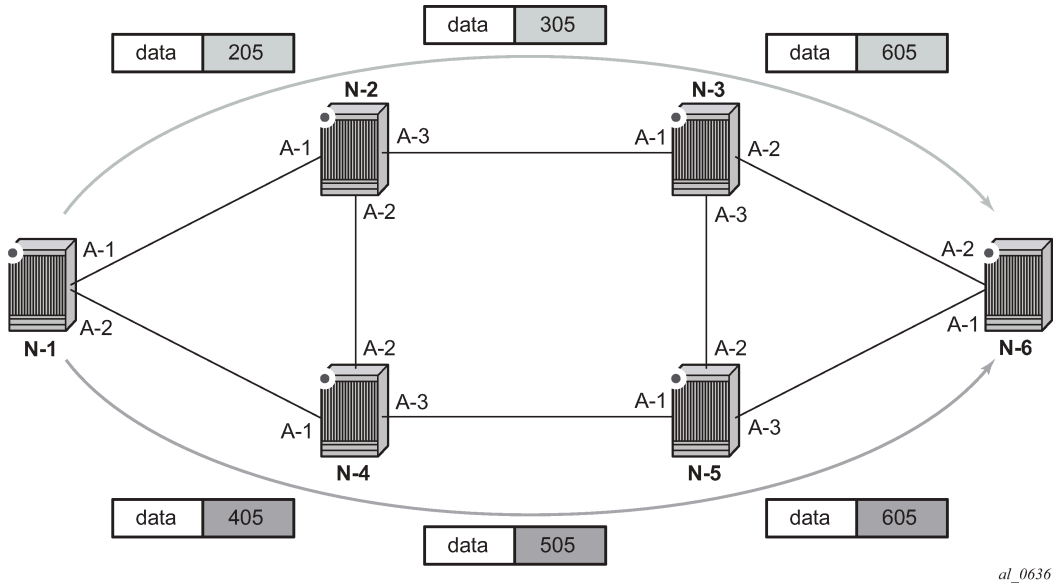
The key parameter is the configuration of the prefix SID index range and the offset label value that this IGP instance uses. Because each prefix SID represents a network global IP address, the SID index for a prefix must be unique network-wide. Thus, all routers in the network are expected to configure and advertise the same prefix SID index range for an IGP instance. However, the label value used by each router to represent this prefix, which is the label programmed in the ILM, can be local to that router by the use of an offset label, referred to as a start label. The relationship between the labels and SIDs is as follows:

Local label (prefix SID) = start label + {SID index}

The label operation in the network is similar to LDP when operating in the independent label distribution mode (RFC 5036), with the difference that the label value used to forward a packet to each downstream router is computed by the upstream router based on the advertised prefix SID index using the above formula.

The following figure shows an example of a router advertising its loopback address and the resulting packet label encapsulation throughout the network.

Figure 1: Packet label encapsulation using segment routing tunnel



Router N-6 advertises loopback 10.10.10.1/32 with a prefix index of 5. Routers N-1 to N-6 are configured with the same SID index range of [1,100] and an offset label of 100 to 600 respectively. The following are the actual label values programmed by each router for the prefix of PE2.

- N-6 has a start label value of 600 and programs an ILM with label 605.
- N-3 has a start label of 300 and swaps incoming label 305 to label 605.
- N-2 has a start label of 200 and swaps incoming label 205 to label 305.

Similar operations are performed by N-4 and N-5 for the bottom path.

N-1 has an SR tunnel to N-6 with two ECMP paths. It pushes label 205 when forwarding an IP or service packet to N-6 via downstream next-hop N-2 and pushes label 405 when forwarding via downstream next-hop N-4.

The CLI syntax for configuring the prefix SID index range and offset label value for an IGP instance are as follows:

```
config>router>isis>segment-routing>prefix-sid-range {global | start-label label-value max-
index index-value}
config>router>ospf>segment-routing>prefix-sid-range {global | start-label label-value max-
index index-value}
```

There are two mutually-exclusive modes of operation for the prefix SID range on the router. In the global mode of operation, the user configures the global value and this IGP instance takes the start label value as the lowest label value in the SRGB and the prefix SID index range size equal to the range size of the SRGB. After one IGP instance selects the **global** option for the prefix SID range, all IGP instances on the system are restricted to the same **global**.

The user must shut down segment routing context and delete the **prefix-sid-range** command in all IGP instances to change the SRGB. After the SRGB is changed, the user must re-enter the **prefix-sid-range** command. The SRGB range change fails if an already allocated SID index or label goes out of range.

In the per-instance mode of operation, the user partitions the SRGB into non-overlapping subranges among the IGP instances. The user configures a subset of the SRGB by specifying the start label value and the prefix SID index range size. All resulting net label values (start label + index) must be within the SRGB or the configuration fails. Furthermore, the code checks for overlaps of the resulting net label value range across IGP instances and strictly enforces values that do not overlap between ranges.

The user must shut down the segment routing context of an IGP instance to change the SID index or label range of that IGP instance using the **prefix-sid-range** command. Any range change fails if an already allocated SID index or label goes out of range.

The user can change the SRGB at any time as long as it does not reduce the current per-IGP instance SID index or label range defined with the **prefix-sid-range**. Otherwise, the user must shut down the segment routing context of the IGP instance, then delete and reconfigure the **prefix-sid-range** command.

Finally, the user brings up segment routing on that IGP instance by shutting down the context:

```
config>router>isis>segment-routing>no shutdown
config>router>ospf>segment-routing>no shutdown
```

This command fails if the user has not previously enabled the **router-capability** option in the IGP instance. Segment routing must be advertised to all routers in a domain so that routers that support the capability only program the node SID in the datapath toward neighbors that also support it.

```
config>router>isis>advertise-router-capability {area | as}
config>router>ospf>advertise-router-capability {link | area | as}
```

The IGP segment routing extensions are area-scoped. The user must configure the flooding scope as **area** in OSPF and as **area** or **as** in IS-IS.

Next, the user uses one of the following commands to assign a node SID index or label to the prefix, representing the primary address of a network interface of type **system** or **loopback**. A separate SID value can be configured for each IPv4 and IPv6 primary address of the interface.

```
config>router>isis>interface>ipv4-node-sid index value
config>router>ospf>area>interface>node-sid index value
```

```
config>router>ospf3>area>interface>node-sid index value
config>router>isis>interface>ipv4-node-sid label value
config>router>ospf>area>interface>node-sid label value
config>router>ospf3>area>interface>node-sid label value
config>router>isis>interface>ipv6-node-sid index value
config>router>isis>interface>ipv6-node-sid label value
```

The secondary address of an IPv4 interface cannot be assigned a node SID index and does not inherit the SID of the primary IPv4 address. The same applies to the non-primary IPv6 addresses of an interface.

In IS-IS, an interface inherits the configured IPv4 or IPv6 node SID value in any level the interface participates in (Level 1, Level 2, or both).

In OSPFv2 and OSPFv3, the node SID is configured in the primary area but is inherited in any other area in which the interface is added as secondary.

The preceding commands fail if the network interface is not of type **system** or **loopback**, or if the interface is defined in an IES or a VPRN context. Assigning the same SID index or label value to the same interface in two different IGP instances is not allowed within the same node.

For OSPF, the protocol version number and the instance number dictate if the node-SID index or label is for an IPv4 or IPv6 address of the interface. Specifically, the support of address families in OSPF is as follows:

- for ospfv2, always IPv4 only
- for ospfv3, instance 0..31, ipv6 only
- for ospfv3, instance 64..95, ipv4 only

The value of the label or index SID is taken from the range configured for this IGP instance. When using the global mode of operation, a new segment routing module checks that the same index or label value is not assigned to more than one loopback interface address. When using the per-instance mode of operation, this check is not required because the index and the label ranges of the various IGP instances are not allowed to overlap.

For an individual adjacency, values for the label may be provisioned for an IS-IS or OSPF interface. If they are not provisioned, they are dynamically allocated by the system from the dynamic label range. The following CLI commands are used:

```
config>router>isis>interface
   [no] ipv4-adjacency-sid label value
   [no] ipv6-adjacency-sid label value

config>router>ospf>area>interface
   [no] adjacency-sid label value
```

The *value* must correspond to a label in a reserved label block in provisioned mode referred to by the **srlb** command (see Segment routing local block for more details of SRLBs).

A static label *value* for an adjacency SID is persistent. Therefore, the P-bit of the Flags field in the Adjacency-SID TLV advertised in the IGP is set to 1.

By default, a dynamic adjacency SID is advertised for an interface. However, if a static adjacency SID value is configured, then the dynamic adjacency SID is deleted and only the static adjacency SID used. Changing an adjacency SID from dynamic (for example, **no adjacency-sid**) to static, or the other way around, may result in traffic being dropped as the ILM is reprogrammed.

For a provisioned adjacency SID of an interface, a backup is calculated similar to a regular adjacency SID when **sid-protection** is enabled for that interface.

Provisioned adjacency SIDs are only supported on point-to-point interfaces.

2.1.2 Configuring single shared loopback SR SID

When configuring an IPv4 or IPv6 SR SID for OSPF or IS-IS instances, the single shared SID for loopback or system interfaces can be enabled by the routing protocol independent **sr-mpls**>**prefix-sids** command. One or more IGP protocol instances can have a unique **sr-mpls**>**prefix-sids** configured and share this interface SID for an interface. This enhancement relaxes the otherwise imposed SID uniqueness for a loopback or system interface across all configured routing instances on a device.

It is possible to configure the **sr-mpls**>**prefix-sids** by label or index. The global **prefix-sid-range** must be configured in the routing instance when the **sr-mpls**>**prefix-sids** command is used.

- **configure router isis segment-routing prefix-sid-range global**
- **configure router ospf segment-routing prefix-sid-range global**
- **configure router ospf3 segment-routing prefix-sid-range global**

When a shared SID is configured outside the routing instances, it can be used for all instances when the routing protocol is enabled on the interface. The following CLI configures the prefix SIDs.

```
configure
|
+---router
|   +---segment-routing
|   |   +---sr-mpls
|   |   |   +---prefix-sids [<ip-int-name>]
|   |   |   |   no prefix-sids [<ip-int-name>]
|   |   |   |   +---no ipv4-sid
|   |   |   |   |   ipv4-sid index <[0..4294967295]>
|   |   |   |   |   ipv4-sid label <[32..1048575]>
|   |   |   |   +---no ipv6-sid
|   |   |   |   |   ipv6-sid index <[0..4294967295]>
|   |   |   |   |   ipv6-sid label <[32..1048575]>
|   |   |   |   |
|   |   |   |   |---node-sid
|   |   |   |   |   no node-sid
```

The following commands are used for configuration:

- **ipv4-sid**

  This command is used to configure the SID associated with the primary IPv4 address of the loopback or system interface.

- **ipv6-sid**

  This command is used to configure the SID associated with the primary IPv6 address of the loopback or system interface.

- **node-sid**

  This command sets the N-flag. The N-flag is set when the prefix SID is a node SID, as described in RFC 8402. If the N-flag is not set, the address is an SR anycast SID.

The following considerations apply for shared **sr-mpls**>**prefix-sids**:

- When an **sr-mpls**>**prefix-sids** is shared between IGP instances, all instances must share the same SR label range. This means that the instances must use the "global" SRGB range.

- Locally configured shared **sr-mpls**>**prefix-sids** share the statistics on that node, if configured. As a result, when incoming SID statistics on both OSPF and IS-IS are enabled and the SID is shared, the same statistics are displayed for both IGPs.

The following restrictions apply when configuring the **sr-mpls**>**prefix-sids**:

- The **sr-mpls**>**prefix-sids** command can only be used for loopback and system interfaces.
- Exporting **sr-mpls**>**prefix-sids** into BGP and using it for stitching an SR IGP domain with BGP-based SR MPLS tunnels is not supported.
- On the same interface, sharing the node SID across different address families is not allowed (for example, IPv4 node SID in ISIS and IPv6 in OSPFv3 or even IPv4 and IPv6 in the same ISIS instance).
- Configuring a SID as a prefix SID in one instance and as node SID in another instance is not allowed. For example, if IS-IS has assigned an IPv4 node SID or IPv6 node SID to a loopback in an IS-IS instance, OSPF cannot install the same SID on that loopback as a shared **sr-mpls**>**prefix-sids**.
- Each **sr-mpls**/**prefix-sids** SID must be unique across all routing instances.
- A regular IGP node SID and SR-MPLS prefix SID can be configured on a single interface for a single IGP algorithm. In this case, the IGP overrides the **configure**>**router**>**segment-routing**>**sr-mpls**>**prefix-sids** configuration, and only the IGP node SID is advertised.

Use the **show router segment-routing sr-mpls prefix-sids** and **tools dump router segment-routing tunnel** CLI commands to verify the operation of the shared SIDs. For more information, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Clear, Monitor, Show, and Tools CLI Command Reference Guide*.

Example: Show command output

```
*A:Dut-A# show router segment-routing sr-mpls prefix-sids

===============================================================================
Rtr Base SR-MPLS Prefix-SIDs
===============================================================================
Interface Name                  AF     SID       Label     State
-------------------------------------------------------------------------------
System                          IPv4   123       100123    enabled
System                          IPv6   234       100234    ifFailed
loopback.0                      IPv4   345       100345    ifDown
loopback.0                      IPv6   456       100456    ifDown
loopback.4                      IPv4   567       100567    failed
loopback.4                      IPv6   -         -         adminDown
loopback.6                      IPv4   -         -         adminDown
loopback.6                      IPv6   678       100678    notPref
-------------------------------------------------------------------------------
No. of Prefix-SIDs: 4
===============================================================================
```

```
*A:Dut-C# tools dump router segment-routing tunnel
===========================================================================================================
Legend: (B) - Backup Next-hop for Fast Re-Route
        (D) - Duplicate
label stack is ordered from top-most to bottom-most
===========================================================================================================
-----------------------------------------------------------------------------------------------------+
 Prefix                                                                                              |
 Sid-Type        Fwd-Type       In-Label  Prot-Inst(algoId)                                           |
                 Next Hop(s)                                          Out-Label(s) Interface/Tunnel-ID |
-----------------------------------------------------------------------------------------------------+
 1.1.1.3
 Node            Terminating    20003     IGP-Shared
```

```
 1.1.1.5
 Node            Orig/Transit   20005     ISIS-0
                 10.10.10.2                                    20005       To_1/1/1(E)
 10.10.10.2
 Adjacency       Transit        524287    ISIS-0
                 10.10.10.2                                    3           To_1/1/1(E)
-------------------------------------------------------------------------------------------+
No. of Entries: 3
-------------------------------------------------------------------------------------------+
*A:Dut-C#
```

2.1.3 Segment routing shortest path forwarding with IS-IS

This section describes the segment routing shortest path forwarding with IS-IS.

2.1.3.1 IS-IS control protocol changes

The following TLVs and sub-TLVs are defined in *draft-ietf-isis-segment-routing-extensions* and are supported in the implementation of segment routing in IS-IS:

- Prefix Segment Identifier (Prefix-SID) sub-TLV
- Adjacency Segment Identifier (Adj-SID) sub-TLV
- SID/Label Binding TLV
- SR-Capabilities sub-TLV
- SR-Algorithm sub-TLV

This section describes the behaviors of the IS-IS support of the segment routing TLVs and sub-TLVs.

SR OS supports advertising the IS-IS Router Capability TLV (RFC 4971) for topology MT0 and MT2.

Special attention is taken when leaking the IS-IS router capability when both MT0 and MT2 are enabled. When leaking router capability between IS-IS levels, as defined in RFC 7981, a reachability check must be performed. The router performs the following reachability check with MT-IS-IS MT2 enabled:

- leak the router capability TLVs with a valid IPv4 router ID via IPv4 MT0
- for router capabilities with no valid IPv4 router ID but a valid IPv6 router ID, perform a reachability check via MT0 and MT2
- when either an IS-IS IPv4 or IPv6 router ID is reachable, then redistribute router capability is redistributed

If Prefix-SID sub-TLVs for the same prefix are received in different MT numbers of the same IS-IS instance, a tiebreaking mechanism is applied to resolve the Prefix-SID. The IS-IS MT0 and MT2 tiebreaking mechanism sorts a specific prefix and gives precedence as follows:

1. smaller route preference sorts ahead
2. smaller route metric sorts ahead
3. MT0 sorts ahead of MT2
4. if all are equal, the final step is for the smaller IS-IS instance ID to sort ahead

When a duplicate Prefix-SID exists between two different prefixes, an error is logged and a trap is generated, as described in Error and resource exhaustion handling.

The I and V flags are both set to 1 when originating the SR-Capabilities sub-TLV to indicate support for processing both SR MPLS-encapsulated IPv4 and IPv6 packets on the network interfaces of the router. These flags are not checked when the sub-TLV is received. Only the SRGB range is processed.

The algorithm field is set to 0, meaning the Shortest Path First (SPF) algorithm based on the link metric, when originating the SR-Algorithm Capability sub-TLV but is not checked when the sub-TLV is received.

SR OS originates a single Prefix-SID sub-TLV per the IS-IS IP-reachability TLV and processes the first Prefix-SID sub-TLV only if multiple sub-TLVs are received within the same IS-IS IP-reachability TLV.

SR OS encodes the 32-bit index in the Prefix-SID sub-TLV. The 24-bit label is not supported.

Prefix-SID sub-TLV encoding

SR OS originates a Prefix-SID sub-TLV with the following encoding of flags and the following processing rules:

- The R-flag is set if the Prefix-SID sub-TLV, along with its corresponding IP-reachability TLV, is propagated between the levels.
- The N-flag is always set because SR OS supports a Prefix-SID type that is node SID only.
- The P-flag (no-PHP flag) is always set, meaning the label for the Prefix-SID is pushed by the PHP router when forwarding to this router. The SR OS PHP router processes a received Prefix-SID with the P-flag set to 0 and uses implicit-null for the outgoing label toward the router that advertised it, as long as the P-flag is also set to 1.
- The E-flag (Explicit-Null flag) is always set to 0. An SR OS PHP router, however, processes a received Prefix-SID with the E-flag set to 1. When the P-flag is also set to 1, it pushes explicit-null for the outgoing label toward the router that advertised it.
- The V-flag is always set to 0 to indicate an index value for the SID.
- The L-flag is always set to 0 to indicate that the SID index value is not locally significant.
- The algorithm field is always set to 0 to indicate that the SPF algorithm based on the link metric is used and is not checked when the Prefix-SID sub-TLV is received.
- SR OS resolves a Prefix-SID sub-TLV received without the N-flag set but with the prefix length equal to 32. A trap, however, is raised by IS-IS.
- SR OS does not resolve a Prefix-SID sub-TLV received with the N-flag set and a prefix length other than 32. A trap is raised by IS-IS.
- SR OS resolves a Prefix-SID received within an IP-reachability TLV based on the following route preference:

  1. a SID received via Layer 1 in a Prefix-SID sub-TLV part of the IP-reachability TLV

  2. a SID received via Layer 2 in a Prefix-SID sub-TLV part of the IP-reachability TLV
- A prefix received in an IP-reachability TLV is propagated, along with the Prefix-SID sub-TLV, by default from Layer 1 to Layer 2 by an Layer 1/Layer 2 (L1/L2) router. A router in Layer 2 sets up an SR tunnel to the Layer 1 router via the L1/L2 router, which acts as a Label Switching Router (LSR).
- A prefix received in an IP-reachability TLV is not propagated, along with the Prefix-SID sub-TLV, by default from Level 2 to Level 1 by an L1/L2 router. If the user adds a policy to propagate the received prefix, a router in Layer 1 sets up an SR tunnel to the Layer 2 router via the L1/L2 router, which acts as an LSR.

- If a prefix is summarized by an Area Border Router (ABR), the Prefix-SID sub-TLV is not propagated with the summarized route between levels. To propagate the node SID for a /32 prefix, route summarization must be disabled.
- SR OS propagates the Prefix-SID sub-TLV when exporting the prefix to another IS-IS instance; however, it does not propagate if the prefix is exported from a different protocol. When the corresponding prefix is redistributed from another protocol such as OSPF, the prefix SID is removed.

Adj-SID sub-TLV encoding

SR OS originates an Adj-SID sub-TLV with the following encoding of the flags:

- The F-flag is set to 0 to indicate the IPv4 family and is set to 1 for IPv6 family for the adjacency encapsulation.
- The B-flag is set to 0 and is not processed on receipt.
- The V-flag is always set to 1.
- The L-flag is always set to 1.
- The S-flag is set to 0 because assigning an Adj-SID to parallel links between neighbors is not supported. A received Adj-SID with S-flag set is not processed.
- The weight octet is not supported and is set to all zeros.

SID/Label Binding TLV rules and limitations

SR OS can originate the SID/Label Binding TLV as part of the Mapping Server feature (see Segment routing mapping server function for IPv4 prefixes for more information) for IS-IS MT0 only. Consider the following rules and limitations:

- Only the mapping server Prefix-SID sub-TLV within the TLV is processed and the ILMs installed if the prefixes in the provided range are resolved.
- The range and FEC prefix fields are processed. Each FEC prefix is resolved similar to the Prefix-SID sub-TLV, meaning there must be an IP-reachability TLV received for the exact matching prefix.
- If the same prefix is advertised with both a Prefix-SID sub-TLV and a mapping server Prefix-SID sub-TLV. The resolution follows the following route preference:
  1. SID received via Level 1 in a Prefix-SID sub-TLV part of IP-reachability TLV
  2. SID received via Level 2 in a Prefix-SID sub-TLV part of IP-reachability TLV
  3. SID received via Level 1 in a mapping server Prefix-SID sub-TLV
  4. SID received via Level 2 in a mapping server Prefix-SID sub-TLV
- The entire TLV can be propagated between levels based on the settings of the S-flag. The TLV cannot be propagated between IS-IS instances (see Segment routing mapping server function for IPv4 prefixes for more information). Finally, a Level 1 or Level 2 router does not propagate the Prefix-SID sub-TLV from the SID/Label Binding TLV (received from a mapping server) into the IP-reachability TLV if the latter is propagated between levels.
- The mapping server that advertised the SID/Label Binding TLV does not need to be in the shortest path for the FEC prefix.
- If the same FEC prefix is advertised in multiple binding TLVs by different routers, the SID in the binding TLV of the first router that is reachable is used. If that router becomes unreachable, the next reachable one is used.

- No check is performed if the content of the binding TLVs from different mapping servers are consistent or not.
- Any other sub-TLV, for example, the SID/Label sub-TLV, ERO metric and unnumbered interface ID ERO, is ignored but the user can view the octets of the received-but-not-supported sub-TLVs using the IGP **show** command.

2.1.3.2 Segment routing multitopology considerations

Segment routing with IS-IS is supported with Multitopology IS-IS MT0 (standard IPv4 and IPv6 topology) and MT2 (IPv6-only topology). Some segment routing functionality is restricted to MT0 only. The following table lists the functionality constraints that apply when MT2 is used.

Table 1: MT2 functionality restrictions

Functionality	IS-IS MT2 support
Mix of MT0 and MT2	Yes
MT0-only: IPv6 protected Adj-SID follow MT0	Yes
MT2-only: IPv6 protected Adj-SID follow MT2	Yes
MT0/MT2: IPv6 protected Adj-SID follow MT2	Yes
IPv6 mapping server	No
uLoop avoidance	No
Legacy Traffic Engineering (TE) attributes for local links	No
SR-policy Application Specific Link Attributes (ASLA)	Yes
IS-IS Flexible Algorithm Exclude Shared Risk Link Group (SRLG) sub-TLV	No
Loop-Free Alternate (LFA), Remote Loop-Free Alternate (RLFA), and Topology-Independent Loop-Free Alternate (TI-LFA)	Yes
MT2 IS-IS Prefix-SIDs	Yes
BGP-LS for SR-ISIS MT2	Yes
Traffic Engineering (TE)	No
IGP shortcut	No

When a user configures the following command, Prefix-SIDs and Adj-SIDs are advertised for MT-ISIS MT2.

```
configure router isis segment-routing multi-topology mt2
```

The encoding the SR-MPLS Prefix-SIDs and Adj-SIDs depends on the combined usage of the following commands.

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

IPv6 advertised in MT0 only

The following logic applies for IPv6 advertised in MT0 only using the following commands:

- **MD-CLI**

  ```
  configure router isis segment-routing multi-topology mt2 false
  ```

- **classic CLI**

  ```
  configure router isis no segment-routing multi-topology
  ```

- All protected and unprotected IPv4 and IPv6 Adj-SIDs are advertised in the Traffic Engineering Neighbor (TE-NBR) TLVs.
- For the protected Adj-SIDs a backup is programmed following the MT0 topology, which is the same for both IPv4 and IPv6.

IPv6 advertised in MT2 only

The following logic applies for IPv6 advertised in MT2 only (using the **ipv6-routing mt** and **multi-topology ipv6-unicast** commands):

- IPv4 protected and unprotected Adj-SIDs are advertised in the TE-NBR TLVs.
- IPv6 protected and unprotected Adj-SIDs are advertised in the Multitopology Neighbor (MT-NBR) TLVs.
- For protected Adj-SIDs:
  - For IPv4 protected Adj-SIDs, a backup is programmed following the MT0 topology
  - For IPv6 protected Adj-SIDs, a backup is programmed following the MT2 topology

IPv6 advertised in MT0 and MT2 simultaneously

The following logic applies for IPv6 advertised in MT0 and MT2 simultaneously (using the **ipv6-routing native** and **multi-topology ipv6-unicast** commands):

- IPv4 protected and unprotected Adj-SIDs are advertised in the TE-NBR TLVs.
- IPv6 protected and unprotected Adj-SIDs are advertised in the MT-NBR TLVs.
- IPv6 unprotected Adj-SIDs are also advertised in the TE-NBR TLVs (The advertised unprotected Adj-SID is identical as the IPv6 Adj-SID advertised in MT-NBR TLV of the MT2).
- Programming of backup paths for protected Adj-SIDs:
  - IPv4 Adj-SIDs – a backup is programmed following the MT0 topology
  - IPv6 Adj-SIDs – a backup is programmed following the MT2 topology

Note: No protected IPv6 Adj-SID exists in MT0. Only an unprotected IPv6 Adj-SID exists in MT0.

The **multi-topology mt2** command operates as follows for IPv6 routes:

- **Multitopology MT2 segment routing disabled**
  By default, Segment Routing in multitopology IS-IS encoding is disabled. Only SR-MPLS tunnels that are IPv6 native SR-MPLS routes (for IPv6 SIDs in MT0) are programmed.
- **Multitopology MT2 segment routing enabled**
  When MT2 segment routing is enabled, the following applies:
  - The **multi-topology mt2** command instructs the IS-IS router to program SR-MPLS tunnels for multi-topology IPv6 routes in MT2.
  - A router configured as IPv6 MT2 only (using the **ipv6-routing mt** and **multi-topology ipv6-unicast** commands), contains only SR-MPLS MT2 topology tunnels that are programmed for IPv6. Only IPv4 tunnels will be programmed for SR-MPLS MT0 topologies.
  - A router configured for both MT0 and IPv6 MT2 (using the **ipv6-routing native** and **multi-topology ipv6-unicast** commands), has IPv6 and IPv4 SR-MPLS tunnels programmed for MT0 and MT2 routes.

The use of IGP shortcuts (RSVP or SR-TE) in MT2 is not supported.

BGP-LS enables the export of Multitopology IS-IS MT2 and MT0 prefixes and Segment Routing SIDs. When TE attributes as defined in RFC 5305 and RFC 8750 are received on a router from remote devices within the MT2 topology, these attributes are seamlessly integrated into the Traffic Engineering Database (TEDB). They are then conveyed through BGP-LS NRLI encoding for dissemination.

IS-IS Link State Packets (LSP) encoding examples

For SR OS, a user can enable Segment Routing MPLS within IS-IS MT0 alone, MT2 alone, or simultaneously in both MT0 and MT2. This choice has a notable effect on how Prefix-SIDs are presented in IS-IS LSPs. The following sections display encoding examples for each of these three scenarios.

IPv6 in MT0

The following applies for IPv6 in MT0.

Traffic Engineering Neighbor (TE-NBR) TLVs advertise all protected and unprotected IPv4 and IPv6 Adj-SIDs.

Example: Encoding for IPv6 in MT0

```
TE IS Nbrs   :
    Nbr   : Dut-A.00
    Default Metric  : 10
    Sub TLV Len     : 153
    IF Addr   : 1.1.3.3
    IPv6 Addr : 3ffe::101:303
    Nbr IP    : 1.1.3.1
    Nbr IPv6  : 3ffe::101:301
    MaxLink BW: 10000000 kbps
    Resvble BW: 10000000 kbps
    Unresvd BW:
        BW[0] : 10000000 kbps
        BW[1] : 10000000 kbps
        BW[2] : 10000000 kbps
        BW[3] : 10000000 kbps
        BW[4] : 10000000 kbps
        BW[5] : 10000000 kbps
        BW[6] : 10000000 kbps
        BW[7] : 10000000 kbps
    Admin Grp : 0x0
    TE Metric : 1000
```

```
    TE APP LINK ATTR    :
      SABML-flag:Non-Legacy SABM-flags:   X
        Delay Min : 1000000 Max : 1000000
        TE Metric : 1000
    Adj-SID: Flags:v4BVL Weight:0 Label:524287
    Adj-SID: Flags:v4VL Weight:0 Label:524285
    Adj-SID: Flags:v6BVL Weight:0 Label:524286
    Adj-SID: Flags:v6VL Weight:0 Label:524284
```

IPv6 in MT2

The following applies for IPv6 in MT2:

- TE-NBR TLVs advertise IPv4 protected and unprotected Adj-SIDs.
- Multitopology Neighbor (MT-NBR) TLVs advertise IPv6 protected and unprotected Adj-SIDs.

Example: Encoding for IPv6 in MT2

```
TE IS Nbrs   :
    Nbr   : Dut-A.00
    Default Metric  : 10
    Sub TLV Len     : 103
    IF Addr   : 1.1.3.3
    Nbr IP    : 1.1.3.1
    MaxLink BW: 10000000 kbps
    Resvble BW: 10000000 kbps
    Unresvd BW:
        BW[0] : 10000000 kbps
        BW[1] : 10000000 kbps
        BW[2] : 10000000 kbps
        BW[3] : 10000000 kbps
        BW[4] : 10000000 kbps
        BW[5] : 10000000 kbps
        BW[6] : 10000000 kbps
        BW[7] : 10000000 kbps
    Admin Grp : 0x0
    TE Metric : 1000
    TE APP LINK ATTR    :
      SABML-flag:Non-Legacy SABM-flags:   X
        Delay Min : 1000000 Max : 1000000
        TE Metric : 1000
    Adj-SID: Flags:v4BVL Weight:0 Label:524287
    Adj-SID: Flags:v4VL Weight:0 Label:524281
  MT IS Nbrs     :
    MT ID           : 2
    Nbr   : Dut-A.00
    Default Metric  : 10
    Sub TLV Len     : 220
    IPv6 Addr : 3ffe::101:303
    TE APP LINK ATTR    :
      SABML-flag:Non-Legacy SABM-flags:   X
        Delay Min : 1000000 Max : 1000000
        TE Metric : 1000
    Adj-SID: Flags:v6BVL Weight:0 Label:524286
    Adj-SID: Flags:v6VL Weight:0 Label:524280
    End.X-SID: 300::2000 flags:BP algo:0 weight:0 endpoint:End.X-PSP
    End.X-SID: 310::2000 flags:BP algo:128 weight:0 endpoint:End.X-PSP
    End.X-SID: 320::2000 flags:BP algo:129 weight:0 endpoint:End.X-PSP
    End.X-SID: 330::2000 flags:BP algo:130 weight:0 endpoint:End.X-PSP
    End.X-SID: 340::2000 flags:BP algo:131 weight:0 endpoint:End.X-PSP
    End.X-SID: 350::2000 flags:BP algo:132 weight:0 endpoint:End.X-PSP
    End.X-SID: 360::2000 flags:BP algo:133 weight:0 endpoint:End.X-PSP
```

IPv6 advertised in both MT0 and MT2

The following applies for IPv6 in both MT0 and MT2:

- TE-NBR TLVs advertise IPv4 protected and unprotected Adj-SIDs.
- MT-NBR TLVs advertise IPv6 protected and unprotected Adj-SIDs.
- IPv6 unprotected Adj-SIDs are also advertised in TE-NBR TLVs (in the same Adj-SID).

Example: Encoding for IPv6 advertised in both MT0 and MT2

```
TE IS Nbrs   :
    Nbr   : Dut-A.00
    Default Metric  : 10
    Sub TLV Len     : 146
    IF Addr   : 1.1.3.3
    IPv6 Addr : 3ffe::101:303
    Nbr IP    : 1.1.3.1
    Nbr IPv6  : 3ffe::101:301
    MaxLink BW: 10000000 kbps
    Resvble BW: 10000000 kbps
    Unresvd BW:
        BW[0] : 10000000 kbps
        BW[1] : 10000000 kbps
        BW[2] : 10000000 kbps
        BW[3] : 10000000 kbps
        BW[4] : 10000000 kbps
        BW[5] : 10000000 kbps
        BW[6] : 10000000 kbps
        BW[7] : 10000000 kbps
    Admin Grp : 0x0
    TE Metric : 1000
    TE APP LINK ATTR    :
      SABML-flag:Non-Legacy SABM-flags:   X
        Delay Min : 1000000 Max : 1000000
        TE Metric : 1000
    Adj-SID: Flags:v4BVL Weight:0 Label:524275
    Adj-SID: Flags:v4VL Weight:0 Label:524273
    Adj-SID: Flags:v6VL Weight:0 Label:524272
  MT IS Nbrs     :
    MT ID           : 2
    Nbr   : Dut-A.00
    Default Metric  : 10
    Sub TLV Len     : 220
    IPv6 Addr : 3ffe::101:303
    TE APP LINK ATTR    :
      SABML-flag:Non-Legacy SABM-flags:   X
        Delay Min : 1000000 Max : 1000000
        TE Metric : 1000
    Adj-SID: Flags:v6BVL Weight:0 Label:524274
    Adj-SID: Flags:v6VL Weight:0 Label:524272
    End.X-SID: 300::2000 flags:BP algo:0 weight:0 endpoint:End.X-PSP
    End.X-SID: 310::2000 flags:BP algo:128 weight:0 endpoint:End.X-PSP
    End.X-SID: 320::2000 flags:BP algo:129 weight:0 endpoint:End.X-PSP
    End.X-SID: 330::2000 flags:BP algo:130 weight:0 endpoint:End.X-PSP
    End.X-SID: 340::2000 flags:BP algo:131 weight:0 endpoint:End.X-PSP
    End.X-SID: 350::2000 flags:BP algo:132 weight:0 endpoint:End.X-PSP
    End.X-SID: 360::2000 flags:BP algo:133 weight:0 endpoint:End.X-PSP
```

2.1.3.3 Announcing ELC, MSD-ERLD, and MSD-BMI with IS-IS

IS-IS can announce node Entropy Label Capability (ELC), the Maximum Segment Depth (MSD) for node Entropy Readable Label Depth (ERLD) and the MSD for node Base MPLS Imposition (BMI). If needed, exporting the IS-IS extensions into BGP-LS requires no additional configuration. These extensions are standardized through *draft-ietf-isis-mpls-elc-10*, *Signaling Entropy Label Capability and Entropy Readable Label Depth Using IS-IS*, and RFC 8491, *Signaling Maximum SID Depth (MSD) Using IS-IS*.

When entropy and segment routing are enabled on a router, it automatically announces the ELC, ERLD, and BMI IS-IS values when IS-IS prefix attributes and router capabilities are announced. The following configuration logic is used:

- The router automatically announces ELC for host prefixes associated with an IPv4 or IPv6 node SID when **segment-routing** , **segment-routing entropy-label**, and **prefix-attributes-tlv** are enabled for IS-IS. Although the ELC capability is a node property, it is assigned to prefixes to allow inter-area or inter-AS signaling. Consequently, the prefix-attribute TLV must be enabled accordingly within IS-IS.
- The router announces the maximum node ERLD for IS-IS when **segment-routing** and **segment-routing entropy-label** are enabled together with **advertise-router-capability**.
- The router announces the maximum node MSD-BMI for IS-IS when **segment-routing** and **advertise-router-capability** are enabled.
- Exporting ELC, MSD-ERLD, and MSD-BMI IS-IS extensions into BGP-LS encoding is enabled automatically when database-export for BGP-LS is configured.
- The announced value for maximum node MSD-ERLD and MSD-BMI can be modified to a smaller number using the **override-bmi** and **override-erld** commands. This can be useful when services (such as EVPN) or more complex link protocols (such as Q-in-Q) are deployed. Provisioning correct ERLD and BMI values helps controllers and local Constrained Shortest Path First (CSPF) to construct valid segment routing label stacks to be deployed in the network.

Segment routing parameters are configured in the following contexts:

```
configure>router>isis>segment-routing>maximum-sid-depth
configure>router>isis>segment-routing>maximum-sid-depth>override-bmi value
configure>router>isis>segment-routing>maximum-sid-depth>override-erld value
```

2.1.3.4 Entropy label for IS-IS segment routing

The router supports the MPLS entropy label, as specified in RFC 6790, on IS-IS segment-routed tunnels. LSR nodes in a network can load-balance labeled packets in a more granular way than by hashing on the standard label stack. See the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide* for more information.

The router can announce Entropy Label Capability (ELC); however, it cannot process ELC signaling for IS-IS segment-routed tunnels. Instead, ELC is configured at the head-end LER using the **configure router isis entropy-label override-tunnel-elc** command. This command configures the router to ignore any advertisements for ELC that may or may not be received from the network, and instead to assume that the whole domain supports entropy labels.

2.1.3.5 IPv6 segment routing using MPLS encapsulation

This feature supports SR IPv6 tunnels in IS-IS MT=0. The user can configure a node SID for the primary IPv6 global address of a loopback interface, which is then advertised in IS-IS. IS-IS automatically assigns and advertises an adjacency SID for each adjacency with an IPv6 neighbor. After the node SID is resolved, it is used to install an IPv6 SR-ISIS tunnel in the TTM for use by the services.

2.1.3.5.1 IS-IS MT=0 extensions

The IS-IS MT=0 extensions support the advertising and resolution of the prefix SID sub-TLV within the IP reach TLV-236 (IPv6), as defined in RFC 5308. The adjacency SID is still advertised as a sub-TLV of the Extended IS Reachability TLV 22, as defined in RFC 5305, *IS-IS Extensions for Traffic Engineering*, as in the case of an IPv4 adjacency. The router sets the V-flag and I-flag in the SR-capabilities sub-TLV to indicate that it can process SR MPLS-encapsulated IPv4 and IPv6 packets on its network interfaces.

2.1.3.5.2 Service and forwarding contexts supported

The service and forwarding contexts supported with the SR IS-IS IPv6 tunnels are:

- SDP of type **sr-isis** with **far-end** option using an IPv6 address
- VLL, VPLS, IES/VPRN spoke-interface, and R-VPLS
- support of PW redundancy within Epipe/Ipipe VLL, Epipe spoke termination on VPLS and R-VPLS, and Epipe/Ipipe spoke termination on IES/VPRN
- IPv6 static route resolution to indirect next hop using Segment Routing IPv6 tunnel
- remote mirroring and Layer 3 encapsulated lawful interface

2.1.3.5.3 Services using SDP with an SR IPv6 tunnel

The MPLS SDP of type **sr-isis** with a **far-end** option using an IPv6 address is supported. Note the SDP must have the same IPv6 **far-end** address, used by the control plane for the T-LDP session, as the prefix of the node SID of the SR IPv6 tunnel.

```
configure
        – service
            – [no] sdp sdp-id mpls
                – [no] far-end ipv6-address
                – sr-isis
                – no sr-isis
```

The **bgp-tunnel**, **lsp**, **sr-te lsp**, **sr-ospf,** and **mixed-lsp-mode** commands are blocked within the SDP configuration context when the far end is an IPv6 address.

SDP admin groups are not supported with an SDP using an SR IPv6 tunnel, or with SR-OSPF for IPv6 tunnels, and the attempt to assign them is blocked in the CLI.

Services that use LDP control plane such as T-LDP VPLS and R-VPLS, VLL, and IES/VPRN spoke interface have the spoke SDP (PW) signaled with an IPv6 T-LDP session because the **far-end** option is configured to an IPv6 address. The spoke SDP for these services binds to an SDP that uses an SR IPv6 tunnel where the prefix matches the **far-end** address. SR OS also supports the following:

- the IPv6 PW control word with both data plane packets and VCCV OAM packets
- hash label and entropy label, with the above services
- network domains in VPLS

The PW switching feature is not supported with LDP IPv6 control planes. As a result, the CLI does not allow the user to enable the **vc-switching** option whenever one or both spoke SDPs uses an SDP that has the **far-end** configured as an IPv6 address.

L2 services that use BGP control plane such as dynamic MS-PW, BGP-AD VPLS, BGP-VPLS, BGP-VPWS, and EVPN MPLS cannot bind to an SR IPv6 tunnel because a BGP session to a BGP IPv6 peer does not support advertising an IPv6 next hop for the L2 NLRI. As a result, these services do not auto-generate SDPs using an SR IPv6 tunnel. In addition, they skip any provisioned SDPs with **far-end** configured to an IPv6 address when the **use-provisioned-sdp** option is enabled.

SR OS also supports multi homing with T-LDP active/standby FEC 128 spoke SDP using SR IPv6 tunnel to a VPLS/B-VPLS instance. BGP multi homing is not supported because BGP IPv6 does not support signaling an IPv6 next hop for the L2 NLRI.

The Shortest Path Bridging (SPB) feature works with spoke SDPs bound to an SDP that uses an SR IPv6 tunnel.

2.1.3.6 Segment routing mapping server function for IPv4 prefixes

The mapping server feature supports the configuration and advertisement, in IS-IS, of the node SID index for prefixes of routers in the LDP domain. This is performed in the router acting as a mapping server and using a prefix-SID sub-TLV within the SID/label binding TLV in IS-IS.

Use the following command syntax to configure the SR mapping database in IS-IS:

```
configure
        – router
            – [no] isis
                – segment-routing
                – no segment-routing
                    – mapping-server
                        – sid-map node-sid {index 0..4294967295 [range  0..65535]} prefix {{ip-
address/mask} | {ip-address}{netmask}} [set-flags {s}] [level {1 | 2 | 1/2}]
                        – no sid-map node-sid index 0..4294967295
```

The user enters the node SID index, for one prefix or a range of prefixes, by specifying the first index value and, optionally, a range value. The default value for the range option is 1. Only the first prefix in a consecutive range of prefixes must be entered. If the user enters the first prefix with a mask lower than 32, the SID/label binding TLV is advertised, but a router that receives it does not resolve the prefix SID and instead generates a trap.

The **no** form of the **sid-map** command deletes the range of node SIDs beginning with the specified index value. The **no** form of the **mapping-server** command deletes all node SID entries in the IS-IS instance.

The S-flag indicates to the IS-IS routers in the network that the flooding scope of the SID/label binding TLV is the entire domain. In that case, a router receiving the TLV advertisement leaks it between IS-IS levels. If leaked from Level 2 to Level 1, the D-flag must be set; this prevents the TLV from being leaked back into level 2. Otherwise, the S-flag is clear by default and routers receiving the mapping server advertisement do not leak the TLV.

Note:
SR OS does not leak this TLV between IS-IS instances and does not support the multitopology SID/label binding TLV format. In addition, the user can specify the flooding scope of the mapping server for the generated SID/label binding TLV using the **level** option. This option allows further narrowing of the flooding scope configured under the router IS-IS level-capability for one or more SID/label binding TLVs if required. The default flooding scope of the mapping server is L1 or L2, which can be narrowed by what is configured under the router IS-IS level-capability.

The A-flag indicates that a prefix for which the mapping server prefix SID is advertised is directly attached. The M-flag advertises a SID for a mirroring context to provide protection against the failure of a service node. None of these flags are supported on the mapping server; the mapping client ignores them.

Each time a prefix or a range of prefixes is configured in the SR mapping database in any routing instance, the router issues for this prefix, or range of prefixes, a prefix-SID sub-TLV within an IS-IS SID/label binding TLV in that instance. The flooding scope of the TLV from the mapping server is determined as previously described. No further check of the reachability of that prefix in the mapping server route table is performed. No check of the SID index is performed to determine whether the SID index is a duplicate of an existing prefix in the local IGP instance database or if the SID index is out of range with the local SRGB.

2.1.3.6.1 IP prefix resolution for segment routing mapping server

The following processing rules apply for IP prefix resolution:

- SPF calculates the next hops, up to **max-ecmp**, to reach a destination node.
- Each prefix inherits the next hops of one or more destination nodes advertising it.
- A prefix advertised by multiple nodes, all reachable with the same cost, inherits up to **max-ecmp** next hops from the advertising nodes.
- The next-hop selection value, up to **max-ecmp**, is based on sorting the next hops by:
  - lowest next-hop router ID
  - lowest interface index, for parallel links to same router ID

  Each next hop keeps a reference to the destination nodes from which it was inherited.

2.1.3.6.2 Prefix SID resolution for segment routing mapping server

This section describes the processing rules for prefix SID resolution.

- For a specific prefix, IGP selects the SID value among multiple advertised values in the following order:
  1. the local intra-area SID owned by this router
  2. the prefix SID sub-TLV advertised within an IP reach TLV

     If multiple SIDs exist, the IGP selects the SID corresponding to the destination router or the ABR with the lowest system ID that is reachable using the first next hop of the prefix.
  3. the IS-IS SID and label binding TLV from the mapping server

     If multiple SIDs exist, the IGP selects the following, using the preference rules in *draft-ietf-spring-conflict-resolution-05* when applied to the SRMS entries of the conflicting SIDs. The order of these rules is as follows:

     a. smallest range

b. smallest starting address

c. smallest algorithm

d. smallest starting SID

Note: If an L1L2 router acts as a mapping server and also re-advertises the mapping server prefix SID from other mapping servers, the redistributed mapping server prefix SID is preferred by other routers resolving the prefix, which may result in not selecting the mapping server respecting these rules.

- The selected SID is used with all ECMP next hops from the IP prefix resolution in step toward all destination nodes or ABR nodes that advertised the prefix.
- If duplicate prefix SIDs exist for different prefixes after these processing steps are completed, the first SID that is processed is programmed according to its corresponding prefix. Subsequent SIDs cause a duplicate SID trap message and are not programmed. The corresponding prefixes are still resolved and programmed normally using IP next-next-hops.

2.1.3.6.3 SR tunnel programming for segment routing mapping server

The following processing rules apply for SR tunnel programming:

- If the prefix SID is resolved from a prefix SID sub-TLV advertised within an IP Reachability TLV, one of the following applies:
  - The SR ILM label is swapped to an SR NHLFE label, as in SR tunnel resolution when the next hop of the IS-IS prefix is SR-enabled.
  - The SR ILM label is stitched to an LDP FEC of the same prefix when either the next hop of the IS-IS prefix is not SR-enabled (no SR NHLFE) or an import policy rejects the prefix (SR NHLFE is deprogrammed).

    The LDP FEC can also be resolved by using the same or a different IGP instance as that of the prefix SID sub-TLV or by using a static route.
- If the prefix SID is resolved from a mapping server advertisement, one of the following applies:
  - The SR ILM label is stitched to an LDP FEC of the same prefix, if one exists. The stitching is performed even if an import policy rejects the prefix in the local IS-IS instance.

    The LDP FEC can also be resolved by using a static route, a route within an IS-IS instance, or a route within an OSPF instance. The IS-IS or OSPF instances can be the same as, or different form the IGP instance that advertised the mapping server prefix SID sub-TLV.
  - The SR ILM label is swapped to an SR NHLFE label. This is only possible if a route is exported from another IGP instance into the local IGP instance without propagating the prefix SID sub-TLV with the route. Otherwise, the SR ILM label is swapped to an SR NHLFE label toward the stitching node.

2.1.4 Segment routing shortest path forwarding with OSPF

This section describes the segment routing shortest path forwarding with OSPF.

2.1.4.1 OSPFv2 control protocol changes

The following TLVs and sub-TLVs are defined in *draft-ietf-ospf-segment-routing-extensions-04* and are required for the implementation of segment routing in OSPF:

- the prefix SID sub-TLV part of the OSPFv2 Extended Prefix TLV
- the prefix SID sub-TLV part of the OSPFv2 Extended Prefix Range TLV
- the adjacency SID sub-TLV part of the OSPFv2 Extended Link TLV
- SID/Label Range capability TLV
- SR-Algorithm capability TLV

This section describes the behaviors and limitations of OSPF support of segment routing TLVs and sub-TLVs.

SR OS originates a single prefix SID sub-TLV per OSPFv2 Extended Prefix TLV and processes the first one only if multiple prefix SID sub-TLVs are received within the same OSPFv2 Extended Prefix TLV.

SR OS encodes the 32-bit index in the prefix SID sub-TLV. The 24-bit label or variable IPv6 SID is not supported.

SR OS originates a prefix SID sub-TLV with the following encoding of the flags:

- The NP-Flag is always set. The label for the prefix SID is pushed by the PHP router when forwarding to this router. The SR OS PHP router processes a received prefix SID with the NP-flag set to zero and uses implicit-null for the outgoing label toward the router that advertised it.
- The M-Flag is always unset because SR OS does not support originating a mapping server prefix-SID sub-TLV.
- The E-flag is always set to zero. An SR OS PHP router, however, processes a received prefix SID with the E-flag set to 1, and when the NP-flag is also set to 1, it pushes explicit-null for the outgoing label toward the router that advertised it.
- The V-flag is always set to 0 to indicate an index value for the SID.
- The L-flag is always set to 0 to indicate that the SID index value is not locally significant.
- The algorithm field is set to zero to indicate Shortest Path First (SPF) algorithm based on link IGP metric or to the flexible algorithm number.

SR OS resolves a prefix SID received within an Extended Prefix TLV based on the following route preference:

- SID received via an intra-area route in a prefix SID sub-TLV part of the Extended Prefix TLV
- SID received via an inter-area route in a prefix SID sub-TLV part of the Extended Prefix TLV

SR OS originates an adjacency SID sub-TLV with the following encoding of the flags:

- The B-flag is set to zero and is not processed on receipt.
- The V-flag is always set.
- The L-flag is always set.
- The G-flag is not supported.
- The weight octet is not supported and is set to all zeros.

An adjacency SID is assigned to next hops over both the primary and secondary interfaces.

SR OS can originate the OSPFv2 Extended Prefix Range TLV as part of the Mapping Server feature and can process it properly, if received. Consider the following rules and limitations:

- Only the prefix SID sub-TLV within the TLV is processed and the ILMs are installed if the prefixes are resolved.
- The range and address prefix fields are processed. Each prefix is resolved separately.
- If the same prefix is advertised with both a prefix SID sub-TLV in an IP-reachability TLV and a mapping server Prefix-SID sub-TLV, the resolution follows the following route preference:
  - the SID received via an intra-area route in a prefix SID sub-TLV part of Extended Prefix TLV
  - the SID received via an inter-area route in a prefix SID sub-TLV part of Extended Prefix TLV
  - the SID received via an intra-area route in a prefix SID sub-TLV part of a OSPFv2 Extended Range Prefix TLV
  - the SID received via an inter-area route in a prefix SID sub-TLV part of a OSPFv2 Extended Range Prefix TLV
- No leaking of the entire TLV is performed between areas. An ABR does not propagate the prefix-SID sub-TLV from the Extended Prefix Range TLV into an Extended Prefix TLV if the latter is propagated between areas.
- The mapping server which advertised the OSPFv2 Extended Prefix Range TLV does not need to be in the shortest path for the FEC prefix.
- If the same FEC prefix is advertised in multiple OSPFv2 Extended Prefix Range TLVs by different routers, the SID in the TLV on the first router that is reachable is used. If that router becomes unreachable, the next reachable one is used.
- There is no check to determine whether the contents of the OSPFv2 Extended Prefix Range TLVs received from different mapping servers are consistent.
- Any other sub-TLV (for example, the ERO metric and unnumbered interface ID ERO) is ignored, but the user can use the IGP **show** command to see the octets of the received but not supported sub-TLVs.

SR OS supports propagation on the ABR of external prefix LSAs into other areas with routeType set to *3* as per *draft-ietf-ospf-segment-routing-extensions-04*.

SR OS supports propagation on the ABR of external prefix LSAs with route type 7 from a Not-So-Stubby Area (NSSA) into other areas with route type set to 5 as per *draft-ietf-ospf-segment-routing-extensions-04*. SR OS does not support propagating the prefix SID sub-TLV between OSPF instances.

When the user configures an OSPF import policy, the outcome of the policy applies to prefixes resolved in the RTM and the corresponding tunnels in the TTM. A prefix removed by the policy does not appear as both a route in the RTM and as a segment routing tunnel in the TTM.

2.1.4.2 OSPFv3 control protocol changes

The OSPFv3 extensions support the following TLVs:

- **a prefix SID that is a sub-TLV of the OSPFv3 prefix TLV**

  The OSPFv3 prefix TLV is a new top-level TLV of the extended prefix LSA introduced in *draft-ietf-ospf-ospfv3-lsa-extend*. The OSPFv3 instance can operate in either LSA sparse mode or extended LSA mode.

  The **config**>**router**>**extended-lsa only** command advertises the prefix SID sub-TLV in the extended LSA format in both cases.

- **an adjacency SID that is a sub-TLV of the OSPFv3 router-link TLV**

  The OSPFv3 router-link TLV is a new top-level TLV in the extended router LSA introduced in *draft-ietf-ospf-ospfv3-lsa-extend*. The OSPFv3 instance can operate in either LSA sparse mode or extended LSA mode. The **config**>**router**>**extended-lsa only** command advertises the adjacency SID sub-TLV in the extended LSA format in both cases.

- **the SR-Algorithm TLV and the SID/Label range TLV**

  Both of these TLVs are part of the TLV-based OSPFv3 Router Information Opaque LSA defined in RFC 7770.

2.1.4.3 Announcing ELC, MSD-ERLD, and MSD-BMI with OSPF

OSPF can announce node ELC, MSD for node ERLD, and the MSD for node BMI. If needed, exporting these OSPF extensions into BGP-LS requires no additional configuration. These extensions are standardized through *draft-ietf-ospf-mpls-elc-12*, *Signaling Entropy Label Capability and Entropy Readable Label-stack Depth Using OSPF*, and RFC 8476, *Signaling Maximum SID Depth (MSD) Using OSPF*.

When entropy and segment routing is enabled on a router, it automatically announces the ELC, ERLD, and BMI OSPF values. The following configuration logic is used:

- ELC is automatically announced for host prefixes associated with a node SID when **segment-routing** and **segment-routing entropy-label** are enabled for OSPF.
- The router maximum node ERLD is announced for OSPF when **segment-routing** and **segment-routing entropy-label** are enabled together with **advertise-router-capability**.
- The router maximum node MSD-BMI for OSPF is announced when **segment-routing advertise-router-capability** are enabled.
- Exporting ELC, MSD-ERLD and MSD-BMI OSPF extensions into BGP-LS encoding occurs automatically when **database-export** for BGP-LS is configured.
- The announced value for maximum node MSD-ERLD and MSD-BMI can be modified to a smaller number using the **override-bmi** and **override-erld** commands. This can be useful when services (such as EVPN) or more complex link protocols (such as Q-in-Q) are deployed. Provisioning correct ERLD and BMI values helps controllers and local CSPF to construct valid segment routing label stacks to be deployed in the network.

Segment routing parameters are configured in the following contexts:

```
configure>router>ospf>segment-routing
configure>router>ospf>segment-routing>override-bmi value
configure>router>ospf>segment-routing>override-erld value
```

2.1.4.4 Entropy label for OSPF segment routing

The router supports the MPLS entropy label, as specified in RFC 6790, on OSPF segment-routed tunnels. LSR nodes in a network can load-balance labeled packets in a more granular way than by hashing on the standard label stack. See the *MPLS Guide* for more information.

The router can announce ELC; however, it cannot process ELC signaling for OSPF segment-routed tunnels. Instead, ELC is configured at the head-end LER using the **configure router ospf entropy-label override-tunnel-elc** command. This command configures the router to ignore any advertisements for ELC

that may or may not be received from the network, and to assume that the whole domain supports entropy labels.

2.1.4.5 IPv6 segment routing using MPLS encapsulation in OSPFv3

This feature supports SR IPv6 tunnels in OSPFv3 instances 0 to 31. The user can configure a node SID for the primary IPv6 global address of a loopback interface, which then gets advertised in OSPFv3. OSPFv3 automatically assigns and advertises an adjacency SID for each adjacency with an IPv6 neighbor. After the node SID is resolved, it is used to install an IPv6 SR-OSPF3 tunnel in the TTMv6 for use by the routes and services.

2.1.4.6 Segment routing mapping server for IPv4 prefixes

The mapping server feature configures and advertises, in OSPF, the node SID index for prefixes of routers in the LDP domain. This is performed in the router acting as a mapping server and using a prefix-SID sub-TLV within an OSPF Extended Prefix Range TLV.

Use the following command syntax to configure the SR mapping database in OSPF:

```
configure
        – router
            – [no] ospf
                – segment-routing
                – no segment-routing
                    – mapping-server
                        – sid-map node-sid {index 0 to 4294967295 [range 1 to 65535]} prefix
 {{ip-address/mask}|{netmask}}[scope {area area-id | as}]
                        – no sid-map node-sid index 0 to 4294967295
```

The user enters the node SID index, for one prefix or a range of prefixes, by specifying the first index value and, optionally, a range value. The default value for the **range** option is 1. Only the first prefix in a consecutive range of prefixes must be entered. If the user enters the first prefix with a mask lower than 32, the OSPF Extended Prefix Range TLV is advertised, but a router that receives the OSPF Extended Prefix Range TLV does not resolve the SID and instead generates a trap.

The **no** form of the **sid-map** command deletes the range of node SIDs beginning with the specified index value. The **no** form of the **mapping-server** command deletes all node SID entries in the OSPF instance.

Use the **scope** option to specify the flooding scope of the mapping server for the generated OSPF Extended Prefix Range TLV. There is no default value. If the scope is a specific area, the TLV is flooded only in that area.

An ABR that propagates an intra-area OSPF Extended Prefix Range TLV flooded by the mapping server in that area into other areas sets the inter-area flag (IA-flag). The ABR also propagates the TLV if it is received with the IA-flag set from other ABR nodes but only from the backbone to leaf areas and not leaf areas to the backbone. However, if the identical TLV was advertised as an intra-area TLV in a leaf area, the ABR does not flood the inter-area TLV into that leaf area.

Note: SR OS does not leak the OSPF Extended Prefix Range TLV between OSPF instances.

Each time a prefix or a range of prefixes is configured in the SR mapping database in any routing instance, the router issues for this prefix, or range of prefixes, a prefix-SID sub-TLV within an OSPF Extended Prefix Range TLV in that instance. The flooding scope of the TLV from the mapping server is determined

as previously described. The reachability of that prefix in the mapping server route table is not checked. Additionally, the SR OS does not check whether the SID index is a duplicate of an existing prefix in the local IGP instance database or if the SID index is out of range with the local SRGB.

2.1.4.6.1 IP prefix resolution for segment routing mapping server

The following processing rules apply for IP prefix resolution:

- SPF calculates the next hops, up to **max-ecmp**, to reach a destination node.
- Each prefix inherits the next hops of one or more destination nodes advertising it.
- A prefix advertised by multiple nodes, all reachable with the same cost, inherits up to **max-ecmp** next hops from the advertising nodes.
- The next-hop selection value, up to **max-ecmp**, is based on sorting the next hops by:
  - lowest next-hop router ID
  - lowest interface index, for parallel links to same router ID

  Each next hop keeps a reference to the destination nodes from which it was inherited.

2.1.4.6.2 Prefix SID resolution for segment routing mapping server

The following processing rules apply for prefix SID resolution:

- For a specific prefix, IGP selects the SID value among multiple advertised values in the following order:
  1. local intra-area SID owned by this router
  2. prefix SID sub-TLV advertised within a OSPF Extended Prefix TLV

     If multiple SIDs exist, select the SID corresponding to the destination router or ABR with the lowest OSPF router ID which is reachable via the first next hop of the prefix
  3. OSPF Extended Prefix Range TLV from mapping server

     If multiple SIDs exist, select the following, using the preference rules in *draft-ietf-spring-conflict-resolution-05* when applied to the SRMS entries of the conflicting SIDs. The order of these rules is as follows:

     a. smallest range

     b. smallest starting address

     c. smallest algorithm

     d. smallest starting SID
- The selected SID is used with all ECMP next hops from step 1 toward all destination nodes or ABR nodes which advertised the prefix.
- If duplicate prefix SIDs exist for different prefixes after above steps, the first SID which is processed is programmed for its corresponding prefix. Subsequent SIDs causes a duplicate SID trap message and are not programmed. The corresponding prefixes are still resolved normally using IP next hops.

2.1.4.6.3 SR tunnel programming for segment routing mapping server

The following processing rules apply for SR tunnel programming:

- If the prefix SID is resolved from a prefix SID sub-TLV advertised within an OSPF Extended Prefix TLV, one of the following applies.
  - The SR ILM label is swapped to an SR NHLFE label as in SR tunnel resolution when the next hop of the OSPF prefix is SR-enabled.
  - The SR ILM label is stitched to an LDP FEC of the same prefix when either the next hop of the OSPF prefix is not SR enabled (no SR NHLFE) or an import policy rejects the prefix (SR NHLFE deprogrammed).

    The LDP FEC can also be resolved using the same or a different IGP instance as that of the prefix SID sub-TLV or using a static route.
- If the prefix SID is resolved from a mapping server advertisement, one of the following applies.
  - The SR ILM label is stitched to an LDP FEC of the same prefix, if one exists. The stitching is performed even if an import policy rejects the prefix in the local OSPF instance.

    The LDP FEC can also be resolved using a static route, a route within an OSPF instance, or a route within an OSPF instance. The latter two can be the same as, or different from the IGP instance that advertised the mapping server prefix SID sub-TLV.
  - The SR ILM label is swapped to an SR NHLFE label toward the stitching node.

2.1.5 Segment routing with BGP

Segment routing allows a router, potentially by action of an SDN controller, to source route a packet by prepending a segment router header containing an ordered list of SIDs. Each SID can be viewed as a topological or service-based instruction. A SID can have a local impact to one particular node or it can have a global impact within the SR domain, such as the instruction to forward the packet on the ECMP-aware shortest path to reach a prefix, "P". With SR-MPLS, each SID is an MPLS label and the complete SID list is a stack of labels in the MPLS header.

To ensure that all the routers in a network domain have a common understanding of a topology SID, the association of the SID with an IP prefix must be propagated by a routing protocol. Traditionally, this is done by an IGP protocol; however, in some cases, the meaning of a SID may need to be propagated across network boundaries that extend beyond IGP protocol boundaries. For these cases, BGP can carry the association of an SR-MPLS SID with an IP prefix by attaching a prefix SID BGP path attribute to an IP route belonging to a labeled-unicast address family.

The prefix SID attribute attached to a labeled-unicast route for prefix P advertises a SID corresponding to the network-wide instruction to forward the packet along the ECMP-aware BGP-computed best path or paths to reach P. The prefix SID attribute is an optional transitive BGP path attribute with type code 40. This attribute encodes a 32-bit label index into the SRGB space and can provide details about the SRGB space of the originating router. The encoding of this BGP path attribute and its semantics are further described in *draft-ietf-idr-bgp-prefix-sid*.

Using the **block-prefix-sid** BGP command, an SR OS router with upgraded software that processes the prefix SID attribute can prevent it from propagating outside the segment routing domain where it is applicable. The **block-prefix-sid** command removes the prefix SID attribute from all routes sent and received to and from the iBGP and eBGP peers included in the scope of the command. By default, the attribute propagates without restriction.

SR OS attaches a meaning to a prefix SID attribute only when it is attached to routes belonging to the labeled-unicast IPv4 and labeled-unicast IPv6 address families. When attached to routes of unsupported address families, the prefix SID attribute is ignored but still propagated, as with any other optional transitive attribute.

Segment routing must be administratively enabled under BGP using the **config router bgp segment-routing no shutdown** command. When segment routing is configured, the following considerations apply:

- For BGP to redistribute a static or IGP route for a /32 IPv4 prefix as a label-ipv4 route, or a /128 IPv6 prefix as a label-ipv6 route, with a prefix SID attribute, a **route-table-import** policy with an **sr-label-index** action is required.
- For BGP to add or modify the prefix SID attribute in a received label-ipv4 or label-ipv6 route, a BGP **import** policy with an **sr-label-index** action is required.
- For BGP to advertise a label-ipv4 or label-ipv6 route with an incoming datapath label based on the attached prefix SID attribute when BGP segment routing is disabled, new label values assigned to label-ipv4 or label-ipv6 routes come from the dynamic label range of the router and have no network-wide impact.

To enable BGP segment routing, the base router BGP instance must be associated with a **prefix-sid-range**. This command specifies which SRGB label block to use (for example, to allocate labels). This command also specifies which SRGB label block to advertise in the Originator SRGB TLV of the prefix SID attribute. The **global** parameter value indicates that BGP should use the SRGB as configured under **config**>**router**>**mpls-labels**>**sr-labels**. The **start-label** and **max-index** parameters are used to restrict the BGP prefix SID label range to a subset of the global SRGB.

Note: The **start-label** and **max-index** values must be within the global SRGB range or the command fails.

This is useful when partitioning of the SRGB into non-overlapping subranges dedicated to different IGP/BGP protocol instances is required. Segment routing under BGP must be shutdown before any changes can be made to the **prefix-sid-range** command.

A unique label-index value is assigned to each unique IPv4 or IPv6 prefix that is advertised with a BGP prefix SID. If label-index N1 is assigned to a BGP-advertised prefix P1, and N1 plus the SRGB start label creates a label value that conflicts with another SR programmed LFIB entry, the conflict situation is addressed according to the following rules:

- If the conflict is with another BGP route for prefix P2 that was advertised with a prefix SID attribute, all the conflicting BGP routes for P1 and P2 are advertised with a normal BGP-LU label from the dynamic label range.
- If the conflict is with an IGP route and BGP is not attempting to redistribute that IGP route as a label-ipv4 or label-ipv6 route with a route-table-import policy action that uses the **prefer-igp** keyword in the **sr-label-index** command, the IGP route takes priority and the BGP route is advertised with a normal BGP-LU label from the dynamic label range.
- If the conflict is with an IGP route and BGP is attempting to redistribute that IGP route as a label-ipv4 or label-ipv6 route with a route-table-import policy action that uses the **prefer-igp** keyword in the **sr-label-index** command, this is not considered a conflict and BGP uses the IGP-signaled label-index to derive its advertised label. This has the effect of stitching the BGP segment routing tunnel to the IGP segment routing tunnel.

Note: This use of the **prefer-igp** option is only possible when BGP segment routing is configured with the **prefix-sid-range global** command.

Any /32 label-ipv4 or /128 label-ipv6 BGP routes containing a prefix SID attribute are resolvable and used in the same way as /32 label-ipv4 or /128 label-ipv6 routes without a prefix SID attribute. These routes are installed in the route table and tunnel table (unless **disable-route-table-install** or **selective-label-ipv4-install** are enabled). These routes can have ECMP next hops or FRR backup next hops and be used as transport tunnels for any service that supports BGP-LU transport.

Note:
Receiving a /32 label-ipv4 or /128 label-ipv6 route with a prefix SID attribute does not create a tunnel in the segment-routing database; it only creates a label swap entry when the route is re-advertised with a new next hop.

It is recommended the first SID in any SID-list of an SR policy should not be based on a BGP prefix SID; if this recommendation is not followed, then the SID-list may appear to be valid but the datapath is not programmed correctly. However, it is acceptable to use a BGP prefix SID for any SID other than first SID in any SR policy.

2.1.6 Segment routing operational procedures

This section describes the segment routing operational procedures.

2.1.6.1 Prefix advertisement and resolution

After segment routing is successfully enabled in the IS-IS or OSPF instance, the router performs the following operations:

1. The router advertises the Segment Routing Capability sub-TLV to routers in all areas or levels of this IGP instance. Only neighbors with which the router established an adjacency can interpret the SID and label range information and use it for calculating the label to swap to or push for a specific resolved prefix SID.
2. The router advertises the assigned index for each configured node SID in the new prefix SID sub-TLV with the N-flag (node SID flag) set. The segment routing module then programs the ILM with a pop operation for each local node SID in the datapath.
3. The router assigns and advertise an adjacency SID label for each formed adjacency over a network IP interface in the Adjacency SID sub-TLV, according to the following rules and limitations:
   - The Adjacency SID sub-TLV is advertised for both numbered and unnumbered network IP interfaces.
   - The Adjacency SID is not advertised for an IES interface because access interfaces do not support MPLS.
   - The Adjacency SID sub-TLV must be unique per instance and per adjacency.

     ISIS MT=0 can establish an adjacency for both IPv4 and IPv6 address families over the same link. In this case, a different adjacency SID is assigned to each next hop. However, the existing IS-IS implementation assigns a single Protect-Group ID (PG-ID) to the adjacency and therefore when the state machine of a BFD session tracking the IPv4 or IPv6 next hop times out, an action is triggered for the prefixes of both address families over that adjacency.

   The segment routing module programs the ILM with a swap to an implicit null label operation for each advertised adjacency SID.
4. The router resolves received prefixes. If a prefix SID sub-TLV exists, the segment routing module programs the ILM with a swap operation and an LTN with a push operation, both pointing to the primary/LFA NHLFE. A segment routing tunnel is also added to the TTM. If a node SID resolves over an IES interface, the datapath is not programmed and a trap message is generated. Only next-hops of an ECMP set corresponding to network IP interfaces are programmed in the datapath; next-hops corresponding to IES interfaces are not programmed. If the user configures the interface as network

on one side and IES on the other side, MPLS packets for the segment routing tunnel received on the access side are dropped.

Note: LSA filtering causes SIDs not to be sent in one direction, which means that some node SIDs are resolved in parts of the network upstream of the advertisement suppression.

When the user enables segment routing in an IGP instance, the main SPF and LFA SPF are computed normally and the primary next-hop and LFA backup next-hop for a received prefix are added to RTM without the label information advertised in the prefix SID sub-TLV. In all cases, the SR tunnel is not added into RTM.

See the following sections for more information about all TLVs and sub-TLVs for both IS-IS and OSPF protocols.

- IS-IS control protocol changes
- OSPFv2 control protocol changes
- OSPFv3 control protocol changes

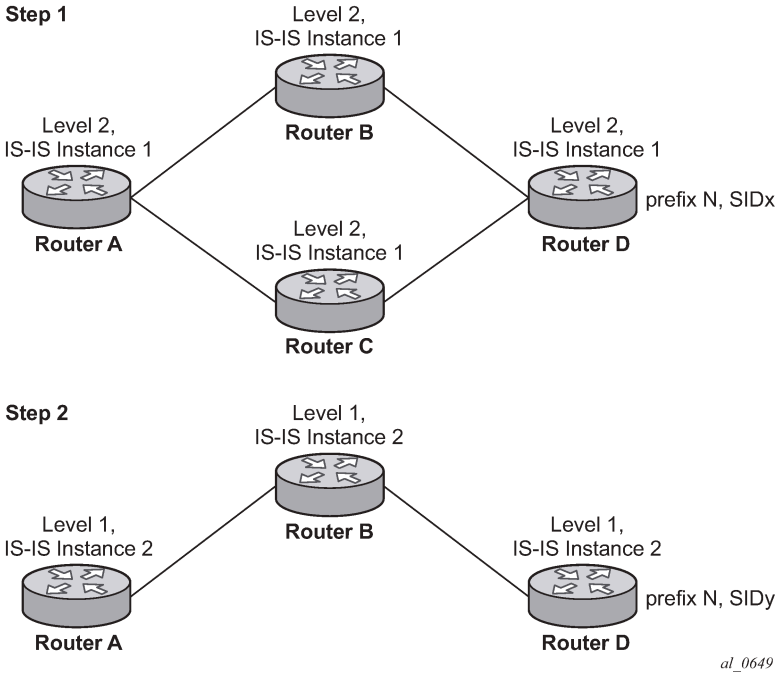
2.1.6.2 Error and resource exhaustion handling

The router performs the procedures described in the following sections when resolving a node SID prefix.

2.1.6.2.1 Supporting multiple topologies for the same destination prefix

SR OS can assign different prefix-SID indexes and labels to the same prefix in different IGP instances. While other routers that receive these prefix SIDs program a single route into the RTM based on the winning instance ID as per RTM route type preference, SR OS adds two tunnels to this destination prefix in the TTM. This supports multiple topologies for the same destination prefix. Figure 2: Programming multiple tunnels to the same destination shows two different instances (Level 2, IS-IS instance 1 and Level 1, IS-IS instance 2), where Router D has the same prefix destination with different SIDs (SIDx and SIDy).

Figure 2: Programming multiple tunnels to the same destination



Assume the following route-type preference in the RTM and tunnel-type preference in the TTM are configured:

- ROUTE_PREF_ISIS_L1_INTER (RTM) 15
- ROUTE_PREF_ISIS_L2_INTER (RTM) 18
- ROUTE_PREF_ISIS_TTM 10

Note: The TTM tunnel type preference is not used by the segment routing module. It is put in the TTM and is used by other applications, such as VPRN auto-bind and BGP shortcut, to select a TTM tunnel.

1. Router A performs the following resolution within the single Level 2, IS-IS instance 1. All metrics are the same and ECMP = 2.
   - For prefix N, the RTM entry is:
     - prefix N
     - nhop1 = B
     - nhop2 = C
     - preference 18
   - For prefix N, the SR tunnel TTM entry is:
     - tunnel-id 1: prefix N-SIDx
     - nhop1 = B
     - nhop2 = C
     - tunl-pref 10

2. Add Level 1, IS-IS instance 2 in the same configuration, but in routers A, B, and C only.
   - For prefix N, the RTM entry is:
     – prefix N
     – nhop1 = B
     – preference 15

     The RTM prefers Level 1 route over Level 2 route.
   - For prefix N, there are two SR tunnel entries in TTM:

     SR entry for Level 2:
     – tunnel-id 1: prefix N-SIDx
     – nhop1 = B
     – nhop2= C
     – tunl-pref 10

     The SR entry for Level 1 is tunnel-id 2: prefix N-SIDy.

2.1.6.2.2 Resolving received SID indexes or labels to different routes of the same prefix within the same IGP instance

The router can perform the following variations of this procedure:

- When the SR OS does not allow assigning the same SID index or label to different routes of the same prefix within the same IGP instance, the router resolves only one of the duplicate SIDs if the SIDs are received from another segment routing implementation and the SIDs are based on the RTM active route selection.
- When SR OS does not allow assigning different SID indexes or labels to different routes of the same prefix within the same IGP instance, the router resolves only one of the duplicate SIDs if the SIDs are received from another segment routing implementation and the SIDs are based on the RTM active route selection.

  The selected SID is used for ECMP resolution to all neighbors. If the route is inter-area and the conflicting SIDs are advertised by different ABRs, ECMP toward all ABRs uses the selected SID.

2.1.6.2.3 Checking for SID errors before programming the ILM and NHLFE

If any of the following conditions are true, the router logs a trap, generates a syslog error message, and does not program the ILM and NHLFE for the prefix SID:

- The received prefix SID index falls outside of the locally configured SID range.
- One or more resolved ECMP next-hops for a received prefix SID did not advertise the SR Capability sub-TLV.
- The received prefix SID index falls outside the advertised SID range of one or more resolved ECMP next-hops.

2.1.6.2.4 Programming ILM/NHLFE for duplicate prefix-SID indexes/labels for different prefixes

The router can perform the following variations of this procedure:

- For received duplicate prefix-SID indexes or labels for different prefixes within the same IGP instance, the router:
  - programs the ILM/NHLFE for the first prefix-SID index or label
  - logs a trap and generates a syslog error message
  - does not program the subsequent prefix-SID index or label in the datapath
- For received duplicate prefix-SID indexes or labels for different prefixes across IGP instances, there are two options.
  - In the global SID index range mode of operation, the resulting ILM label value is the same across the IGP instances. The router:
    - programs ILM/NHLFE for the prefix of the winning IGP instance based on the RTM route-type preference
    - logs a trap and generates a syslog error message
    - does not program the subsequent prefix SIDs in the datapath
  - In the per-instance SID index range mode of operation, the resulting ILM label has different values across the IGP instances. The router programs ILM/NHLFE for each prefix as expected.

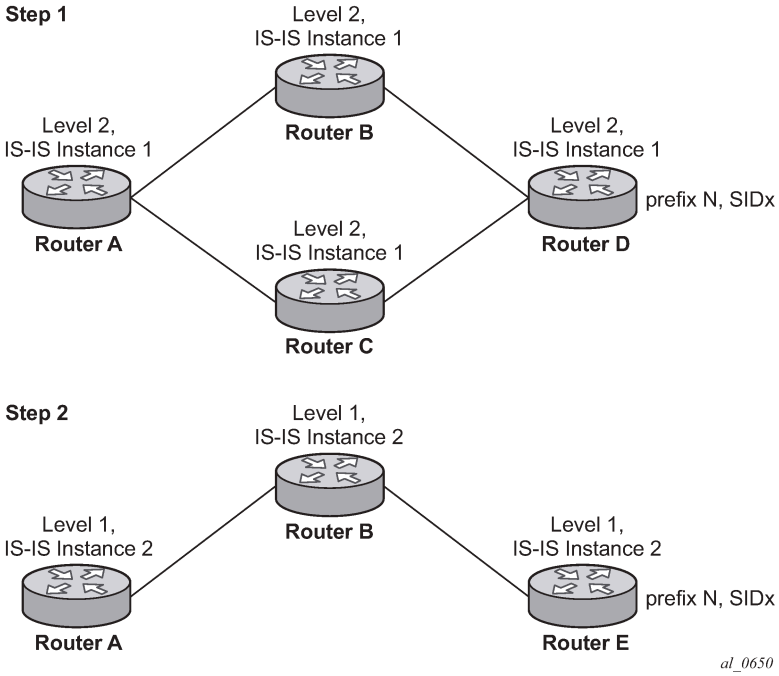
2.1.6.2.5 Programming ILM/NHLFE for the same prefix across IGP instances

In global SID index range mode of operation, the resulting ILM label value is the same across the IGP instances. The router programs ILM/NHLFE for the prefix of the winning IGP instance based on the RTM route-type preference. The router logs a trap and generates a syslog error message, and does not program the other prefix SIDs in the datapath.

In the per-instance SID index range mode of operation, the resulting ILM label has different values across the IGP instances. The router programs ILM/NHLFE for each prefix as expected.

The following figure shows an IS-IS example of handling in case of a global SID index range.

Figure 3: Handling of the same prefix and SID in different IS-IS instances



Assume the following route-type preference in RTM and tunnel-type preference in TTM are configured:

- ROUTE_PREF_ISIS_L1_INTER (RTM) 15
- ROUTE_PREF_ISIS_L2_INTER (RTM) 18
- ROUTE_PREF_ISIS_TTM 10

Note: The TTM tunnel-type preference is not used by the SR module. It is put in the TTM and is used by other applications, such as VPRN auto-bind and BGP shortcut, to select a TTM tunnel.

1. Router A performs the following resolution within the single level 2, IS-IS instance 1. All metrics are the same and ECMP = 2.
   - For prefix N, the RTM entry is:
     - prefix N
     - nhop1 = B
     - nhop2 = C
     - preference 18
   - For prefix N, the SR tunnel TTM entry is:
     - tunnel-id 1: prefix N-SIDx
     - nhop1 = B
     - nhop2 = C
     - tunl-pref 10
2. Add Level 1, IS-IS instance 2 in the same configuration, but in routers A, B, and E only.

- For prefix N, the RTM entry is:
  - prefix N
  - nhop1 = B
  - preference 15

  The RTM prefers L1 route over L2 route.
- For prefix N, there is one SR tunnel entry for L2 in TTM:
  - tunnel-id 1: prefix N-SIDx
  - nhop1 = B
  - nhop2 = C
  - tunl-pref 10

2.1.6.2.6 Handling ILM resource exhaustion while assigning a SID index/label

If the system exhausted an ILM resource while assigning a SID index/label to a local loopback interface, then index allocation fails and an error is displayed in the CLI. The router logs a trap and generates a syslog error message.

2.1.6.2.7 Handling ILM, NHLFE, or other IOM or CPM resource exhaustion while resolving or programming a SID index/label

If the system exhausted an ILM, NHLFE, or any other IOM or CPM resource while resolving and programming a received prefix SID or programming a local adjacency SID, the following occurs:

1. The IGP instance goes into overload and a trap and syslog error message are generated.
2. The segment routing module deletes the tunnel.

The user must manually clear the IGP overload condition after freeing resources. After the IGP is brought back up, it attempts to program all tunnels that previously failed the programming operation at the next SPF.

2.1.7 Segment routing tunnel management

The segment routing module adds a shortest path SR tunnel entry to TTM for each resolved remote node SID prefix and programs the datapath with the corresponding LTN with the push operation pointing to the primary and LFA backup NHLFEs. The LFA backup next hop for a prefix that was advertised with a node SID is only computed if the **loopfree-alternates** option is enabled in the IS-IS or OSPF instance. The resulting SR tunnel that is populated in TTM is automatically protected with FRR when an LFA backup next hop exists for the prefix of the node SID.

With ECMP, a maximum of 32 primary next-hops (NHLFEs) are programmed for the same tunnel destination for each IGP instance. ECMP and LFA next-hops are mutually exclusive, as in the current implementation.

The default preference for shortest path segment routing tunnels in the TTM is set lower than LDP tunnels but higher than BGP tunnels to allow controlled migration of customers without disrupting their current deployment when they enable segment routing.

The global default TTM preferences for the tunnel types, including the preference of both segment routing tunnels based on shortest path (referred to as SR IS-IS and SR-OSPF) is as follows:

- ROUTE_PREF_RSVP 7
- ROUTE_PREF_SR_TE 8
- ROUTE_PREF_LDP 9
- ROUTE_PREF_OSPF_TTM 10
- ROUTE_PREF_ISIS_TTM 11
- ROUTE_PREF_BGP_TTM 12
- ROUTE_PREF_GRE 255

The default value for SR IS-IS or SR-OSPF is the same, regardless of whether one or more IS-IS or OSPF instances are programming a tunnel for the same prefix. In this case, the router selects an SR tunnel based on the lowest IGP instance ID.

The TTM preference is used in the case of BGP shortcuts, VPRN auto-bind, or BGP transport tunnel when the tunnel binding commands are configured to the **any** value, which parses the TTM for tunnels in the protocol preference order. The user can use the global TTM preference or explicitly list the tunnel types to be used. When the tunnel types are listed explicitly, the TTM preference is still used to select one type over the other. In both cases, a fallback to the next preferred tunnel type is performed if the selected one fails. When a more preferred tunnel type becomes available, the system reverts to that tunnel type.

See BGP shortcuts using segment routing tunnels, BGP labeled route resolution using segment routing tunnels, and Service packet forwarding with segment routing for the detailed service and shortcut binding CLI.

For SR IS-IS and SR-OSPF, the user can configure the preference of each IGP instance in addition to the default values.

```
config>router>isis>segment-routing>tunnel-table-pref preference 1 to 255
config>router>ospf>segment-routing>tunnel-table-pref preference 1 to 255
```

Note:
The preference of SR-TE LSP is not configurable and is the second-most preferred tunnel type after RSVP-TE. The preference of SR-TE LSP is independent of whether the SR-TE LSP was resolved in IS-IS or OSPF.

2.1.7.1 Tunnel MTU determination

The MTU of a segment routing tunnel populated into the TTM is determined in the same way as an IGP tunnel; for example, LDP LSP is based on the outgoing interface MTU minus the label stack size. Segment routing, however, supports remote LFA and TI-LFA, which can program an LFA repair tunnel by adding one or more labels.

To configure the MTU of all segment routing tunnels within each IGP instance, use the following commands:

```
config>router>isis>segment-routing>tunnel-mtu bytes bytes
config>router>ospf>segment-routing>tunnel-mtu bytes bytes
```

There is no default value for this command. If the user does not configure a segment routing tunnel MTU, the MTU, in bytes, is determined by IGP as follows:

SR_Tunnel_MTU = MIN {*Cfg_SR_MTU*, *IGP_Tunnel_MTU* - (1 + frr - overhead) × 4}

Where:

- *Cfg_SR_MTU* is the MTU configured by the user for all segment routing tunnels within an IGP instance using the preceding CLI commands. If no value was configured by the user, the segment routing tunnel MTU is determined by the IGP interface calculation.
- *IGP_Tunnel_MTU* is the minimum of the IS-IS or OSPF interface MTU among all the ECMP paths or among the primary and LFA backup paths of this segment routing tunnel.
- *frr-overhead* is set the following parameters:
  - *value* of ti-lfa **[max-sr-frr-labels** labels**]** if **loopfree-alternates** and **ti-lfa** are enabled in this IGP instance
  - 1 if **loopfree-alternates** and **remote-lfa** are enabled but **ti-lfa** is disabled in this IGP instance
  - otherwise, it is set to 0

The SR tunnel MTU is dynamically updated anytime any of the parameters used in its calculation change. This includes when the set of the tunnel next-hops changes or the user changes the configured SR MTU or interface MTU value.

Note:
The calculated SR tunnel MTU is used to determine an SDP MTU and to check the Layer 2 service MTU. When fragmenting IP packets forwarded in GRT or in a VPRN over an SR shortest path tunnel, the datapath always deducts the worst-case MTU (5 labels or 6 labels if hash label feature is enabled) from the outgoing interface MTU when deciding whether to fragment the packet. In this case, the above formula is not used.

2.1.8 Segment routing local block

Some labels that are provisioned through CLI or a management interface must be allocated from the Segment Routing Local Block (SRLB). The SRLB is a reserved label block configured under **config**>**router**>**mpls-labels**. See the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide* for more information about reserved label blocks.

The label block to use is specified by the **srlb** command under IS-IS or OSPF:

```
config>router>isis>segment-routing
   [no] srlb reserved-label-block-name

config>router>ospf> segment-routing
   [no] srlb reserved-label-block-name
```

Provisioned labels for adjacency SIDs and adjacency SID sets must be allocated from the configured SRLB. The request is rejected if any of the following are true:

- no SRLB is specified
- the requested label does not fall within the SRLB
- the label is already allocated

2.1.8.1 Bundling adjacencies in adjacency sets

An adjacency set is a bundle of adjacencies, represented by a common adjacency SID for the bundled set. It enables, for example, a path for an SR-TE LSP through a network to be specified while allowing the local node to spray packets across the set of links identified by a single adjacency SID.

SR OS supports both parallel adjacency sets (for example, those where adjacencies originating on one node terminate on a second, common node), and the ability to associate multiple interfaces on a specified node, irrespective of whether the far end of the respective links of those interfaces terminate on the same node.

An adjacency set is created under IS-IS or OSPF using the following CLI commands:

```
config
   router
      isis | ospf
         segment-routing
            [no] adjacency-set id
               family [ipv4 | ipv6]
               parallel [no-advertise]
               no parallel
               exit
...
.              exit
            exit
config
   router
      ospf
         segment-routing
            [no] adjacency-set id
               parallel [no-advertise]
               no parallel
               exit
...
.
            exit
```

The **adjacency-set** *id* command specifies an adjacency set, where *id* is an unsigned integer from 0 to 4294967295.

In IS-IS, each adjacency set is assigned an address family, IPv4 or IPv6. The family command for IS-IS indicates the address family of the adjacency set. For OSPF, the address family of the adjacency set is implied by the OSPF version and the instance.

The **parallel** command indicates that all members of the adjacency set must terminate on the same neighboring node. When the **parallel** command is configured, the system generates a trap message if a user attempts to add an adjacency terminating on a neighboring node that differs from the existing members of the adjacency set. See Associating an interface with an adjacency set for details about how to add interfaces to an adjacency set. The system stops advertising the adjacency set and deprograms it from TTM. The **parallel** command is enabled by default.

By default, parallel adjacency sets are advertised in the IGP. The **no-advertise** option prevents a parallel adjacency set from being advertised in the IGP; it is only advertised if the **parallel** command is configured. To prevent issues in the case of ECMP if a non-parallel adjacency set is used, an external controller may be needed to coordinate the label sets for SIDs at all downstream nodes. As a result, non-parallel adjacency sets are not advertised in the IGP. The label stack below the adjacency set label must be valid at any downstream node that exposes it, even though it is sprayed over multiple downstream next-hops.

Parallel adjacency sets are programmed in TTM (unless there is an erroneous configuration of a non-parallel adjacency). Non-parallel adjacency sets are not added to TTM or RTM, meaning they cannot be used as a hop at the originating node. Parallel adjacency sets that are advertised are included in the link-state database and TE database, but non-parallel adjacency sets are not included because they are not advertised.

An adjacency set with only one next hop is also advertised as an individual adjacency SID with the S flag set. However, the system does not calculate a backup for an adjacency set even if it has only one next hop.

2.1.8.1.1 Associating an interface with an adjacency set

IS-IS or OSPF interfaces are associated with one or more adjacency sets using the following CLI commands. Both numbered and unnumbered interfaces can be assigned to the same adjacency set.

```
config
   router
      isis
         interface
           [no] adjacency-set id
           [no] adjacency-set id
           [no] adjacency-set id
config
   router
      ospf
         area
            interface
               [no] adjacency-set id
               [no] adjacency-set id
               [no] adjacency-set id
```

If an interface is assigned to an adjacency set, then a common adjacency SID value is advertised for every interface in the set, in addition to the adjacency SID corresponding to the IPv4 and or IPv6 adjacency for the interface. Each IS-IS or OSPF advertisement therefore contains two adjacency SID TLVs for an address family:

- an adjacency SID for the interface (a locally-unique value)
- an adjacency SID TLV for the adjacency set

  This TLV is distinguished by having the S-bit (IS-IS) or G-bit (OSPF) in the flags field set to 1. Its value is the same as other adjacency SIDs in the set at that node.

By default, both the adjacency SID for an interface and the adjacency SID for a set are dynamically allocated by the system. However, it is possible for the user to configure an alternate, static value for the SID; see Provisioning adjacency SID values for an adjacency set for more information.

A maximum of 32 interfaces can be bound to a common adjacency set. Configuring more than 32 interfaces is blocked by the system and a CLI error is generated.

Only point-to-point interfaces can be assigned to an adjacency set.

If a user attempts to assign an IES interface to an adjacency set, the system generates a CLI warning and segment routing does not program the association.

The IGP blocks the configuration of an adjacency set under an interface when the adjacency set has not yet been created under segment-routing.

In IS-IS, it is possible to add Layer 1, Layer 2, or a mix of Layer 1 and Layer 2 adjacencies to the same adjacency set.

2.1.8.1.2 Provisioning adjacency SID values for an adjacency set

For an adjacency set, static values are configured using the **sid** CLI command, as follows:

```
config>router>isis>segment-routing
      [no] adjacency-set id
         family [ipv4 | ipv6]
         [no] sid label value
         parallel [no-advertise]
         no parallel
         exit
      [no] adjacency-set id
          family [ipv4 | ipv6]
          [no] sid label value
          parallel [no-advertise]
          no parallel
          exit
                      ...

config>router>ospf>segment-routing
      [no] adjacency-set id
         [no] sid label value
         parallel [no-advertise]
         no parallel
         exit
      [no] adjacency-set id
         [no] sid label value
         parallel [no-advertise]
         no parallel
         exit
       ...
```

If **no sid** is configured, a dynamic value is allocated to the adjacency set. A user may change the dynamic value to specify a static SID value. Changing an adjacency set value from dynamic to static, or static to dynamic, may result in traffic being dropped as the ILM is reprogrammed.

The *value* must correspond to a label in the reserved label block in provisioned mode referred to by the **srlb** command. A CLI error is generated if a user attempts to configure an invalid *value*. If a label is not configured, then the label *value* is dynamically allocated by the system from the dynamic labels range. If a static adjacency set label is configured, then the system does not advertise a dynamic adjacency set label.

A static label value for an adjacency set SID is persistent. Therefore, the P-bit of the flags field in the Adjacency-SID TLV, referring to the adjacency set must be set to 1.

2.1.9 Loop-free alternates

This section describes LFA implementation and configuration.

2.1.9.1 Remote LFA with segment routing

The user enables the remote LFA next-hop calculation by the IGP LFA SPF by configuring the **remote-lfa** option in the command that enables LFA calculation:

```
config>router>isis>loopfree-alternates remote-lfa
config>router>ospf>loopfree-alternates remote-lfa
```

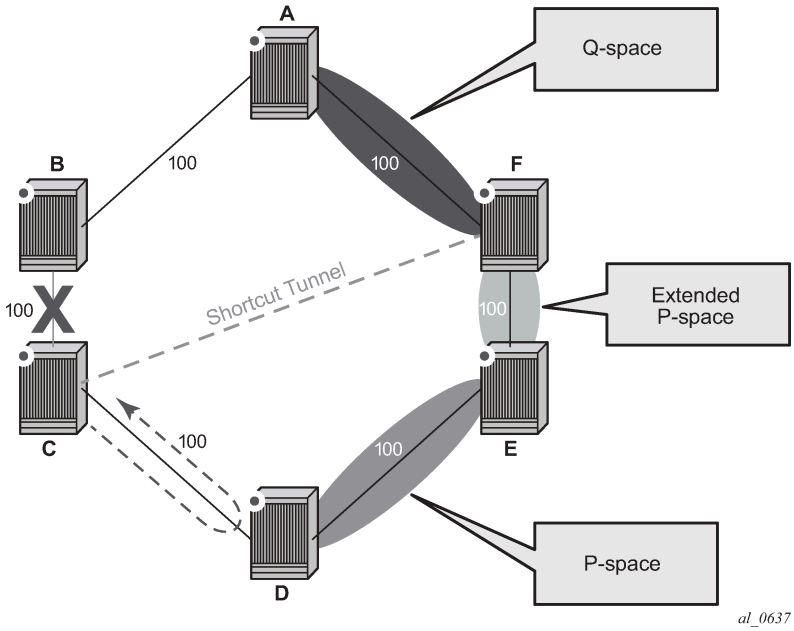
SPF performs the additional remote LFA computation following the regular LFA next-hop calculation when both of the following conditions are met.

- The **remote-lfa** option is enabled in an IGP instance.
- The LFA next-hop calculation did not result in protection for one or more prefixes resolved to a specific interface.

Remote LFA extends the protection coverage of LFA-FRR to any topology by automatically computing and establishing or tearing down shortcut tunnels or repair tunnels, to a remote LFA node, which puts the packets back into the shortest path without looping them back to the node that forwarded them over the repair tunnel. A repair tunnel can, in theory, be an RSVP LSP, an LDP-in-LDP tunnel, or an SR tunnel. In SR OS, this feature is restricted to use an SR repair tunnel to the remote LFA node.

The remote LFA algorithm for link protection is described in RFC 7490, *Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)*. Unlike a typical LFA calculation, which is calculated per prefix, the LFA algorithm for link protection is a per-link LFA SPF calculation. The algorithm provides protection for all destination prefixes that share the protected link by using the neighbor on the other side of the protected link as a proxy for all of these destinations. Assume the topology in the following figure.

Figure 4: Example of remote LFA topology



When the LFA SPF in node C computes the per-prefix LFA next hop, prefixes that use link C-B as the primary next hop have no LFA next hop because of the ring topology. If node C used node link C-D as a

backup next hop, node D would loop a packet back to node C. The remote LFA then runs the following PQ algorithm, per RFC 7490.

1. Compute the extended P space of Node C with respect to link C-B. The extended P space is the set of nodes reachable from node C without any path transiting the protected link (link C-B). This computation yields nodes D, E, and F.

   The determination of the extended P space by node C uses the same computation as the regular LFA by running SPF on behalf of each of the neighbors of C.

   Note:
   RFC 7490 introduced the concept of P space, which would have excluded node F because, from the node C perspective, node C has a couple of ECMP paths, one of which goes via link C-B. However, because the remote LFA next hop is activated when link C-B fails, this rule can be relaxed and node F can be included, which then yields the extended P space.

   The user can limit the search for candidate P nodes to reduce the number of SPF calculations in topologies where many eligible P nodes can exist. The following commands can be used to configure the maximum IGP cost from node C for a P node to be eligible:

   - **config>router>isis>loopfree-alternates remote-lfa max-pq-cost** *value*
   - **config>router>ospf>loopfree-alternates remote-lfa max-pq-cost** *value*

2. Compute the Q space of node B with respect to link C-B: the set of nodes from which the destination proxy (node B) can be reached without any path transiting the protected link (link C-B).

   The Q space calculation is effectively a reverse SPF of node B. In general, one reverse SPF is run on behalf of each neighbor of C to protect all destinations resolving over the link to the neighbor. This yields nodes F and A in the example of Figure 4: Example of remote LFA topology.

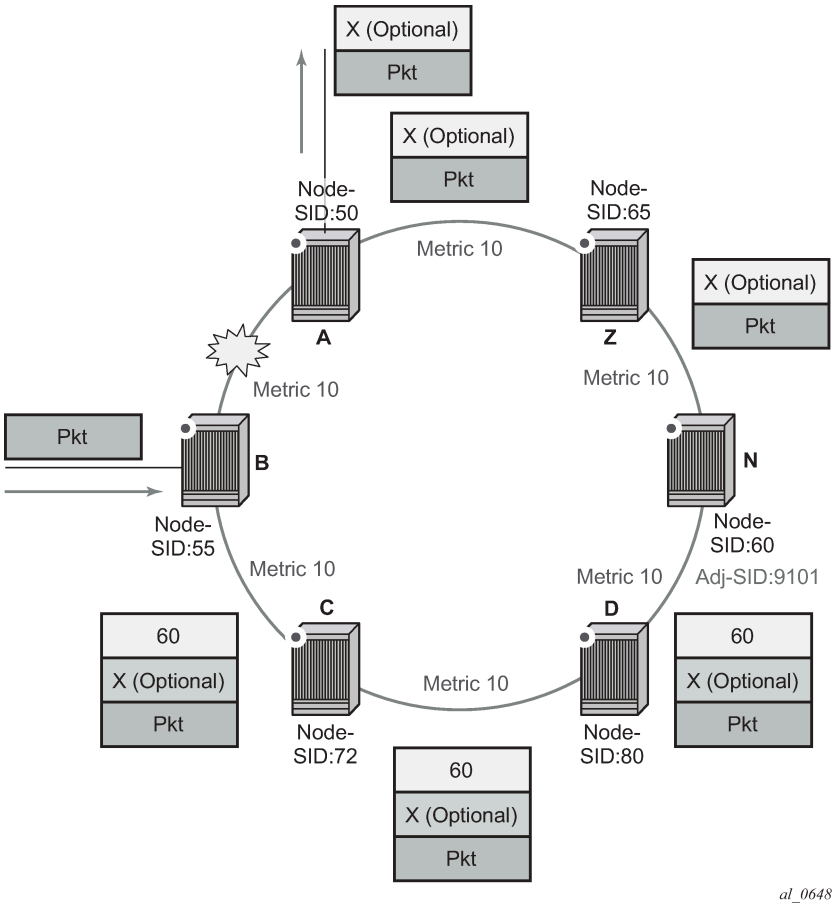
   The user can limit the search for candidate Q nodes to reduce the number of SPF calculations in topologies where many eligible Q nodes can exist. The CLI commands in step 1 are also used to configure the maximum IGP cost from node C for a Q node to be eligible.

3. Select the best alternate node: this is the intersection of extended P and Q spaces. The best alternate node or PQ node is node F in the example of Figure 4: Example of remote LFA topology. From node F onwards, traffic follows the IGP shortest path.

   If many PQ nodes exist, the lowest IGP cost from node C is used to narrow down the selection, and if more than one PQ node remains, the node with lowest router ID is selected.

The details of the label stack encoding when the packet is forwarded over the remote LFA next hop are shown in the following figure.

Figure 5: Remote LFA next hop in segment routing



The label corresponding to the node SID of the PQ node is pushed on top of the original label of the SID of the resolved destination prefix. If node C has resolved multiple node SIDs corresponding to different prefixes of the selected PQ node, it pushes the lowest node SID label on the packet when forwarded over the remote LFA backup next hop.

If the PQ node is also the advertising router for the resolved prefix, the label stack is compressed in the following cases, depending on the IGP:

- In IS-IS, the label stack is always reduced to a single label, which is the label of the resolved prefix owned by the PQ node.
- In OSPF, the label stack is reduced to the single label of the resolved prefix when the PQ node advertised a single node SID in this OSPF instance. If the PQ node advertised a node SID for multiple loopback interfaces within this same OSPF instance, the label stack is reduced to a single label only if the SID of the resolved prefix is the lowest SID value.

The following rules and limitations apply to the remote LFA implementation:

- If the user excludes a network IP interface from being used as an LFA next-hop using the **loopfree-alternate-exclude** command under the IS-IS or OSPF context of the interface, the interface is also excluded from being used as the outgoing interface for a remote LFA tunnel next hop.
- As with the regular LFA algorithm, the remote LFA algorithm computes a backup next hop to the ABR advertising an inter-area prefix and not to the destination prefix.

2.1.9.2 Topology-independent LFA

The Topology-Independent LFA (TI-LFA) feature improves the protection coverage of a network topology by computing and automatically instantiating a repair tunnel to a Q-node that is not in the shortest path from the computing node. The repair tunnel uses the shortest path to the P-node and a source-routed path from the P-node to the Q-node.

In addition, the TI-LFA algorithm selects the backup path that matches the post-convergence path. This helps capacity planning in the network because traffic always flows on the same path when transitioning to the FRR next hop and then on to the new primary next hop.

At a high level, the TI-LFA link protection algorithm searches for the closest Q-node to the computing node and then selects the closest P-node to this Q-node, up to a maximum number of labels. This is performed on each of the post-convergence paths to each destination node or prefix D.

When the TI-LFA feature is enabled in IS-IS, it provides a TI-LFA link-protect backup path in IS-IS MT=0 for an SR IS-IS IPv4 or SR IS-IS IPv6 tunnel (node SID and adjacency SID), for an IPv4 SR-TE LSP, and for LDP IPv4 FEC when the LDP **fast-reroute backup-sr-tunnel** option is enabled.

2.1.9.2.1 TI-LFA configuration

Use the following command to enable TI-LFA in an IS-IS instance:

config>router>isis>loopfree-alternates [**remote-lfa** [**max-pq-cost** *value*]] [**ti-lfa** [**max-sr-frr-labels** *value*]]

The **ti-lfa** option in IS-IS provides a TI-LFA link-protect backup path in IS-IS MT=0 for an SR IS-IS IPv4 and IPv6 tunnel (node SID and adjacency SID), for an IPv4 SR-TE LSP, and for an LDP IPv4 FEC when the LDP **fast-reroute backup-sr-tunnel** option is enabled. For more information about the applicability of the various LFA options, see LFA protection option applicability.

The **max-sr-frr-labels** parameter limits the search for the LFA backup next hop based on the value of the label as follows:

- **0**

  The IGP LFA SPF restricts the search to the TI-LFA backup next hop that does not require a repair tunnel, meaning that the P-node and Q-node are the same and match a neighbor. This is also the case when both the P- and Q-nodes match the advertising router for a prefix.

- **1 to 3**

  The IGP LFA SPF widens the search to include a repair tunnel to a P-node, which itself is connected to the Q-nodes with a zero to two hops for a maximum total of three labels: one node SID to the P-node and two adjacency SIDs from the P-node to the Q-node. If the P-node is a neighbor of the computing node, its node SID is compressed, meaning that up to three adjacency SIDs can separate the P- and Q-nodes.

- **2 (default)**

  Corresponds to a repair tunnel to a non-adjacent P that is adjacent to the Q-node. If the P-node is a neighbor of the computing node, the node SID of the P-node is compressed, and the default value of two labels corresponds to two adjacency SIDs between the P- and Q-nodes.

If the user attempts to change the **max-sr-frr-labels** parameter to a value that results in a change to the computed FRR overhead, the IGP checks that all SR-TE LSPs can properly account for the overhead

based on the configuration of the LSP **max-sr-labels** and **additional-frr-labels** parameter values; otherwise, the change is rejected.

The FRR overhead is computed by IGP and its value is set as follows:

- 0 if **segment-routing** is disabled in the IGP instance
- 0 if **segment-routing** is enabled but **remote-lfa** is disabled and **ti-lfa** is disabled
- 1 if **segment routing** is enabled and **remote-lfa** is enabled but **ti-lfa** is disabled, or if **segment-routing** is enabled and **remote-lfa** is enabled and **ti-lfa** is enabled but **ti-lfa max-sr-frr-labels** *labels* is set to 0.
- the value of **ti-lfa max-sr-frr-labels** *labels* if **segment-routing** is enabled and **ti-lfa** is enabled, regardless if **remote-lfa** is enabled or disabled.

2.1.9.2.2 TI-LFA link-protect operation

This section describes TI-LFA protection behavior when the **loopfree-alternates** command is enabled with the **remote-lfa** and **ti-lfa** options, as described in TI-LFA configuration.

2.1.9.2.2.1 LFA protection option applicability

Depending on the configured options of the **loopfree-alternates** command, the LFA SPF in an IGP instance runs the following algorithms in order.

1. The LFA SPF computes a regular LFA for each node and prefix.

   In this step, a computed backup next hop satisfies any applied LFA policy. This backup next hop protects that specific prefix or node in the context of IP FRR, LDP FRR, SR FRR, and SR-TE FRR.

2. The LFA SPF runs the TI-LFA algorithm if the **ti-lfa** option is enabled for all prefixes and nodes, regardless of the outcome of the first step.

   If the LDP **fast-reroute backup-sr-tunnel** option is enabled, a prefix or node for which a TI-LFA backup next hop is found overrides the result from step 1 in the context of LDP FRR in SR FRR and in SR-TE FRR.

   With SR FRR and SR-TE FRR, the TI-LFA next hop protects the node-SID of that prefix and any adjacency SID terminating on the node-SID of that prefix.

   The prefix or node continues to use the backup next hop found either in the context of LDP FRR (if the LDP **fast-reroute backup-sr-tunnel** option is disabled), or in the IP FRR.

3. The LFA SPF runs remote LFA only for the next hop of prefixes and nodes that remain unprotected after step 1 and step 2 if the **remote-lfa** option is enabled.

   A prefix or node for which a remote LFA backup next hop is found uses it in the context of LDP FRR in SR FRR and in SR-TE FRR when the LDP **fast-reroute backup-sr-tunnel** option is enabled.

To protect an adjacency SID, the LFA selection algorithm uses the following preference order:

1. adjacency of an alternate parallel link to the same neighbor.

   If more than one adjacency exists, select one as follows:

   a. adjacency with the lowest metric

   b. adjacency to the neighbor with the lowest router ID (OSPF) or system-id (IS-IS), and the lowest metric

   c. with the lowest interface index and the lowest router ID (OSPF) or system-id (IS-IS)

2. an ECMP next hop to a node-SID of the same neighbor that is different from the next hop of the protected adjacency.

   If more than one next hop exists, select one as follows:

   a. next hop with the lowest metric

   b. next hop to the neighbor with the lowest router ID (OSPF) or system-id (IS-IS) if same lowest metric

   c. next hop to the lowest interface index if same neighbor router ID (OSPF) or system-id (IS-IS)

3. LFA backup outcome of a node SID of the same neighbor. The following is the preference order:

   a. TI-LFA backup

   b. LFA backup

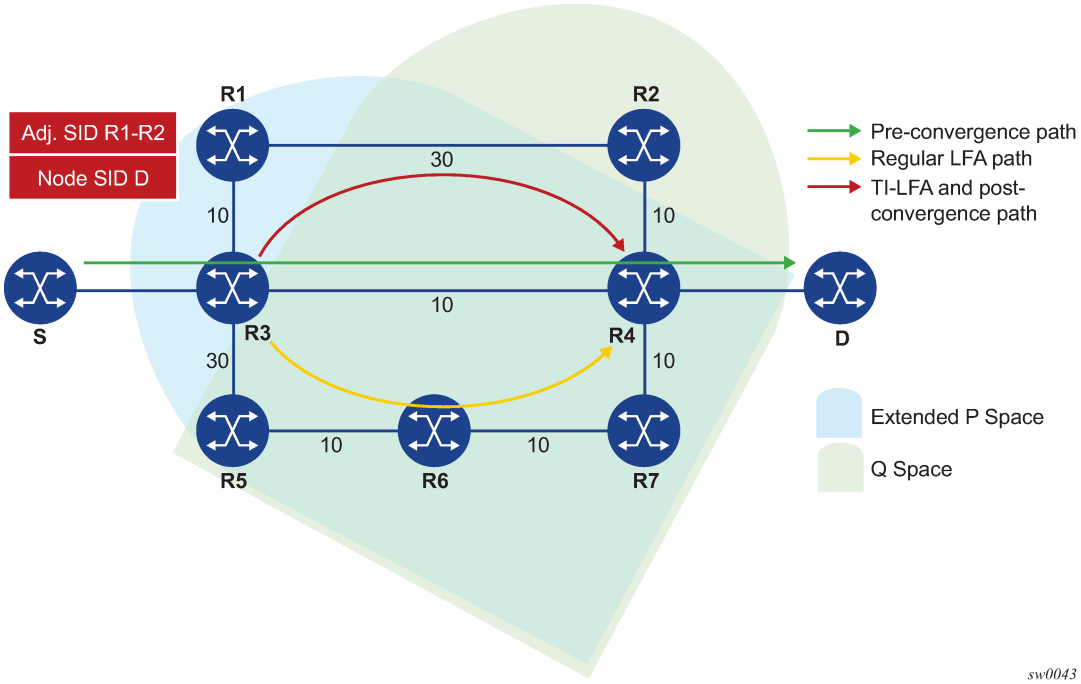
   c. RLFA backup

2.1.9.2.2.2 TI-LFA algorithm

The TI-LFA link protection algorithm searches for the closest Q-node to the computing node and then selects the closest P-node to this Q-node, up to a number of labels corresponding to the **ti-lfa max-sr-frr-labels** *labels* value, on each of the post-convergence paths to each destination node or prefix D.

The following figure shows a topology where router R3 computes a TI-LFA next hop for protecting link R3-R4.

Figure 6: Selecting link-protect TI-LFA backup path



Applying the topology in the preceding figure, router R3 computes the link-protected TI-LFA backup path in the following order.

1. The router computes the post-convergence SPF on the topology without the protected link.

In the preceding figure, R3 finds a single post-convergence path to destination D via R1.

Note:
The post-convergence SPF does not include IGP shortcut.

2. The router computes the extended P-Space of R3 with respect to protected link R3-R4 on the post-convergence paths.

   This is the set of nodes Yi in the post-convergence paths that are reachable from R3 neighbors without any path transiting the protected link R3-R4.

   R3 computes an LFA SPF rooted at each of its neighbors within the post-convergence paths, that is, R1, using the following equation:

   Distance_opt(R1, Yi) < Distance_opt(R1, R3) + Distance_opt(R3, Yi)

   Where, Distance_opt(A,B) is the shortest distance between A and B. The extended P-space calculation yields only node R1.

3. The router computes the Q-space of R3 with respect to protected link R3-R4 in the post-convergence paths.

   This is the set of nodes Zi in the post-convergence paths from which the neighbor node R4 of the protected link, acting as a proxy for all destinations D, can be reached without any path transiting the protected link R3-R4.

   Distance_opt(Zi, R4) < Distance_opt(Zi, R3) + Distance_opt(R3, R4)

   The Q-space calculation yields nodes R2 and R4.

   This is the same computation of the Q-space performed by the remote LFA algorithm, except that the TI-LFA Q-space computation is performed only on the post-convergence paths.

4. For each post-convergence path, the router searches for the closest Q-node and selects the closest P-node to this Q-node, up to the number of labels corresponding to the configured **ti-lfa max-sr-frr-labels** parameter value.

   The topology in Figure 6: Selecting link-protect TI-LFA backup path shows a single post-convergence path, a single P-node (R1), and that R2 is the closest of the two found Q-nodes to the P-Node.

   R3 installs the repair tunnel to the P-Q set and includes the node-SID of R1 and the adjacency SID of the adjacency over link R1-R2 in the label stack. Because the P-node R1 is a neighbor of the computing node R3, the node SID of R1 is not needed and the label stack of the repair tunnel is compressed to the adjacency SID over link R1-R2 as shown in Figure 6: Selecting link-protect TI-LFA backup path.

   When a P-Q set is found on multiple ECMP post-convergence paths, the following selection rules are applied, in ascending order, to select a set from a single path:

   a. the lowest number of labels

   b. the next hop to the neighbor router with the lowest **router-id** (OSPF) or **system-id** (ISIS)

   c. the next hop corresponding to the Q node with the lowest **router-id** (OSPF) or **system-id** (ISIS)

   If multiple links with adjacency SID exist between the selected P-node and the selected Q-node, the following rules are used for link selection:

   a. the adjacency SID with the lowest metric

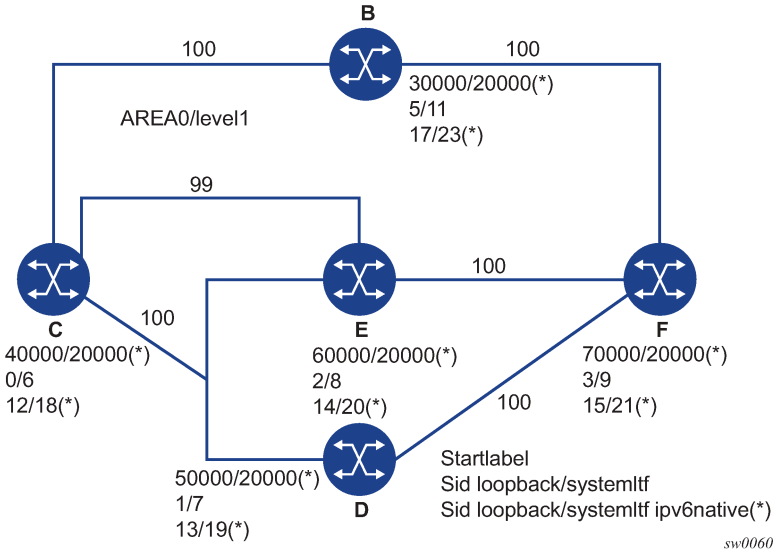
   b. the adjacency SID with the lowest SID value if the lowest metric is the same

2.1.9.2.2.3 TI-LFA feature interaction and limitations

The following are feature interactions and limitations of the TI-LFA link protection:

- Enabling the **ti-lfa** command in an IS-IS or OSPF instance overrides the user configuration of the **loopfree-alternate-exclude** command under the interface context in that IGP instance; that is, the TI-LFA SPF uses that interface as a backup next hop if it matches the post-convergence next hop.
- Any prefix excluded from LFA protection using the **loopfree-alternates exclude prefix-policy** command under the IGP instance context is also excluded from TI-LFA.
- Because the post-convergence SPF does not use paths transiting on a node in IS-IS overload, the TI-LFA backup path automatically does not transit on such a node.
- IES interfaces are skipped in TI-LFA computation because they do not support segment routing with MPLS encapsulation. If the only found TI-LFA backup next hop matches an IES interface, IGP treats this as if there were no TI-LFA backup paths and falls back to using either a remote LFA or regular LFA backup path in accordance with the selection rules described in LFA protection option applicability.
- The TI-LFA feature provides link-protection only. Therefore, if the protected link is a broadcast interface, the TI-LFA algorithm only guarantees protection of that link and not of the pseudonode (PN) corresponding to that shared subnet. That is, if the PN is in the post-convergence path, the TI-LFA backup path may still traverse again the PN. For example, node E in Figure 7: TI-LFA backup path via a pseudo-node computes a TI-LFA backup path to destination D via E-C-PN-D because it is the post-convergence path when excluding link E-PN from the topology. This TI-LFA backup does not protect against the failure of the PN.

Figure 7: TI-LFA backup path via a pseudo-node

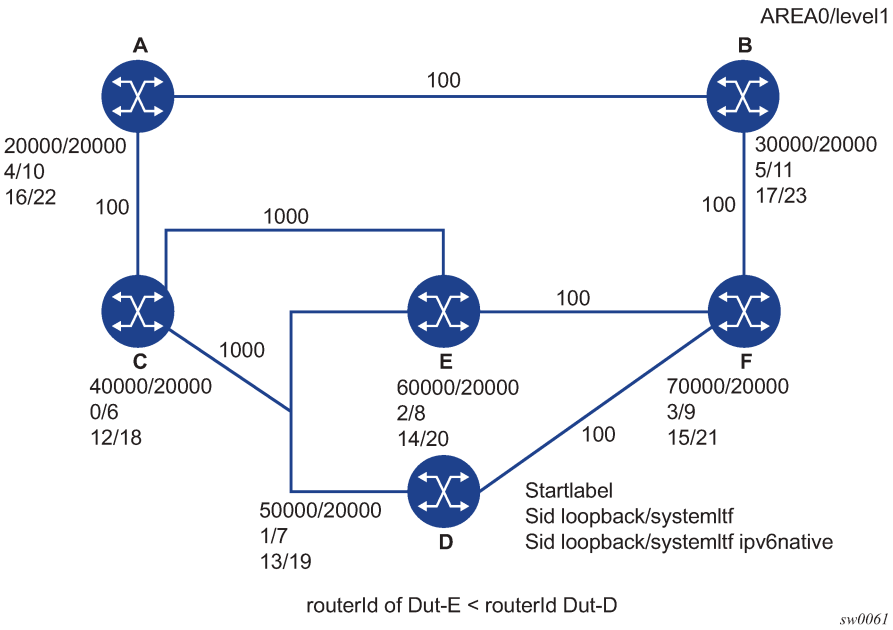


- When the computing router selects an adjacency SID among a set of parallel adjacencies between the P- and Q-nodes, the selection rules in step 4 of TI-LFA algorithm. However, these rules may not yield the same interface the P-node would have selected in its post-convergence SPF, because the latter is based on the lowest value of the locally managed interface index.

  For example, node A in Figure 8: Parallel adjacencies between P and Q nodes computes the link-protect TI-LFA backup path for destination node E as path A-C-E, where C is the P-node and E is

the Q-node and destination. Node C has a pair of adjacency SIDs with the same metric to E. Node A selects the adjacency over the P2P link C-E because it has the lowest SID value, but node C may select the interface C-PN in its post-convergence path calculation, if that interface has a lower interface index than P2P link C-E.

Figure 8: Parallel adjacencies between P and Q nodes



- When a node SID is advertised by multiple routers (anycast SID), the TI-LFA algorithm on a router that resolves the prefix of this SID computes the backup next hop toward a single node owner of the prefix, based on the rules for prefix and SID ECMP next-hop selection.

2.1.9.2.3 Datapath support

The TI-LFA repair tunnel can have a maximum of three additional labels pushed in addition to the label of the destination node or prefix. The user can set a lower maximum value for the additional FRR labels by configuring the **ti-lfa max-sr-frr-labels** *labels* CLI option. The default value is 2.

The datapath models the backup path like a SR-TE LSP and therefore uses a super-NHLFE pointing to the NHLFE of the first hop in the repair tunnel. That first hop corresponds to either an adjacency SID or a node SID of the P node.

There is the special case where the P node is adjacent to the node computing the TI-LFA backup, and the Q node is the same as the P node or adjacent to the P node. In this case, the datapath at the computing router pushes either zero labels or one label for the adjacency SID between the P and Q nodes. The backup path uses a regular NHLFE in this case, as for base LFA or remote LFA features. Figure 6: Selecting link-protect TI-LFA backup path shows an example of a single label in the backup NHLFE.

2.1.9.3 Node protection support in TI-LFA and remote LFA

This feature extends the remote LFA and TI-LFA features by adding support for node protection. The extensions are additions to the original link-protect LFA SPF algorithm.

When node protection is enabled, the router prefers a node-protect over a link-protect repair tunnel for a prefix if both are found in the remote LFA or TI-LFA SPF computations. This feature protects against the failure of a downstream node in the path of the prefix of a node SID except for the node owner of the node SID.

2.1.9.3.1 Node protection in TI-LFA and remote LFA configuration

Use the following CLI commands to configure the remote LFA and TI-LFA node protection feature.

```
configure
        – router
            – [no] isis
                – [no] loopfree-alternates
                    – [no] remote-lfa [max-pq-cost 0 to 4294967295, default=4261412864]
                        – [no] node-protect [max-pq-nodes 1 to 32, default=16]
                    – [no] ti-lfa [max-sr-frr-labels 0 to 3, default=2]
                        – [no] node-protect
                    – exclude
                        – [no] prefix-policy prefix-policy [prefix-policy…(up to 5 max)]
                    – exit
                – exit
```

The CLI commands enable the node-protect calculation to both Remote LFA (**node-protect** [**max-pq-nodes** <*1 to 32*, default=16>]) and TI-LFA (**node-protect**).

If the **node-protect** command is enabled, the router prefers a node-protect over a link-protect repair tunnel for a prefix if both are found in the Remote LFA or TI-LFA SPF computations. The SPF computations may only find a link-protect repair tunnel for prefixes owned by the protected node.

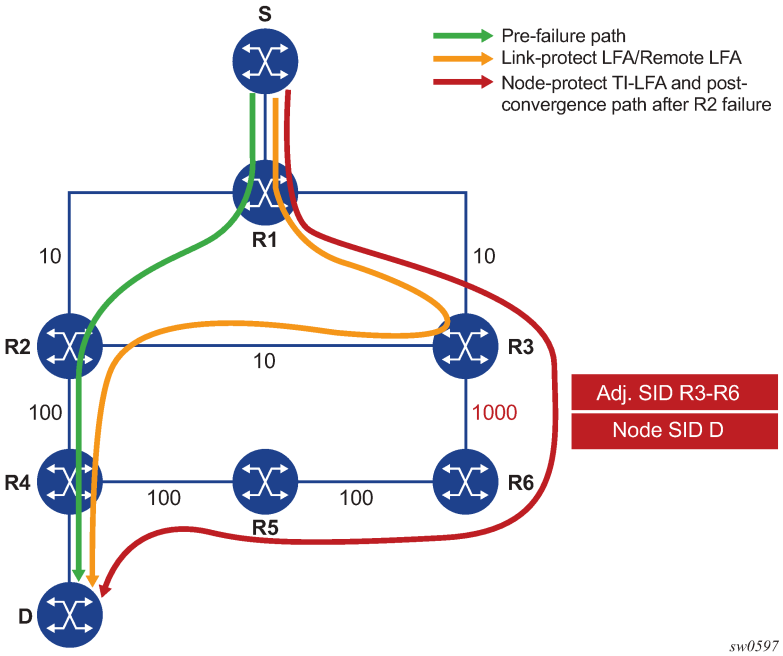
The **max-pq-nodes** parameter in the **remote-lfa** command controls the maximum number of candidate PQ-nodes found in the LFA SPFs for which the node protection check is performed. In the node protection condition, the router must run the original link-protect remote LFA algorithm plus one extra forward SPF on behalf of each PQ-node found, potentially after applying the **max-pq-cost** parameter. This checks the path from the PQ-node to the destination to ensure that the path does not traverse the protected node. Setting the **max-pq-nodes** parameter to a lower value means the LFA SPFs use less computation time and fewer resources, but may result in not finding a node-protect repair tunnel. The default value is 16. For more information, see Remote LFA node-protect operation.

2.1.9.3.2 TI-LFA node-protect operation

SR OS supports the node-protect extensions to the TI-LFA algorithm as described in *draft-ietf-rtgwg-segment-routing-ti-lfa-01*.

The following figure shows a simple topology to illustrate the node-protect operation as described in TI-LFA algorithm.

Figure 9: Application of the TI-LFA algorithm for node protection



The main change as a result of the node-protect extension is that the algorithm protects a node instead of a link.

Using the topology in the preceding figure, the node protection computation is performed in the following sequence:

1. The router computes the post-convergence SPF on the topology without the protected node.
   In Figure 9: Application of the TI-LFA algorithm for node protection, R1 computes TI-LFA on the topology without the protected node R2 and finds a single post-convergence path to destination D via R3 and R6.

   Prefixes owned by all other nodes in the topology have a post-convergence path via R3 and R6 except for prefixes owned by node R2. The latter uses the link R3-R2 and they can only benefit from link protection.

2. The router computes the extended P-Space of R1 with respect to protected node R2 on the post-convergence paths.
   This is the set of nodes Yi in the post-convergence paths that are reachable from R1 neighbors, other than protected node R2, without any path transiting the protected node R2.

   R1 computes an LFA SPF rooted at each of its neighbors within the post-convergence paths. For example, R1 uses using the following calculation to compute the LFA SPF for R3:

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

   Where:

   Distance_opt(A,B) is the shortest distance between A and B.

   The extended P-space calculation yields node R3 only.

3. The router computes the Q-space of R1 with respect to protected link R1-R2 on the post-convergence paths.
   This is the set of nodes Zi in the post-convergence paths from which node R2 can be reached without any path transiting the protected link R1-R2, using the following equation:

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

   The reverse SPF for the Q-space calculation is the same as in the link-protect algorithm and uses the protected node R2 as the proxy for all destination prefixes. To compute the Q space with respect to the protected node R2 instead of link R1-R2, a reverse SPF would have to be performed for each destination D which is very costly and not scalable. However, this means the path from the Q-node to the destination D or the path from the P-node to the Q-node is not guaranteed to avoid the protected node R2. The intersection of the Q-space with post-convergence path is modified in the next step to mitigate this risk.

   This step yields nodes R3, R4, R5, and R6.

4. For each post-convergence path, the router searches for the closest Q-node to destination D and selects the closest P-node to this Q-node, up to the number of labels corresponding to the configured **ti-lfa max-sr-frr-labels** parameter value.
   This step yields the following P-Q sets, depending on the value of the **max-sr-frr-labels** parameter:

   - **max-sr-frr-labels=0**

     R3 is the closest Q-node to the destination D and R3 is the only P-node. This case results in link protection via PQ-node R3.

   - **max-sr-frr-labels=1**

     R6 is the closest Q-node to the destination D and R3 is the only P-node. The repair tunnel for this case uses the SID of the adjacency over link R3-R6 and is shown in Figure 9: Application of the TI-LFA algorithm for node protection.

   - **max-sr-frr-labels=2**

     R5 is the closest Q-node to the destination D and R3 is the only P-node. The repair tunnel for this case uses the SIDs of the adjacencies over links R3-R6 and R6-R5.

   - **max-sr-frr-labels=3**

     R4 is the closest Q-node to the destination D and R3 is the only P-node. The repair tunnel for this case uses the SIDs of the adjacencies over links R3-R6, R6-R5, and R5-R4.

   This step of the algorithm is modified from link protection, which prefers Q-nodes that are the closest to the computing router R1. This is to minimize the probability that the path from the Q-node to the destination D, or the path from the P-node to the Q-node, goes via the protected node R2 as described in step 2. However, there is still a probability that the found P-Q set achieves link protection only.

5. Select the P-Q Set.
   If a candidate P-Q set is found on each of the multiple ECMP post-convergence paths in step 4, the following selection rules are applied in ascending order to select a single set:

   a. the lowest number of labels

   b. the lowest next-hop router ID

   c. the lowest interface index if the same as the next-hop router ID

   If multiple parallel links with adjacency SID exist between the P- and Q-nodes of the selected P-Q set, the following rules are used to select one of them:

a. the adjacency SID with lowest metric

b. the adjacency SID with the lowest SID value, if the same as the lowest metric

For each destination prefix D, R1 programs the TI-LFA repair tunnel (**max-sr-frr-labels**=*1*):

- For prefixes other than those owned by node R2 and R3, R1 programs a node-protect repair tunnel to the P-Q pair R3-R6 by pushing the SID of adjacency R3-R6 on top of the SID for destination D and programming a next hop of R3.
- For prefixes owned by node R2, R1 runs the link-protect TI-LFA algorithm and programs a simple link-protect repair tunnel, which consists of a backup next hop of R3 and pushes no additional label on top of the SID for the destination prefix.
- Prefixes owned by node R3 are not impacted by the failure of R2 because their primary next hop is R3.

2.1.9.3.3 Remote LFA node-protect operation

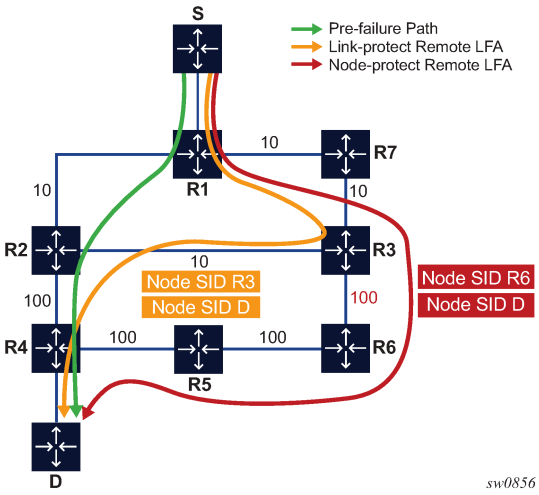
SR OS supports the node-protect extensions to the Remote LFA algorithm, as described in RFC 8102.

Remote LFA follows a similar algorithm as TI-LFA but does not limit the scope of the calculation of the extended P-Space and of the Q-Space to the post-convergence paths.

Remote LFA adds an extra forward SPF on behalf of the PQ node to ensure that, for each destination, the selected PQ-node does not use a path via the protected node.

The following figure shows a slightly modified topology from that in TI-LFA feature interaction and limitations. A new node R7 is added to the top ring and the metric for link R3-R6 is modified to 100.

Figure 10: Application of the remote LFA algorithm for node protection



Using the topology in the preceding figure, the node-protect remote LFA algorithm computation is performed in the following sequence:

1. Compute extended P-Space of R1 with respect to protected node R2.

   This is the set of nodes Yi which are reachable from R1 neighbors, other than protected node R2, without any path transiting the protected node R2.

R1 computes a LFA SPF rooted at each of its neighbors, in this case, R7, using the following equation:

Distance_opt(R7, Yi) < Distance_opt(R7, R2) + Distance_opt(R2, Yi)

Where Distance_opt(A,B) is the shortest distance between A and B.

Nodes R7, R3 and R6 satisfy this inequality.

2. Compute Q-space of R1 with respect to protected link R1-R2.

   This is the set of nodes Zi from which node R2 can be reached without any path transiting the protected link R1-R2, using the following equation.

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

   The reverse SPF for the Q-space calculation is the same as in the remote LFA link-protect algorithm and uses the protected node R2 as the proxy for all destination prefixes.

   This step yields nodes R3, R4, R5, and R6.

   Therefore, the candidate PQ nodes after this step are nodes R3 and R6.

3. For each PQ node found, run a forward SPF to each destination D.

   This step is required to select only the subset of PQ-nodes that do not traverse protected node R2.

   Distance_opt(PQi, D) < Distance_opt(PQi, R2) + Distance_opt(R2, D)

   Of the candidates PQ nodes R3 and R6, only PQ node R6 satisfies this inequality.

   This step of the algorithm is applied to the subset of candidate PQ-nodes out of steps 1 and 2 and to which the **max-pq-cost** parameter was already applied. This subset is further reduced in this step by retaining the candidate PQ-nodes that provide the highest coverage among all protected nodes in the topology, and the number of which does not exceed the value of the **max-pq-nodes** parameter.

   In case of multiple candidate PQ nodes out of this step, the detailed selection rules of a single PQ-node from the candidate list is provided in step 4.

4. Select a PQ-Node.

   If multiple PQ nodes satisfy the criteria in all the above steps, then R1 further selects the PQ node as follows.

   a. R1 selects the lowest IGP cost from R1.

   b. If more than one PQ-nodes remains, R1 selects the PQ-node reachable via the neighbor with the lowest router ID (OSPF) or system ID (IS-IS).

   c. If more than one PQ-node remains, R1 selects the PQ node with the lowest router ID (OSPF) or system ID (IS-IS).

   For each destination prefix D, R1 programs the remote LFA backup path as follows:

   - For prefixes of R5, R4 or downstream of R4, R1 programs a node-protect remote LFA repair tunnel to the PQ node R6 by pushing the SID of node R6 on top of the SID for destination D and programming a next hop of R7.
   - For prefixes owned by node R2, R1 runs the link-protect remote LFA algorithm and programs a simple link-protect repair tunnel which consists of a backup next hop of R7 and pushing the SID of PQ node R3 on top of the SID for the destination prefix D.

- Prefixes owned by nodes R7, R3, and R6 are not impacted by the failure of R2 because their primary next hop is R7.

2.1.9.3.4 TI-LFA and remote LFA node protection feature interaction and limitations

The order of activation of the various LFA types on a per prefix basis is as follows: TI-LFA, followed by base LFA, followed by remote LFA. See LFA protection option applicability for more information about the order of activation.

Node protection is enabled for TI-LFA and remote LFA separately. The base LFA prefers node protection over link protection.

The order of activation of the LFA types supersedes the protection type (node versus link). Consequently, a prefix can be programmed with a link-protect backup next hop by the more preferred LFA type. For example, a prefix is programmed with the only link-protect backup next hop found by the base LFA when a node-protect remote LFA next hop exists.

2.1.9.4 LFA policies

This section describes the application of LFA policies.

2.1.9.4.1 Application of LFA policy to a segment routing node SID tunnel

When a route next-hop policy template is applied to an interface, the LFA backup selection algorithm is extended to also apply to IPv4/IPv6 SR IS-IS, and IPv4 SR-OSPF node-SID tunnels in which a primary next hop is reachable using that interface. The extension applies to the following LFA methods: base LFA, remote LFA (RLFA), and Topology-Independent LFA (TI-LFA).

The following general rules apply across all LFA methods:

- The LFA policy constraints admin-group (**include-group** and **exclude-group**) and SRLG (**srlg-enable**) are only checked against the outgoing interface used by the LFA, RLFA, or TI-LFA backup path.
- The LFA policy parameter **protection-type** {**link** | **node**}, which controls the preference among link and node protection backup types, applies to all LFA methods.

  The base LFA automatically computes both protection backup path types but, on a prefix basis, by default prefers to enforce the node-protect over the link-protect backup next hop.

  By default, RLFA and TI-LFA compute only the link-protect backup path, unless the optional command **node-protect** is enabled, in which case, the preference is reversed.

  For all three LFA methods, when the LFA policy enables a preference for link-protect or node-protect, the backup path is selected from the computed paths based on the configuration for the individual LFA method protection preference and the outcome (node-protect or link-protect) of the actual computation within each method. However, on a per-destination prefix basis, the post-convergence constraint of TI-LFA is selected over the LFA protection type in all cases. The selection rule uses the TI-LFA backup (if one exists), even if it is of a less-preferred protection type than the backup path computed by base LFA and RLFA.

  For example, assume that an LFA policy with **protection-type=node** is applied to an IS-IS interface and the **node-protect** command is enabled in both RLFA and TI-LFA contexts in this IS-IS instance.

If TI-LFA found a link-protect backup path for the destination prefix of a SR IS-IS tunnel, it is always selected over the base LFA node-protect and RLFA node-protect backup paths.

The outcomes of LFA policy selections for specified destination prefixes of SR tunnels are described in the following tables.

Table 2: Outcome of LFA policy with protection-type=node

RLFA outcome	LFA policy protection-type=node								
	Base LFA outcome								
	none			link-protect			node-protect		
	TI-LFA outcome			TI-LFA outcome			TI-LFA outcome		
	none	link-protect	node-protect	none	link-protect	node-protect	none	link-protect	node-protect
none	none	TI-LFA	TI-LFA	LFA	TI-LFA	TI-LFA	LFA	TI-LFA	TI-LFA
link-protect	RLFA	TI-LFA	TI-LFA	LFA	TI-LFA	TI-LFA	LFA	TI-LFA	TI-LFA
node-protect	RLFA	TI-LFA	TI-LFA	RLFA	TI-LFA	TI-LFA	LFA	TI-LFA	TI-LFA

Table 3: Outcome of LFA policy with protection-type=link

RLFA Outcome	LFA policy protection-type=link								
	Base LFA outcome								
	none			link-protect			node-protect		
	TI-LFA outcome			TI-LFA outcome			TI-LFA outcome		
	none	link-protect	node-protect	none	link-protect	node-protect	none	link-protect	node-protect
none	none	TI-LFA	TI-LFA	LFA	TI-LFA	TI-LFA	LFA	TI-LFA	TI-LFA
link-protect	RLFA	TI-LFA	TI-LFA	LFA	TI-LFA	TI-LFA	LFA	TI-LFA	TI-LFA
node-protect	RLFA	TI-LFA	TI-LFA	LFA	TI-LFA	TI-LFA	LFA	TI-LFA	TI-LFA

- LFA policy parameter **nh-type** {**ip** | **tunnel**}, which controls preference among the backup of type IP and type tunnel (IGP shortcut), is not applicable to RLFA and TI-LFA backup paths.

  However, the parameter applies if the LFA policy results in selecting a base LFA backup and the user-enabled resolution of SR-ISIS or SR-OSPF tunnel over IGP shortcut using RSVP-TE LSP.

- When configured on an interface, the route next-hop policy template applies to destination prefixes of the following types:
  - IPv4 and IPv6 SR IS-IS node SID tunnels
  - IPv4 SR-OSPF node SID tunnels
  - IPv4 tunnels where the primary next hop is reachable using that interface

  The route next-hop policy template also indirectly applies to the following:

  - IPv4 or IPv6 SR-TE LSPs
  - IPv4 or IPv6 SR policies that use any of the previously mentioned SR tunnels as the top SID in their SID list

  Finally, the LFA policy indirectly applies to IPv4 LDP FECs when the LDP **fast-reroute backup-sr-tunnel** command is enabled and the FEC is protected using an SR tunnel.
- An LFA policy applied to an interface cannot be selectively enabled or disabled per LFA method.
- As a result of these rules, no more than one backup path remains in each LFA method. In this case, the selection preference order is as follows:
  1. TI-LFA backup IP next hop or repair tunnel
  2. base LFA backup next hop

     This can be of type IP (default or if **nh-type** type preference set to **ip**) or of type tunnel (**nh-type** type preference is set to **tunnel** and family SRv4 or SRv6 resolves to IGP shortcut using RSVP-TE LSP).
  3. remote LFA repair tunnel

2.1.9.4.1.1 Modifying the remote LFA selection algorithm

This section provides detailed steps for modifying the RLFA selection algorithm. The admin-group and SLRG constraints are applied to the neighbors [Ni] prior to the computation of the candidate PQ nodes.

The candidate PQ computations are run for both link protection (link S-E removed) and node protection (node E removed) because there is a need to fall back to the less preferred protection option in accordance with the value of the **protection-type** parameter in the LFA policy applied to a prefix.

Perform the following steps to modify the RLFA selection algorithm.

1. Apply the LFA policy, which is the policy that corresponds to the protected link S-E. If the **node-protect** command is enabled in RLFA, the applied LFA policy is the one corresponding to the primary next hop to the protected node.
2. Apply the following admin-group constraints to each neighbor [Ni].
   a. Prune links that do not include one or more of the admin-groups in the **include-group** statements in the route next-hop policy template.
   b. Prune links that belong to admin-groups that have been explicitly excluded using the **exclude-group** statement in the route next-hop policy template.
   c. Exclude a neighbor [Ni] when it is only reachable by interfaces that violate the previous admin-group constraints.
3. Apply the following SRLG constraints to each neighbor [Ni].
   a. Prune links that belong to the SRLGs used by the primary next hop of a destination prefix.
   b. Exclude a neighbor [Ni] when it is reachable only by interfaces that violate the above SRLG constraint.
4. Apply one of the following **protection-type** preferences:
   - Perform one of the following if **protection-type**=**link**.
     - If **node-protect** is disabled in RLFA, select the results of the link-protect calculation. In some cases, the computed backup might be node-protecting but shows as link-protect in the output of the **tools dump router ospf**/**isis sr-database** command.

  - If **node-protect** is enabled in RLFA, select the results of the link-protect calculation in preference over the results of the node-protect calculation.
- Perform one of the following if **protection-type**=**node**.
  - If **node-protect** is disabled in RLFA, select the results of the link-protect calculation.
  - If **node-protect** is enabled in RLFA, select the results of the node-protect calculation in preference over the results of the link-protect calculation.

5. Apply the **next-hop** type preference (not applicable to RLFA).
6. Select the best [Ni] next hop among the remaining ones in the paths as candidate PQ nodes of prefix E (acting as proxy for destination prefix D), according to the following rules (in ascending order):
   a. prefer the next hop, avoiding the pseudo-node (PN) used by the primary next hop
   b. within the remaining subset, prefer the node-protect type or link-protect type according to the value of the **protection-type** option in the route next-hop policy template
   c. within the remaining subset, select the best admin group or groups according to the preference specified in the value of the **include-group** option in the route next-hop policy template
   d. select the [Ni] next hops corresponding to the PQ nodes with the same lowest cost (for example, the closest to RLFA computing node [S])
   e. if more than one [Ni] next hop remains, select the next hops of the [Ni] with the lowest router ID (OSPF) or system ID (IS-IS)
   f. if more than one [Ni] next hop remains, select the next hops of the [Ni] corresponding to the PQ node with the lowest router ID (OSPF) or system ID (IS-IS)

2.1.9.4.1.2 Modifying the TI-LFA selection algorithm

This section provides detailed steps to modify the TI-LFA selection algorithm.

The admin-group and SLRG constraints are applied to the outgoing interfaces of the next hops of neighbors [Ni] resulting from the post-convergence SPF computation of the destination prefix D with link S-E removed (link-protect) or node E removed (node-protect). Consequently, the number of next hops and outgoing interfaces selected by the post-convergence SPF, which is influenced by the router **ecmp** value, may violate the LFA policy constraints.

Therefore, the destination prefix may remain unprotected, or may be protected with a less-preferred next hop by TI-LFA even if another LFA policy complies or a more preferred outgoing link exists but was not selected by the post-convergence SPF. This is because the post-convergence SPF part of TI-LFA must select the same outgoing interface and next hop as the post-convergence main SPF computation performed by the node for the destination prefix

The post-convergence SPF and PQ set computations are run for both link protection (with link S-E removed) and node protection (with node E removed), because there is a need to fall back to the less-preferred protection option according to the value of parameter **protection-type** in the LFA policy applied to a prefix.

Perform the following steps to modify the TI-LFA selection algorithm.

1. Apply the LFA policy that corresponds to the protected link S-E. If the **node-protect** command is enabled in TI-LFA, the applied LFA policy corresponds to the link that is the primary next hop to the protected node.

2. Apply the following admin-group constraints to each outgoing interface to a neighbor [Ni] in a post-convergence path to a destination prefix D.
   a. Prune links that do not include one or more of the admin-groups in the **include-group** statements in the route next-hop policy template.
   b. Prune links that belong to admin-groups that have been explicitly excluded using the **exclude-group** statement in the route next-hop policy template.
   c. Exclude a neighbor [Ni] when it is only reachable by outgoing interfaces that violate the previous admin-group constraints.
3. Apply the following SRLG constraints to each outgoing interface of a neighbor [Ni] found in the post-convergence path to a destination prefix D.
   a. Prune links that belong to the SRLGs used by the primary next hop of a destination prefix.
   b. Exclude a neighbor [Ni] when only reachable by outgoing interfaces that violate the previous SRLG constraint.
4. Apply one of the following **protection-type** preferences:
   - Perform one of the following if **protection-type**=**link**.
     – If **node-protect** is disabled in TI-LFA, select the results of link-protect calculation. In some cases, the computed backup might be node-protecting but show as link-protect in the output of the **tools dump router ospf**/**isis sr-database** command.
     – If **node-protect** is enabled in TI-LFA, select the results of link-protect calculation in preference over the results of the node-protect calculation.
   - Perform one of the following if **protection-type**=**node**.
     – If **node-protect** is disabled in TI-LFA, select the results of the link-protect calculation.
     – If **node-protect** is enabled in TI-LFA, select the results of the node-protect calculation in preference over the results of the link-protect calculation.
5. Apply the **next-hop** type preference (not applicable to TI-LFA).
6. Select the best [Ni] next hop among the remaining ones in the paths as candidate PQ sets of destination D, according to the following rules (in ascending order):
   a. prefer the next hop, avoiding the pseudo-node (PN) used by the primary next hop
   b. within the remaining subset, prefer the node-protect type or the link-protect type according to the value of the **protection-type** option in the route next-hop policy template
   c. within the remaining subset, select the best admin group or groups according to the preference specified in the value of the **include-group** option in the route next-hop policy template
   d. select the [Ni] next hops corresponding to the PQ sets with the lowest number of labels
   e. if more than one remains, select the next hops to the [Ni] with the lowest router ID (OSPF) or system ID (IS-IS)
   f. if more than one remains, select the next hops to the [Ni] corresponding to the Q node with the lowest router ID (OSPF) or system ID (IS-IS)

2.1.9.4.2 Application of LFA policy to adjacency SID tunnel

The modifications to TI-LFA and RLFA as described in Application of LFA policy to a segment routing node SID tunnel are similarly applied to an adjacency SID tunnel.

The LFA selection algorithm for an adjacency to a neighbor uses the following preference order:

1. Adjacency of an alternate parallel link to the same neighbor, determined as follows:
   a. apply admin-group and SRLG constraints of the LFA policy of the link of the protected adjacency
   b. select the adjacency with best admin-groups according to the preference specified in the value of the **include-group** option in the route next-hop policy template
   c. select the adjacency with lowest metric
   d. select the adjacency to the neighbor with the lowest router ID (OSPF) or system ID (IS-IS), and the lowest metric
   e. select the adjacency over the lowest interface index, and the lowest neighbor router ID (OSPF) or system ID (IS-IS)
2. ECMP next hop to a node-SID of the same neighbor, determined as follows:
   a. apply admin-group and SRLG constraints of the LFA policy of the link of the protected adjacency
   b. select the next hop with the best admin-groups according to the preference specified in the value of the **include-group** option in the route next-hop policy template
   c. select the next hop with lowest metric
   d. select the next hop to the neighbor with the lowest router ID (OSPF) or system ID (ISIS), and the lowest metric
   e. select the next hop over the lowest interface index, and the lowest neighbor router ID (OSPF) or system ID (IS-IS)
3. LFA backup outcome of a node SID of the same neighbor:

   select a LFA backup with an outgoing link that does not conflict with the LFA policy of the link of the protected adjacency

Note:
If a different LFA policy was already applied in the computation of the LFA backup of the node SID of the neighbor, it is possible that some links to that node SID may have been eliminated before applying the LFA policy of the link of the protected adjacency.

2.1.9.4.3 Application of LFA policy to backup node SID tunnel

The backup node SID feature allows OSPF to use the path to an alternate ABR as an RLFA backup for forwarding packets of prefixes outside the local area or domain when the path to the primary ABR fails.

This feature reduces the label stack size by omitting the PQ node label if a regular RLFA algorithm is run.

The backup node SID algorithm consists of the following steps:

1. Perform an SPF on the modified topology with the primary ABR removed.

   This action resolves the backup node SID using the path to the alternate ABR.
2. Install the ILM to use the backup node SID for transit traffic with the maximum ECMP next hops found in step 1.
3. Use the backup node SID as an RLFA backup for prefixes outside the local area or domain. This step is modified as follows to select the backup node SID by applying the LFA policy corresponding the primary next hop of these prefixes, as follows.

a. For each neighbor (Ni) found in step 1, use the LFA policy to select the best next-hop interface.

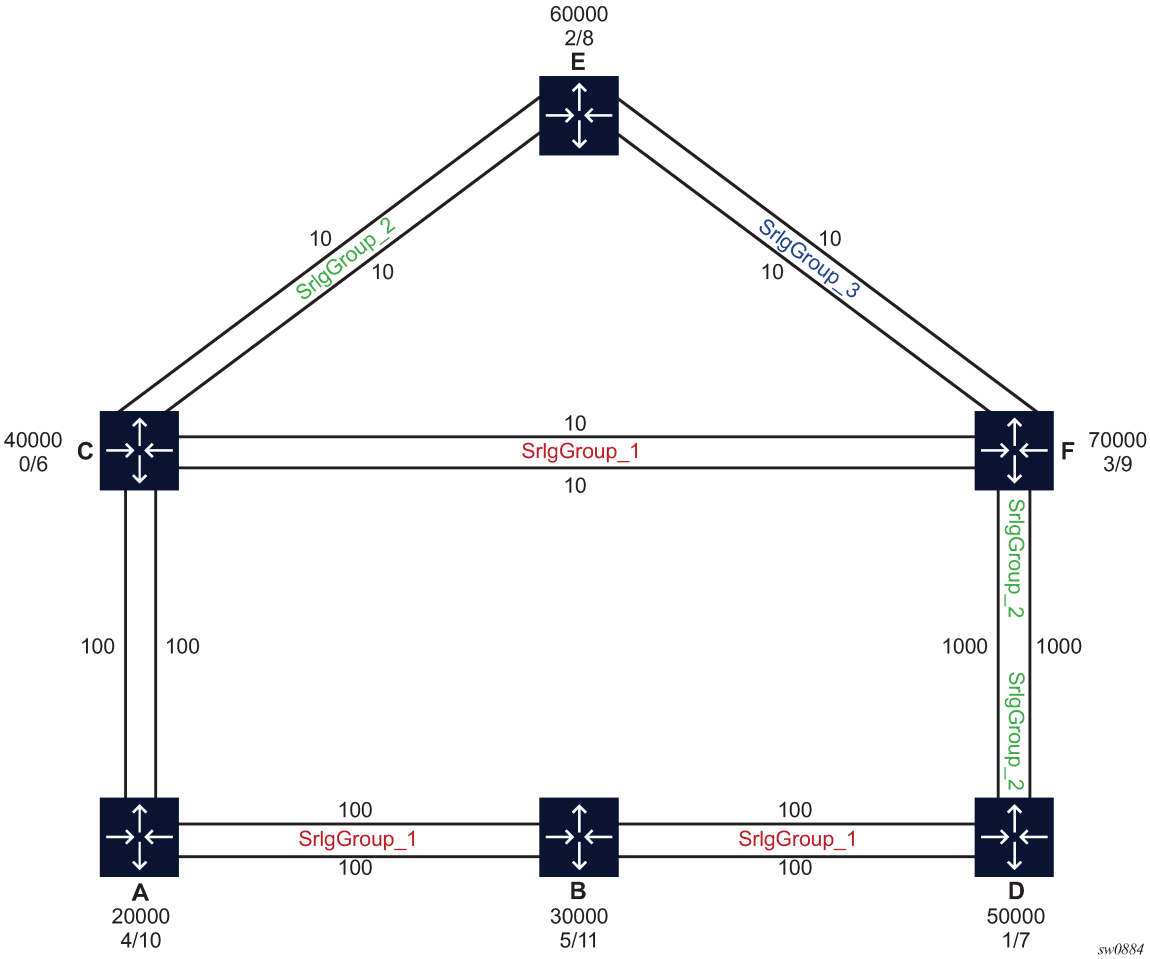
b. Among the remaining interfaces, use the LFA policy to select best (Ni) and select its interface.

Note: A backup node SID is always preferred to a regular RLFA backup. This does not change after applying the LFA policy because the main objective of the backup node SID feature is to reduce the label stack size of the backup tunnel.

2.1.9.4.4 Configuration example of LFA policy use in remote LFA and TI-LFA

The following figure shows an example network topology that uses the OSPF routing protocol and in which the user assigns an SRLG ID to each group of OSPF links to represent fate-sharing among the links in the group. Assume the router **ecmp** value is set to **1**.

Figure 11: Application of LFA policy to RLFA and TI-LFA



The user wants to enforce that the LFA backup computed and programmed by each node for a specific destination prefix avoids the SRLG ID of the primary next hop of that prefix. To that effect, the user applies an LFA policy to each link that is used as a primary next hop to reach destination prefixes.

For example, node F uses the top interface to node C as the primary next hop for the SR-OSPF tunnel to the SID of node C. The LFA policy states that the LFA backup must exclude outgoing interfaces that are members of the SRLG ID of the interface of the primary next hop. Therefore, node F must select an LFA backup that avoids SRLG ID=SrlgGroup_1.

Node F enables base LFA, remote LFA with **node-protect**, and TI-LFA with **node-protect** on the OSPF routing instance. The LFA SPF yields the following candidate LFA backup paths for the tunnel to the SID of node C.

1. Base LFA returns two backup paths: next hop over the second interface to C (cost 10) and next hop over the interface to node E (cost 20).

   After applying the LFA policy, only next hop over the interface to node E (cost 20) remains. The second interface to node C is also a member of SRLG ID=SrlgGroup_1 and, therefore, the LFA next hop using it is excluded.

2. TI-LFA returns a single backup path: the next hop over the second interface to C (cost 10).

   After applying the LFA policy, no LFA backup path remains.

3. Remote LFA returns two backup paths, one backup path by PQ node C over the second interface to C (cost 10) and one by PQ node E over the interface to node E (cost 20).

   After applying the LFA policy, only the backup path by PQ node E over the interface to node E (cost 20) remains.

4. The LFA backup paths found by all three LFA methods are only link-protecting because node C is a neighbor of node F.

5. The final outcome is the selection among the LFA methods and base LFA is preferred to RLFA; therefore, the next hop over the interface to node E (cost 20) is selected and programmed by node F as the backup path for the SR-OSPF tunnel to the SID of node C.

6. The adjacency from node F to node C over first interface to node C also inherits the same LFA backup path as the node SID of C because the same LFA policy applies.

Example

The following are excerpts of the CLI configuration of node F in this specific example. The commands relevant to the LFA policy applied to link F-C are identified by arrows.

In addition, the output of show commands in node F highlights both the primary and the link-protect base LFA backup for both the node SID tunnel to C and the adjacency SID tunnel over the first interface to node C.

Because C is the termination for both its node SID and the adjacency SID tunnels from node F, only link protection can be provided as shown by the output of **tools**>**dump**>**router**>**ospf sr-database** command (field L(R)). However, the output of the same show command for the tunnel to the SID of node D indicates the base LFA backup over the direct interface to node D is node-protecting (field Tn(R)).

```
 *A:Dut-F>config>router# info
----------------------------------------------
#----------------------------------------------------
echo "IP Configuration"
#----------------------------------------------------
        if-attribute                            <-------
            srlg-group "SrlgGroup_1" value 1    <-------
            srlg-group "SrlgGroup_2" value 2
            srlg-group "SrlgGroup_3" value 3
        exit
```

```
        route-next-hop-policy                   <-------
            begin                               <-------
            template "templateSrlgGroup_1"      <-------
                srlg-enable
            exit
            template "templateSrlgGroup_2"
                srlg-enable
            exit
            template "templateSrlgGroup_3"
                srlg-enable
            exit
            commit
        exit
        interface "DUTF_TO_DUTC.1.0"            <-------
            address 1.0.36.6/24
            secondary 51.0.36.6/24
            port 1/1/4:1
            mac 00:00:00:00:00:06
            ipv6
                address 3ffe::100:2406/120 primary-preference 1
                address 3ffe::3300:2406/120 primary-preference 2
            exit
            if-attribute                        <-------
                srlg-group "SrlgGroup_1"        <-------
            exit
            no shutdown
        exit
        interface "DUTF_TO_DUTC.2.0"            <-------
            address 2.0.36.6/24
            secondary 52.0.36.6/24
            port 1/1/4:2
            mac 00:00:00:00:00:06
            ipv6
                address 3ffe::200:2406/120 primary-preference 1
                address 3ffe::3400:2406/120 primary-preference 2
            exit
            if-attribute                        <-------
                srlg-group "SrlgGroup_1"        <-------
            exit
            no shutdown
        exit
        interface "DUTF_TO_DUTD.1.0"
            address 1.0.46.6/24
            secondary 51.0.46.6/24
            port 1/1/1:1
            mac 00:00:00:00:00:06
            ipv6
                address 3ffe::100:2e06/120 primary-preference 1
                address 3ffe::3300:2e06/120 primary-preference 2
            exit
            if-attribute
                srlg-group "SrlgGroup_2"
            exit
            no shutdown
        exit
        interface "DUTF_TO_DUTD.2.0"
            address 2.0.46.6/24
            secondary 52.0.46.6/24
            port 1/1/1:2
            mac 00:00:00:00:00:06
            ipv6
                address 3ffe::200:2e06/120 primary-preference 1
                address 3ffe::3400:2e06/120 primary-preference 2
            exit
```

```
            if-attribute
                srlg-group "SrlgGroup_2"
            exit
            no shutdown
        exit
        interface "DUTF_TO_DUTE.1.0"          <-------
            address 1.0.56.6/24
            secondary 51.0.56.6/24
            port 1/1/2:1
            mac 00:00:00:00:00:06
            ipv6
                address 3ffe::100:3806/120 primary-preference 1
                address 3ffe::3300:3806/120 primary-preference 2
            exit
            if-attribute                      <-------
                srlg-group "SrlgGroup_3"      <-------
            exit
            no shutdown
        exit
        interface "DUTF_TO_DUTE.2.0"          <-------
            address 2.0.56.6/24
            secondary 52.0.56.6/24
            port 1/1/2:2
            mac 00:00:00:00:00:06
            ipv6
                address 3ffe::200:3806/120 primary-preference 1
                address 3ffe::3400:3806/120 primary-preference 2
            exit
            if-attribute                      <-------
                srlg-group "SrlgGroup_3"      <-------
            exit
            no shutdown
        exit
        interface "loopbackF.1.0"
            address 1.0.66.6/32
            secondary 51.0.66.6/32
            loopback
            ipv6
                address 3ffe::100:4206/128 primary-preference 1
                address 3ffe::3300:4206/128 primary-preference 2
            exit
            no shutdown
        exit
        interface "loopbackF.2.0"
            address 2.0.66.6/32
            secondary 52.0.66.6/32
            loopback
            ipv6
                address 3ffe::200:4206/128 primary-preference 1
                address 3ffe::3400:4206/128 primary-preference 2
            exit
            no shutdown
        exit
        interface "system"
            address 10.20.1.6/32
            ipv6
                address 3ffe::a14:106/128
            exit
            no shutdown
        exit
        ip-fast-reroute
        router-id 10.20.1.6
#--------------------------------------------------
echo "MPLS Label Range Configuration"
```

```
#--------------------------------------------------
        mpls-labels
            sr-labels start 20000 end 80000
        exit
#--------------------------------------------------
echo "OSPFv2 Configuration"
#--------------------------------------------------
        ospf 0 10.20.1.6
            traffic-engineering
            database-export identifier 0
            advertise-router-capability area
            loopfree-alternates                <-------
                remote-lfa                     <-------
                    node-protect               <-------
                exit                           <-------
                ti-lfa max-sr-frr-labels 3     <-------
                    node-protect               <-------
                exit                           <-------
            exit                               <-------
            segment-routing
                prefix-sid-range start-label 70000 max-index 999
                egress-statistics
                    adj-set
                    adj-sid
                    node-sid
                exit
                ingress-statistics
                    adj-set
                    adj-sid
                    node-sid
                exit
                no shutdown
            exit
            area 0.0.0.0
                interface "system"
                    node-sid index 9
                    no shutdown
                exit
                interface "DUTF_TO_DUTC.1.0"       <-------
                    interface-type point-to-point
                    hello-interval 2
                    dead-interval 10
                    metric 10
                    lfa-policy-map route-nh-template "templateSrlgGroup_1"  <-------
                    no shutdown
                exit
                interface "DUTF_TO_DUTD.1.0"
                    interface-type point-to-point
                    hello-interval 2
                    dead-interval 10
                    metric 1000
                    lfa-policy-map route-nh-template "templateSrlgGroup_2"
                    no shutdown
                exit
                interface "DUTF_TO_DUTE.1.0"
                    interface-type point-to-point
                    hello-interval 2
                    dead-interval 10
                    metric 10
                    lfa-policy-map route-nh-template "templateSrlgGroup_3"
                    no shutdown
                exit
                interface "loopbackF.1.0"
                    node-sid index 3
```

```
                    no shutdown
                exit
                interface "DUTF_TO_DUTC.2.0"
                    interface-type point-to-point
                    hello-interval 2
                    dead-interval 10
                    metric 10
                    lfa-policy-map route-nh-template "templateSrlgGroup_4"
                    no shutdown
                exit
                interface "DUTF_TO_DUTD.2.0"
                    interface-type point-to-point
                    hello-interval 2
                    dead-interval 10
                    metric 1000
                    lfa-policy-map route-nh-template "templateSrlgGroup_5"
                    no shutdown
                exit
                interface "DUTF_TO_DUTE.2.0"
                    interface-type point-to-point
                    hello-interval 2
                    dead-interval 10
                    metric 10
                    lfa-policy-map route-nh-template "templateSrlgGroup_6"
                    no shutdown
                exit
                interface "loopbackF.2.0"
                    node-sid index 15
                    no shutdown
                exit
            exit
            no shutdown
        exit
-----------------------------------------------
*A:Dut-F# tools dump router segment-routing tunnel
===============================================================================================
====
Legend: (B) - Backup Next-hop for Fast Re-Route

        (D) - Duplicate

label stack is ordered from top-most to bottom-most

===============================================================================================
====
-----------------------------------------------------------------------------------------------
--------+
 Prefix
        |
 Sid-Type        Fwd-Type       In-Label  Prot-Inst
        |
                 Next Hop(s)                                        Out-Label(s) Interface/
Tunnel-ID |
-----------------------------------------------------------------------------------------------
--------+
 1.0.33.3                                          <-------
 Node            Orig/Transit   70000     OSPF-0     <-------
                 1.0.36.3                                           40000       DUTF_TO_
DUTC.1.0 <-------
              (B)1.0.56.5                                           60000       DUTF_TO_
DUTE.1.0 <-------
 1.0.44.4                                          <-------
 Node            Orig/Transit   70001     OSPF-0     <-------
```

```
                1.0.36.3                                        40001       DUTF_TO_
DUTC.1.0 <-------
             (B)1.0.46.4                                        50001       DUTF_TO_
DUTD.1.0 <-------
 1.0.55.5
 Node           Orig/Transit   70002     OSPF-0
                1.0.56.5                                        60002       DUTF_TO_
DUTE.1.0
             (B)1.0.36.3                                        40002       DUTF_TO_
DUTC.1.0
 1.0.66.6
 Node           Terminating    70003     OSPF-0
 1.0.11.1
 Node           Orig/Transit   70004     OSPF-0
                1.0.36.3                                        40004       DUTF_TO_
DUTC.1.0
             (B)1.0.46.4                                        50004       DUTF_TO_
DUTD.1.0
 1.0.22.2
 Node           Orig/Transit   70005     OSPF-0
                1.0.36.3                                        40005       DUTF_TO_
DUTC.1.0
             (B)1.0.46.4                                        50005       DUTF_TO_
DUTD.1.0
 10.20.1.3
 Node           Orig/Transit   70006     OSPF-0
                1.0.36.3                                        40006       DUTF_TO_
DUTC.1.0
             (B)1.0.56.5                                        60006       DUTF_TO_
DUTE.1.0
 10.20.1.4
 Node           Orig/Transit   70007     OSPF-0
                1.0.36.3                                        40007       DUTF_TO_
DUTC.1.0
             (B)1.0.46.4                                        50007       DUTF_TO_
DUTD.1.0
 10.20.1.5
 Node           Orig/Transit   70008     OSPF-0
                1.0.56.5                                        60008       DUTF_TO_
DUTE.1.0
             (B)1.0.36.3                                        40008       DUTF_TO_
DUTC.1.0
 10.20.1.6
 Node           Terminating    70009     OSPF-0
 10.20.1.1
 Node           Orig/Transit   70010     OSPF-0
                1.0.36.3                                        40010       DUTF_TO_
DUTC.1.0
             (B)1.0.46.4                                        50010       DUTF_TO_
DUTD.1.0
 10.20.1.2
 Node           Orig/Transit   70011     OSPF-0
                1.0.36.3                                        40011       DUTF_TO_
DUTC.1.0
             (B)1.0.46.4                                        50011       DUTF_TO_
DUTD.1.0
 2.0.33.3
 Node           Orig/Transit   70012     OSPF-0
                1.0.36.3                                        40012       DUTF_TO_
DUTC.1.0
             (B)1.0.56.5                                        60012       DUTF_TO_
DUTE.1.0
 2.0.44.4
 Node           Orig/Transit   70013     OSPF-0
```

```
                  1.0.36.3                                          40013       DUTF_TO_
DUTC.1.0
              (B)1.0.46.4                                           50013       DUTF_TO_
DUTD.1.0
 2.0.55.5
 Node             Orig/Transit   70014     OSPF-0
                  1.0.56.5                                          60014       DUTF_TO_
DUTE.1.0
              (B)1.0.36.3                                           40014       DUTF_TO_
DUTC.1.0
 2.0.66.6
 Node             Terminating    70015     OSPF-0
 2.0.11.1
 Node             Orig/Transit   70016     OSPF-0
                  1.0.36.3                                          40016       DUTF_TO_
DUTC.1.0
              (B)1.0.46.4                                           50016       DUTF_TO_
DUTD.1.0
 2.0.22.2
 Node             Orig/Transit   70017     OSPF-0
                  1.0.36.3                                          40017       DUTF_TO_
DUTC.1.0
              (B)1.0.46.4                                           50017       DUTF_TO_
DUTD.1.0
 2.0.56.5
 Adjacency        Transit        524282    OSPF-0
                  2.0.56.5                                          3           DUTF_TO_
DUTE.2.0
              (B)1.0.56.5                                           3           DUTF_TO_
DUTE.1.0
  2.0.46.4
 Adjacency        Transit        524283    OSPF-0
                  2.0.46.4                                          3           DUTF_TO_
DUTD.2.0
              (B)1.0.36.3                                           40001       DUTF_TO_
DUTC.1.0
 2.0.36.3
 Adjacency        Transit        524284    OSPF-0
                  2.0.36.3                                          3           DUTF_TO_
DUTC.2.0
              (B)1.0.36.3                                           3           DUTF_TO_
DUTC.1.0
 1.0.56.5
 Adjacency        Transit        524285    OSPF-0
                  1.0.56.5                                          3           DUTF_TO_
DUTE.1.0
              (B)1.0.36.3                                           40002       DUTF_TO_
DUTC.1.0
 1.0.46.4
 Adjacency        Transit        524286    OSPF-0
                  1.0.46.4                                          3           DUTF_TO_
DUTD.1.0
              (B)1.0.36.3                                           40001       DUTF_TO_
DUTC.1.0
 1.0.36.3                                          <-------
 Adjacency        Transit        524287    OSPF-0   <-------
                  1.0.36.3                                          3           DUTF_TO_
DUTC.1.0 <----
              (B)1.0.56.5                                           60000       DUTF_TO_
DUTE.1.0 <----
-------------------------------------------------------------------------------------------
---+
No. of Entries: 24
```

```
-------------------------------------------------------------------------------------------
---+
*A:Dut-F#
*A:Dut-F#    tools dump router ospf sr-database
===============================================================================
Rtr Base OSPFv2 Instance 0 Segment Routing Database
===============================================================================
SID         Label St Type Prefix                                      Stitching
                                 AdvRtr            Area Flags          FRR
-------------------------------------------------------------------------------
0           70000 +R   T1 1.0.33.3/32                                     -   <-------
                           10.20.1.3         0.0.0.0 [NnP       ]  L(R)     <-------
1           70001 +R   T1 1.0.44.4/32                                     -   <-------
                           10.20.1.4         0.0.0.0 [NnP       ] Tn(R)     <-------
2           70002 +R   T1 1.0.55.5/32                                     -
                           10.20.1.5         0.0.0.0 [NnP       ]  L(R)
3           70003 +R  LT1 1.0.66.6/32                                     -
                           10.20.1.6         0.0.0.0 [NnP       ]     -
4           70004 +R   T1 1.0.11.1/32                                     -
                           10.20.1.1         0.0.0.0 [NnP       ] Tn(R)
5           70005 +R   T1 1.0.22.2/32                                     -
                           10.20.1.2         0.0.0.0 [NnP       ] Tn(R)
6           70006 +R   T1 10.20.1.3/32                                    -
                           10.20.1.3         0.0.0.0 [NnP       ]  L(R)
7           70007 +R   T1 10.20.1.4/32                                    -
                           10.20.1.4         0.0.0.0 [NnP       ] Tn(R)
8           70008 +R   T1 10.20.1.5/32                                    -
                           10.20.1.5         0.0.0.0 [NnP       ]  L(R)
9           70009 +R  LT1 10.20.1.6/32                                    -
                           10.20.1.6         0.0.0.0 [NnP       ]     -
10          70010 +R   T1 10.20.1.1/32                                    -
                           10.20.1.1         0.0.0.0 [NnP       ] Tn(R)
11          70011 +R   T1 10.20.1.2/32                                    -
                           10.20.1.2         0.0.0.0 [NnP       ] Tn(R)
12          70012 +R   T1 2.0.33.3/32                                    -
                           10.20.1.3         0.0.0.0 [NnP       ]  L(R)
13          70013 +R   T1 2.0.44.4/32                                    -
                           10.20.1.4         0.0.0.0 [NnP       ] Tn(R)
14          70014 +R   T1 2.0.55.5/32                                    -
                           10.20.1.5         0.0.0.0 [NnP       ]  L(R)
15          70015 +R  LT1 2.0.66.6/32                                    -
                           10.20.1.6         0.0.0.0 [NnP       ]     -
16          70016 +R   T1 2.0.11.1/32                                    -
                           10.20.1.1         0.0.0.0 [NnP       ] Tn(R)
17          70017 +R   T1 2.0.22.2/32                                    -
                           10.20.1.2         0.0.0.0 [NnP       ] Tn(R)
-------------------------------------------------------------------------------
No. of Entries: 18
-------------------------------------------------------------------------------
St:   R:reported  I:incomplete  W:wrong  N:not reported  F:failed
      +:SR-ack  -:no route
Type: L:local  M: mapping Srv  Tx: route type
FRR:  L:Lfa  R:RLfa  T:TiLfa  (R):Reported  (F):Failed
      Ln, Rn, Tn: FRR providing node-protection
===============================================================================
*A:Dut-F#
```

2.1.9.5 LFA protection using a segment routing backup node SID

One of the challenges in MPLS deployments across multiple IGP areas or domains, such as in seamless MPLS design, is the provisioning of FRR local protection in access and metropolitan domains that make

use of a ring, a square, or a partial mesh topology. To implement IP, LDP, or SR FRR in these topologies, the remote LFA feature must be implemented. Remote LFA provides a Segment Routing (SR) tunneled LFA next hop for an IP prefix, an LDP tunnel, or an SR tunnel. For prefixes outside of the area or domain, the access or aggregation router must push the following labels:

- service label
- BGP label for the destination PE
- LDP/RSVP/SR label to reach the exit ABR or ASBR
- label for the remote LFA next hop

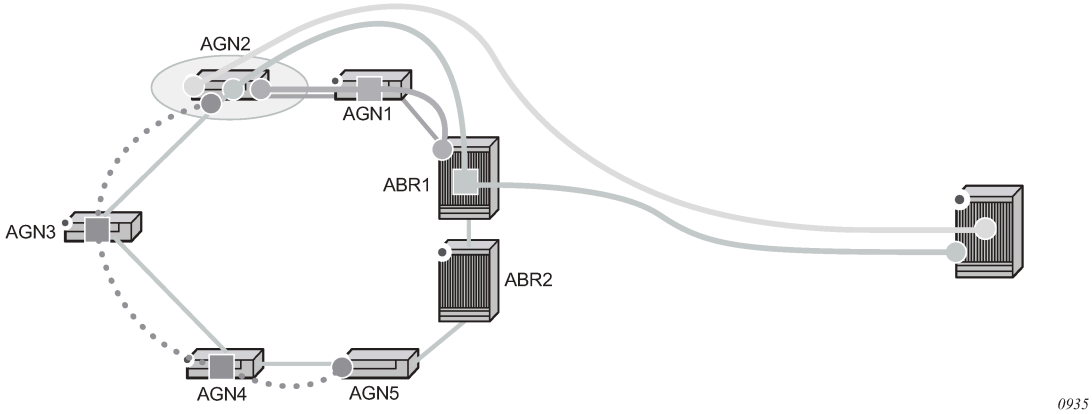
Small routers deployed in these parts of the network have limited MPLS label stack size support.

Figure 12: Label stack for remote LFA in ring topology shows the label stack required for the primary next hop and the remote LFA next hop computed by aggregation node AGN2 for the inter-area prefix of a remote PE. For an inter-area BGP label unicast route prefix for which ABR1 is the primary exit ABR, AGN2 resolves the prefix to the transport tunnel of ABR1 and, therefore, uses the remote LFA next hop of ABR1 for protection. The primary next hop uses two transport labels plus a service label. The remote LFA next hop for ABR1 uses PQ node AGN5 and pushes three transport labels plus a service label.

Seamless MPLS with Fast Restoration requires up to four labels to be pushed by AGN2, as shown in the following figure.

Figure 12: Label stack for remote LFA in ring topology

Label Location	Label Name	Assigned By	Protocol	Use
Label 1 (Bottom)	Service (PW, VC) Label	Remote PE	MP-BGP, T-LDP	Identifies Service on Remote PE
Label 2	Inter-Area Label	ABR1	BGP-LU	Identifies Path to Remote PE
Label 3	Intra-Area	AGN1	LDP, RSVP, SR	Identifies Path to ABR1
Label 4 (Top)	R-LFA Label	AGN3	LDP, RSVP, SR	Identifies Path to AGN5



The objective of the LFA protection with a backup node SID feature is to reduce the label stack pushed by AGN2 for BGP label unicast inter-area prefixes. When link AGN2-AGN1 fails, packets are directed away from the failure and forwarded toward ABR2, which acts as the backup for ABR1 (and the other way around when ABR2 is the primary exit ABR for the BGP label unicast inter-area prefix). This requires that ABR2 advertise a special label for the loopback of ABR1 that attracts packets normally destined for ABR1. These packets are forwarded by ABR2 to ABR1 via the inter-ABR link.

As a result, AGN2 pushes the label advertised by ABR2 to back up ABR1 on top of the BGP label for the remote PE and the service label. This keeps the label stack the same size for the LFA next hop to be the

same size as that of the primary next hop. It is also the same size as the remote LFA next hop for the local prefix within the ring.

2.1.9.5.1 Configuring LFA using a backup node SID in OSPF

LFA using a backup node SID is enabled by configuring a backup node SID at an ABR/ASBR that acts as a backup to the primary exit ABR/ASBR of inter-area/inter-AS routes learned as BGP labeled routes.

```
config>router>ospf>segment-routing$
        – backup-node-sid ip-prefix/prefix-length index 0..4294967295
        – backup-node-sid ip-prefix/prefix-length label 1..4294967295
```

The user can enter either a label or an index for the backup node SID.

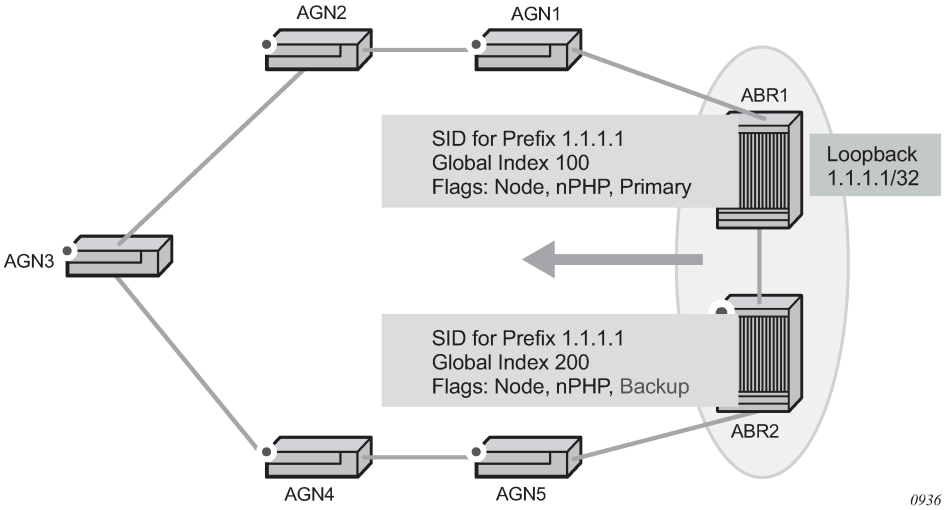
Note: This feature only allows the configuration of a single backup node SID per OSPF instance and per ABR/ASBR. In other words, only one pair of ABR/ASBR nodes can back up each other in an OSPF domain. Each time the user invokes the preceding command within the same OSPF instance, it overrides any previous configuration of the backup node SID. The same ABR/ASBR can, however, participate in multiple OSPF instances and provide a backup support within each instance.

2.1.9.5.2 Detailed operation of LFA protection using a backup node SID

As shown in the following figure, LFA for seamless MPLS supports environments where the boundary routers are either:

- ABR nodes that connect with Interior Border Gateway Protocol (IBGP) multiple domains, each using a different area of the same IGP instance
- ASBR nodes that connect domains running different IGP instances and use IBGP within a domain and External Border Gateway Protocol (EBGP) to the other domains

Figure 13: Backup ABR node SID



The following steps describe the configuration and behavior of LFA protection using a backup node SID.

1. The user configures node SID 100 in ABR1 for its loopback prefix 1.1.1.1/32. This is the regular node SID. ABR1 advertises the prefix SID sub-TLV for this node SID in the IGP and installs the ILM using a unique label.
2. Each router receiving the prefix sub-TLV for node SID 100 resolves it as described in Segment routing in shortest path forwarding. Changes to the programming of the backup NHLFE of node SID 100, based on receiving the backup node SID for prefix 1.1.1.1/32, are defined in Duplicate SID handling.
3. The user configures a backup node SID 200 in ABR2 for the loopback 1.1.1.1/32 of ABR1. The SID value must be different from that assigned by ABR1 for the same prefix. ABR2 installs the ILM, which performs a swap operation from the label of SID 200 to that of SID 100. The ILM must point to a direct link and next hop to reach 1.1.1.1/32 of ABR1 as its primary next hop. The IGP examines all adjacencies established in the same area as that of prefix 1.1.1.1/32 and determines which ones have ABR1 as a direct neighbor and with the best cost. If more than one adjacency has the best cost, the IGP selects the one with the lowest interface index. If there is no adjacency to reach ABR2, the prefix SID for the backup node is flushed and is not resolved, which prevents any other non-direct path being used to reach ABR1. As a result, any received traffic on the ILM of SID 200 traffic is blackholed.
4. If resolved, ABR2 advertises the prefix SID sub-TLV for backup node SID 200 and indicates in the SR Algorithm field that a modified SPF algorithm, referred to as "Backup-constrained-SPF", is required to resolve this node SID.
5. Each router receiving the prefix sub-TLV for backup node SID 200 and performs the following resolution steps.

   Note: The following resolution steps do not require a CLI command to be enabled.

   a. The router determines which router is being backed up. This is achieved by checking the router ID owner of the prefix sub-TLV that was advertised with the same prefix but without the backup flag and which is used as the best route for the prefix. In this case, ABR1 is the router being backed up. Then the router runs a modified SPF by removing node ABR1 from the topology to resolve backup node SID 200. The primary next hop points to the path to ABR2 in the counterclockwise direction of the ring.

      The router does not compute an LFA or a remote LFA for back node SID 200 because the main SPF used a modified topology.

   b. The router installs the ILM and primary NHLFE for the backup node SID.

      Only a swap label operation is configured by all routers for the backup node SID. There is no push operation, and no tunnel for the backup node SID is added into the TTM.

   c. The router programs the backup node SID as the LFA backup for the SR tunnel to node SID of 1.1.1.1/32 of ABR1. In other words, each router overrides the remote LFA backup for prefix 1.1.1.1/32, which is normally PQ node AGN5.

   d. If a router, such as AGN1, is adjacent to ABR1, it also programs the backup node SID as the LFA backup for the protection of any adjacency SID to ABR1.
6. When node AGN2 resolves a BGP labeled route for an inter-area prefix for which the primary ABR exit router is ABR1, it uses the backup node SID of ABR1 as the remote LFA backup instead of the SID to the PQ node (AGN5 in this example) to save on the pushed label stack.

   AGN2 continues to resolve the prefix SID for any remote PE prefix that is summarized into the local area of AGN2, as usual. AGN2 programs a primary next hop and a remote LFA next hop. Remote LFA uses AGN5 as the PQ node and pushes two labels, as it would for an intra-area prefix SID. There is

no need to use the backup node SID for this prefix SID and force its backup path to go to ABR1. The backup path may exit from ABR2 if the cost from ABR2 to the destination prefix is shorter.

7. If the user excludes a link from LFA in the IGP instance using the **config**>**router**>**ospf**>**area**>**interface**>**loopfree-alternate-exclude** command, a backup node SID that resolves to that interface is not used as a remote LFA backup in the same way as regular LFA or PQ remote LFA next hop behavior.
8. If the OSPF neighbor of a router is put into overload or if the metric of an OSPF interface to that neighbor is set to LSInfinity (0xFFFF), a backup node SID that resolves to that neighbor is not used as a remote LFA backup in the same way as regular LFA or PQ remote LFA next hop behavior.
9. The LFA policy is supported with a backup node SID. See Application of LFA policy to backup node SID tunnel.

2.1.9.5.3 Duplicate SID handling

When the IGP issues or receives an LSA or LSP containing a prefix SID sub-TLV for a node SID or a backup node SID with a SID value that is a duplicate of an existing SID or backup node SID, the resolution as described in the following table is followed.

Table 4: Handling of duplicate SIDs

	New LSA/LSP			
Old LSA/LSP	**Backup node SID**	**Local backup node SID**	**Node SID**	**Local node SID**
Backup Node SID	Old	New	New	New
Local Backup Node SID	Old	Equal	New	New
Node SID	Old	Old	Equal/Old[1]	Equal/New[2]
Local Node SID	Old	Old	Equal/Old[1]	Equal/Old[1]

2.1.9.5.4 OSPF control plane extensions

All routers supporting OSPF control plane extensions must advertise support of the Backup-constrained-SPF algorithm of value 2 in the SR-Algorithm TLV, which is advertised in the Router Information Opaque LSA. This is in addition to the default supported algorithm "IGP-metric-based-SPF" of value 0. The

1 Equal/Old means the following.

- If the prefix is duplicate, it is equal and no change is needed. Keep the old LSA/LSP.
- If the prefix is not duplicate, still keep the old LSA/LSP.

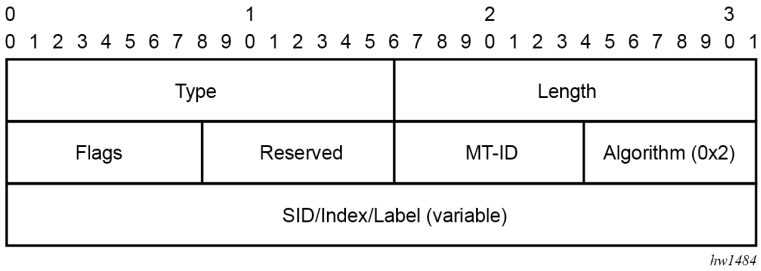
2 Equal/New means the following.

- If the prefix is duplicate, it is equal and no change is needed. Keep the old LSA/LSP.
- If the prefix is not duplicate, pick a new prefix and use the new LSA/LSP.

following figure shows the encoding of the prefix SID sub-TLV to indicate a node SID of type backup and to indicate the modified SPF algorithm in the SR Algorithm field.

Note: The values used in the Flags field and in the Algorithm field are SR OS proprietary. The new Algorithm (0x2) field and values are used by this feature.

Figure 14: OSPF Prefix SID sub-TLV



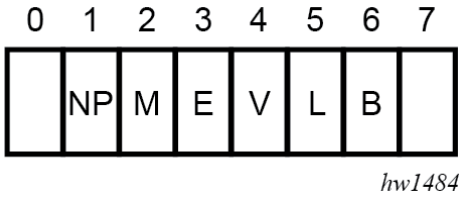
The following table lists the OSPF Prefix SID sub-TLV main field values.

Table 5: OSPF Prefix SID sub-TLV main fields

Field	Value
Type	2
Length	variable
Flags	1 octet field

The following figure shows the details of the OSPF Prefix SID sub-TLV flag field; the B-flag is new and SR OS proprietary.

Figure 15: OSPF Prefix SID sub-TLV flags



The following table describes OSPF Prefix SID sub-TLV flags, including the new B-flag.

Table 6: OSPF Prefix SID sub-TLV flags

Flag	Description
NP-Flag	No-PHP flag If set, the penultimate hop must not pop the prefix SID before delivering the packet to the node that advertised the prefix SID.

Flag	Description
M-Flag	Mapping Server Flag If set, the SID is advertised from the segment routing mapping server functionality as described in *draft-filsfils-spring-segment-routing-ldp-interop*.
E-Flag	Explicit-Null Flag If set, any upstream neighbor of the prefix SID originator must replace the prefix SID with a prefix SID having an Explicit-NULL value (0 for IPv4) before forwarding the packet.
V-Flag	Value/Index Flag If set, the prefix SID carries an absolute value. If not set, the prefix SID carries an index.
L-Flag	Local/Global Flag If set, the value or index carried by the prefix SID has local significance. If not set, then the value or index carried by this sub-TLV has global significance.
B-Flag	This flag is used by the Protection using backup node SID feature. If set, the SID is a backup SID for the prefix. This value is SR OS proprietary.
Other bits	Reserved These must be zero when sent and are ignored when received.
MT-ID	Multitopology ID, as defined in RFC 4915.
Algorithm	This octet identifies the algorithm that the prefix SID is associated with. A value of (0x2) indicates the modified SPF algorithm, which removes from the topology the node that is backed up by the backup node SID. This value is SR OS proprietary.
SID/Index/Label	Based on the V and L flags, this field contains either: • a 32-bit index defining the offset in the SID or Label space advertised by this router • a 24-bit label where the 20 rightmost bits are used for encoding the label value

2.1.9.6 Multihomed prefix LFA extensions in SR-OSPF

This feature makes use of the Multihomed Prefix (MHP) model described in RFC 8518 to compute a backup IP next-hop using an alternate ABR or ASBR for external prefixes and to an alternate router owner for local anycast prefixes.

The feature applies to OSPF routes of external /32 prefixes (OSPFv2 routes types 3, 4, 5, and 7) and local /32 anycast prefixes if the prefix is not protected by base LFA.

The computed IP next-hop based backup path is programmed for SR-OSPF node SID tunnels of external /32 prefixes and to /32 prefixes in same area as the computing node and which are advertised by multiple routers (anycast prefixes) in both algorithm 0 and flexible-algorithm numbers.

See “Multihomed prefix LFA extensions in OSPF” in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide* for more information about the configuration of this feature.

2.1.9.7 Multihomed prefix LFA extensions in SR IS-IS and SRv6 IS-IS

This feature makes use of the Multihomed Prefix (MHP) model described in RFC 8518 to compute a backup IP next-hop using an alternate ABR or ASBR for external prefixes and to an alternate router owner for local anycast prefixes.

The algorithm described in RFC 8518 is limited in scope to only computed backup paths consisting of direct IP next hops and tunneled next hops (IGP shortcuts).

The computed backup paths are added to IS-IS routes of external /32 and /128 prefixes and intra-area /32 and /128 anycast prefixes in the Routing Table Manager (RTM) if the prefix is not protected by the base LFA.

The computed backup path is also programmed for the following tunnels:

- SR IS-IS IPv4 and IPv6 node SID tunnels of external /32 and /128 prefixes and of intra-area /32 and /128 anycast prefixes, in both algorithm 0 and flexible algorithm numbers
- SRv6 IS-IS locator routes and tunnels of external prefixes and of intra-area anycast prefixes of any size, in both algorithm 0 and flexible algorithm numbers

As a result, an SR-TE LSP, an SR-MPLS policy, or an SRv6 policy that uses an SR IS-IS SID or an SRv6 IS-IS SID of those same prefixes in its configured or computed SID list benefits from the multihomed prefix LFA protection.

See “Multihomed prefix LFA extensions in IS-IS” in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide* for more information about the configuration of this feature.

2.1.9.8 LFA solution across IGP area or instance boundary using SR repair tunnel in SR-OSPF

This feature enhances the IP next-hop based MHP backup path calculation specified in RFC 8518 with the addition of the support of an SR repair tunnel. The SR repair tunnel uses a PQ node or a P-Q set to reach the alternate exit ABR or ASBR for external prefixes, or alternate owner router for intra-area anycast prefixes. This capability is in addition to supporting the RFC 8518 algorithm used in the case where the path to prefix P using the alternate exit ABR or ASBR (or alternate owner router) is in the shortest path from the neighbor of the computing node.

This feature applies the computed backup path to SR-OSPF node SID tunnels of external /32 prefixes and to /32 prefixes in the same area as the computing node, and which are advertised by multiple routers (anycast prefixes) in both algorithm 0 and flexible-algorithm numbers. It also extends the protection to any SR-TE LSP or SR policy that uses an SR-OSPF SID of those same prefixes in its configured or computed SID list.

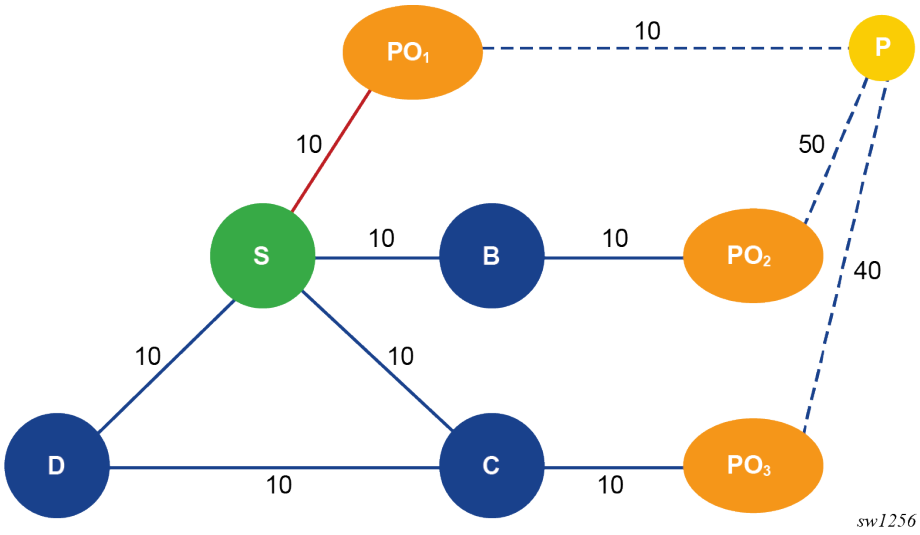
This feature shares the same configuration CLI commands as the MHP LFA feature as described in Multihomed prefix LFA extensions in SR-OSPF. After the IP next-hop based MHP LFA is enabled, the extensions to compute an SR repair tunnel for the MHP LFA in the case of SR-OSPF are automatically enabled if the user enables TI-LFA or RLFA. The computation reuses the SID list of the primary path or

of the TI-LFA or RLFA backup path of the alternate ABR, ASBR, or alternate owner router. The algorithm details are described in Extending MHP LFA coverage with repair tunnels for SR OSPF.

2.1.9.8.1 Extending MHP LFA coverage with repair tunnels for SR OSPF

The following figure shows topology that is used as a reference in this section.

Figure 16: Application of MHP LFA to SR-OSPF tunnel of external prefix



For computing node S, PO_1 is the ABR in the best path (PO_{best}) to reach prefix P. None of the neighbors of node S satisfies the link or node protection inequality of RFC 8518 described in "RFC 8518 multihomed prefix LFA for OSPF" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide*. Therefore, the main aspect of the extension to the RFC 8518 algorithm is for node S to find the best repair tunnel using a PQ node or a P-Q set, which forwards the packet to an alternate exit ABR or ASBR represented by node PO_1, PO_2, or PO_3 in Figure 16: Application of MHP LFA to SR-OSPF tunnel of external prefix.

Note: The same calculation is applied to intra-area /32 anycast prefixes and in that case POi nodes represent the multiple owner routers of the prefix.

The following are the steps of this algorithm.

1. Compute a multihomed LFA repair tunnel for prefix P using each POi.
   a. Node S first attempts to compute a MHP LFA repair tunnel path that matches one ECMP primary path to a POi and that avoids neighbor node E. In other words, the repair tunnel uses POi as a PQ node. Node S further restricts the set of ECMP paths to those over an outgoing interface that satisfy any LFA policy applied to link S-E. Specifically Node S:
      - excludes paths that do not satisfy the admin-group or SRLG constraint in the LFA policy of the primary next hop to E of prefix P
      - applies preference of IP next hops versus tunneled next hops (IGP shortcuts), in accordance with the configuration of the LFA policy and prefers tunneled next hops terminating on the POi node, regardless the protection level

- prefers the LFA next hop not sharing the same pseudo-node (PN) as the primary next hop
- applies preference of node protection versus link protection as per the configuration of the LFA policy
- applies the admin-group preference configured in the LFA policy
- selects the next hops with the lowest IGP cost to the destination prefix P
- selects the tunnel closest (lowest IGP cost) to the destination among equal cost tunnel next hops
- selects the LFA neighbor with the lowest router ID among equal cost tunneled or IP next hops
- selects the lowest tunnel ID or interface ID among next hops to the same LFA neighbor

See LFA solution across IGP area or instance boundary using SR repair tunnel in SR-OSPF for more information about the algorithm interaction with the LFA policy feature.

b. If no path is found in step (1.a), node S computes a MHP LFA repair-tunnel path that matches the node-protect or link-protect LFA, TI-LFA, or RLFA backup path of node POi. In this case, the MHP LFA repair tunnel effectively uses a PQ node or a P-Q set to force the packet to exit the local area at the selected POi.

Note:
If all ECMP candidate paths in step (1.a) are excluded by applying the LFA policy of link S-E, no LFA, TI-LFA, or RLFA backup path of node POi is found in this step because ECMP and LFA are mutually exclusive per prefix.

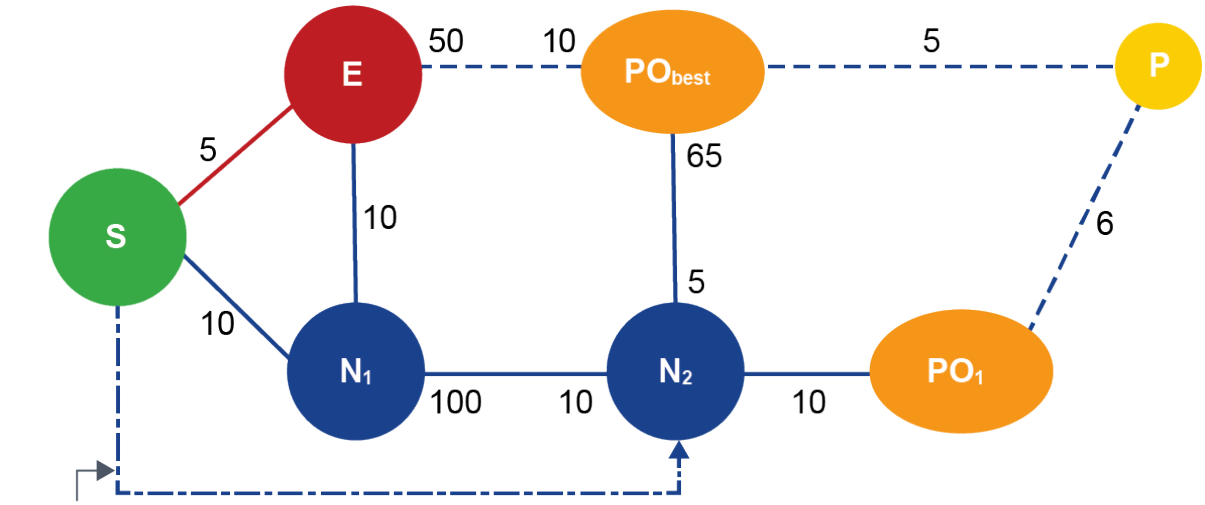
c. If no candidate path is found in steps (1.a) and (1.b), POi is not a candidate alternate ABR, alternate ASBR, or alternate owner router.

2. Create an ordered list of candidate MHP LFA tunnel paths with the following preference order (from highest to lowest).
   a. Prefer the candidate path that uses a POi with the next hop of the primary path, avoiding neighbor node E. Candidate paths are split into two subsets and paths computed from step 1.a preferred over paths computed from step 1.b.
   b. Within each subset, prefer the candidate path that uses POi with lower total cost to prefix P expressed as Min{D_opt(S,POi) + cost(POi, P)}.
   c. If the cost is the same, prefer the candidate path that uses a POi with the lower label stack size.
   d. If the label stack size is the same, prefer the candidate path that uses a POi with the lower router ID.
3. Analyze the ordered list and select the first MHP LFA tunnel path with a segment list size that does not exceed either the value of 1 if RLFA is enabled but TI-LFA is disabled, or the value of the **loopfree-alternate ti-lfa max-sr-frr-labels** command if TI-LFA is enabled.
4. Program in datapath the segment list of the selected MHP LFA repair tunnel for the specific prefix P. The segment list consists of pushing on top of the SID of destination prefix P the SID of the PQ node or the SIDs of the P-Q set.

2.1.9.8.2 Example application of MHP LFA with repair tunnel

The following figure shows topology that is used as a reference in this section.

Figure 17: Application of MHP LFA with repair tunnel to SR-OSPF tunnel of external or anycast prefix




```
-------------------------------------------------------------------+
Prefix                                                             |
Sid-Type      Fwd-Type                                             |
              Next Hop(s)     Out-Label(s)  Interface/Tunnel-ID    |
-------------------------------------------------------------------+
10.20.1.1
Node          Terminating
10.20.1.2
Node          Orig/Transit
              1.1.2.2         19020         to_Dut-B
            (B)1.1.3.3        19020         to_Dut-C
10.20.1.3
Node          Orig/Transit
              1.1.3.3         19030         to_Dut-C
            (B)1.1.2.2        19030         to_Dut-B
10.20.1.4
Node          Orig/Transit
              10.20.1.4       19040             1(RSVP)
10.20.1.5
Node          Orig/Transit
              1.1.3.3         19050         to_Dut-C
            (B)1.1.2.2        19050         to_Dut-B
10.20.1.6
Node          Orig/Transit
              10.20.1.4       19060             1(RSVP)
6.6.6.6
Node          Orig/Transit
              1.1.3.3         19160         to_Dut-C
            (B)10.20.1.4      19060             1(RSVP)
                              19160
```

[S] dut-A node SID 10
[N_1] dut-B node SID 20
[E] dut-C node SID 30
[N_2] dut-D node SID 40
[PO_{best}] dut-E node SID 20
[PO_1] dut-F node SID 60
P] 6.6.6.6/32 anycast SID 160

sw1257

Node S is connected to nodes E and N_1 using IP links, and to node N_2 using an IGP shortcut (RSVP-TE LSP).

Prefix P (6.6.6.6/32) is either:

- an external prefix with prefix SID re-advertised by ASBR nodes PO_{best} and PO_1 and with best path through PO_{best}
  or
- an anycast prefix with prefix SID owned by both routers PO_{best} and PO_1 with best path from node S is to PO_{best}

An SRGB assigned to the OSPF instance uses an offset label value of 19000. Base LFA, RLFA, TI-LFA, and MHP LFA are all enabled in node S. Node protection is also enabled. MHP LFA is preferred. Therefore, the following commands are enabled:

- **MD-CLI**

```
configure router ospf loopfree-alternate remote-lfa node-protect
configure router ospf loopfree-alternate ti-lfa node-protect
configure router ospf loopfree-alternate multi-homed-prefix preference all
```

- **classic CLI**

```
configure router ospf loopfree-alternates remote-lfa node-protect
configure router ospf loopfree-alternates ti-lfa node-protect
configure router ospf loopfree-alternates multi-homed-prefix preference all
```

The resulting LFA computations in node S for prefix P yield the following backup paths:

- base LFA node-protecting path to PO_{best} and using IGP shortcut to neighbor N_2 as next hop
- RLFA node-protecting path to PO_{best} and transiting through PQ node N_2
- TI-LFA node-protecting path to PO_{best} and transiting through PQ node N_2
- a MHP LFA path using the RFC 8518 node-protecting inequality as described in "RFC 8518 multihomed prefix LFA for OSPF" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide*. This yields the same path as the base LFA, meaning a node-protecting path to PO_{best} and using IGP shortcut to neighbor N2 as the next hop.

  Node S does, however, determine that this path does not satisfy the PO_{best} overlap inequality as described in "Enhancement to RFC 8518 Algorithm for backup path overlap with path to PObest in the local area" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide* and, therefore, attempts an SR repair tunnel computation as the next step.
- MHP LFA path to PO_1 and using IGP shortcut to neighbor N_2 as next hop. This backup path forces the packet to arrive and exit (if P is external prefix) PO_1 by pushing PO_1 node SID with an index value of 60 and label value of 19060.

The MHP LFA repair tunnel is therefore the preferred backup path and is programmed in datapath to protect the primary path of prefix P.

2.1.9.9 LFA solution across IGP area or instance boundary using SR repair tunnel in SR IS-IS and SRv6 IS-IS

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

The feature programs the computed backup path for the following tunnels:

- SR IS-IS node SID tunnels of external /32 IPv4 prefixes and /128 IPv6 prefixes, and node SID tunnels of intra-area /32 IPv4 anycast prefixes and /128 anycast IPv6 prefixes, in both algorithm 0 and flexible-algorithms
- SRv6 IS-IS locator routes and tunnels of external prefixes and of intra-area anycast prefixes of any size, in both algorithm 0 and flexible algorithm numbers

As a result, an SR-TE LSP, an SR-MPLS policy, or an SRv6 policy that uses an SR IS-IS SID or an SRv6 IS-IS SID of those same prefixes in its configured or computed SID list benefits from the multihomed prefix LFA protection.

After the IP next-hop based multihomed prefix LFA is enabled, the extensions to compute a SR-TE repair tunnel for the multihomed prefix LFA in the case of SR IS-IS and SRv6 IS-IS are automatically enabled if the user also enabled TI-LFA or Remote LFA. The computation reuses the SID list of the primary path or of the TI-LFA or Remote LFA backup path of the alternate ABR or ASBR or alternate owner router.

The behavior of this feature is the same as in OSPF. See LFA solution across IGP area or instance boundary using SR repair tunnel in SR-OSPF.

2.1.10 Segment routing datapath support

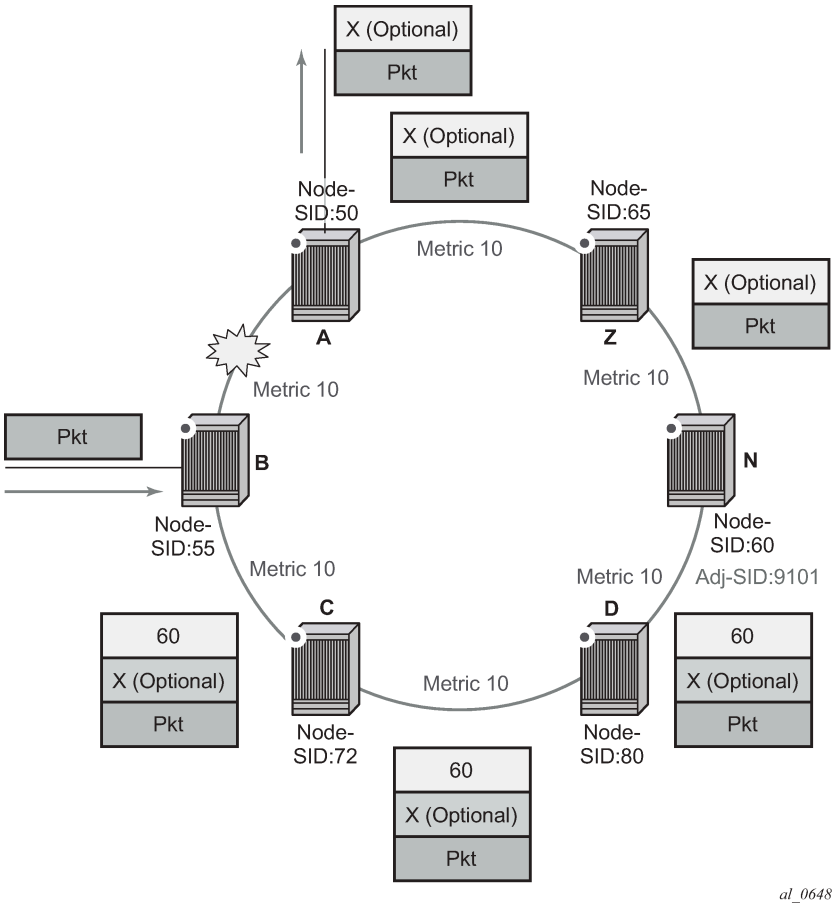
A packet received with a label matching either a node SID or an adjacency SID is forwarded according to the ILM type and operation, as described in the following table.

Table 7: Datapath support

Label type	Operation
Top label is a local node SID	Label is popped and the packet is further processed. If the popped node SID label is the bottom-of-stack label, the IP packet is looked up and forwarded in the appropriate FIB.
Top or next label is a remote node SID	Label is swapped to the calculated label value for the next hop and forwarded according to the primary or backup NHLFE. With ECMP, a maximum of 32 primary next-hops (NHLFEs) are programmed for the same destination prefix and for each IGP instance. ECMP and LFA next-hops are mutually exclusive, as in the current implementation.
Top or next label is an adjacency SID	Label is popped and the packet is forwarded from the interface to the next-hop associated with this adjacency SID label. This datapath operation is modeled like a swap to an implicit-null label instead of a pop.
Next label is BGP 3107 label	The packet is further processed according to the ILM operation, as in the current implementation. • The BGP label may be popped and the packet looked up in the appropriate FIB. • The BGP label may be swapped to another BGP label. • The BGP label may be stitched to an LDP label.
Next label is a service label	The packet is looked up and forwarded in the Layer 2 or VPRN FIB, as in the current implementation.

A router forwarding an IP or a service packet over a segment routing tunnel pushes a maximum of two transport labels with a remote LFA next hop, as shown in the following figure.

Figure 18: Transport label stack in shortest path forwarding with segment routing



Assume that a VPRN service in node B forwards a packet received on a SAP to a destination VPN-IPv4 prefix X advertised by a remote PE2 via ABR/ASBR node A. Router B is in a segment routing domain while PE2 is in an LDP domain. BGP labeled routes are used to distribute the PE /32 loopbacks between the two domains.

When node B forwards over the primary next hop for prefix X, it pushes the node SID of the ASBR followed by the BGP 8277 label for PE2, followed by the service label for prefix X. When the remote LFA next hop is activated, node B pushes one or more segment routing label: the node SID for the remote LFA backup node (node N).

When node N receives the packet while the remote LFA next hop is activated, it pops the top segment routing label that corresponds to a local node SID. This results in popping this label and forwarding of the packet to the ASBR node over the shortest path (link N-Z).

When the ABR/ASBR node receives the packet from either node B or node Z, it pops the segment routing label that corresponds to a local node SID, then swaps the BGP label and pushes the LDP label of PE2, which is the next hop of the BGP labeled route.

2.1.10.1 Hash label and entropy label support

When the **hash-label** option is enabled in a service context, hash label is always inserted at the bottom of the stack as per RFC 6391.

The LSR adds the capability to check a maximum of 16 labels in a stack. The LSR is able to hash on the IP headers when the payload below the label stack of maximum size of 16 is IPv4 or IPv6, including when a MAC header precedes it (**eth-encap-ip** option).

The Entropy Label (EL) feature, as specified in RFC 6790, is supported on RSVP, LDP, segment-routed, and BGP transport tunnels. It uses the Entropy Label Indicator (ELI) to indicate the presence of the entropy label in the label stack. The ELI, followed by the actual entropy label, is inserted immediately below the transport label for which entropy label feature is enabled. If multiple transport tunnels have the entropy label feature enabled, the ELI/EL is inserted below the lowest transport label in the stack.

The LSR hashing operates as follows:

- If the **lbl-only** hashing option is enabled, or if one of the other LSR hashing options is enabled but a IPv4 or IPv6 header is not detected below the bottom of the label stack, the LSR hashes on the EL only.
- If the **lbl-ip** option is enabled, the LSR hashes on the EL and the IP headers.
- If the **ip-only** or **eth-encap-ip** is enabled, the LSR hashes on the IP headers only.

For more information about the Hash Label and Entropy Label features, see the "MPLS Entropy Label and Hash Label" section of the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide*.

2.1.10.2 TTL or hop-limit field handling

The user can configure the TTL or hop-limit propagation for all Segment Routing MPLS (SR-MPLS) tunnels carrying IPv4 or IPv6 packets using the following CLI commands:

```
configure router ttl-propagate sr-mpls-local
configure router ttl-propagate sr-mpls-transit
```

This applies to IPv4 and IPv6 packets of IGP, BGP unlabeled (except 6PE), and static routes in the base router whose next hop is resolved to an SR-MPLS tunnel of any of the following types:

- SR-ISIS
- SR-OSPF
- SR-OSPF3
- SR-TE
- LSP
- SR policy

By default handling, the IP TTL or hop limit is propagated to all labels in the segment routing transport label stack.

The user can configure the TTL or hop-limit propagation separately for CPM-originated IP packets and for transit IP packets. Transit IP packets are packets of base router prefixes received on an access or network interface (with or without tunnel encapsulation), and whose lookup in the FIB results in forwarding them over an SR-MPLS tunnel.

More information about configuring the TTL or hop-limit propagation in other service or routing contexts is available as follows:

- See "Configuration of TTL propagation for VPRN routes" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 3 Services Guide: IES and VPRN* for information about IPv4 and IPv6 packets forwarded in a VPRN or Layer 3 EVPN service context and resolved to an MPLS tunnel, including an SR-MPLS tunnel.
- See “Configuration of TTL propagation for BGP labeled routes” in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide* for information about IPv4 and IPv6 packets of routes in the base router resolved to a BGP-LU or a 6PE tunnel, which itself resolves to an MPLS tunnel, including an SR-MPLS tunnel.

2.1.11 BGP shortcuts using segment routing tunnels

The user enables the resolution of IPv4 prefixes using SR tunnels to BGP next hops in the TTM by configuring the following command:

```
config>router>bgp>next-hop-resolution
        – shortcut-tunnel
            – [no] family {ipv4}
                – resolution {any | disabled | filter}
                – resolution-filter
                    – [no] sr-isis
                    – [no] sr-ospf
                – [no] disallow-igp
                – exit
            – exit
        – exit
```

When **resolution** is set to **any**, any supported tunnel type in the BGP shortcut context is selected according to the TTM preference. The following tunnel types are supported in a BGP shortcut context in order of preference: RSVP, LDP, segment routing, and BGP.

When the **sr-isis** or **sr-ospf** command is enabled, an SR tunnel to the BGP next hop is selected in the TTM from the lowest preference IS-IS or OSPF instance. If many instances have the same lowest preference, the selection of the SR tunnel to the BGP next hop favors the lowest numbered IS-IS or OSPF instance.

See "BGP" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide* for more information.

2.1.12 BGP labeled route resolution using segment routing tunnels

The user enables the resolution of RFC 8277 BGP labeled route prefixes using SR tunnels to BGP next hops in the TTM by configuring the following commands:

```
config>router>bgp>next-hop-resolution
        – labeled-routes
            – transport-tunnel
                – [no] family {label-ipv4 | label-ipv6 | vpn}
                    – resolution {any | disabled | filter}
                    – resolution-filter
                        – [no] sr-isis
                        – [no] sr-ospf
```

```
                    – exit
                – exit
            – exit
        – exit
```

If **resolution** is set to **disabled**, the default binding to LDP tunnels is used. If **resolution** is set to **any**, any supported tunnel type in the BGP labeled route context is selected according to the TTM preference.

The following tunnel types are supported in a BGP labeled route context and are listed in order of preference:

1. RSVP
2. LDP
3. segment routing

When either **sr-isis** or **sr-ospf** is specified using the **resolution-filter** option, a tunnel to the BGP next hop is selected in the TTM from the lowest numbered IS-IS or OSPF instance.

See "BGP" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide* for more information.

2.1.13 Service packet forwarding with segment routing

Two SDP subtypes of the MPLS type allow service binding to a segment routing tunnel programmed in the TTM by IS-IS or OSPF:

- **config**>**service**>**sdp**>**sr-isis**
- **config**>**service**>**sdp**>**sr-ospf**

An SDP configured as **sr-isis** or **sr-ospf** can be used with the **far-end** option. When the **sr-isis** or **sr-ospf** command is enabled, a tunnel to the far-end address is selected in the TTM from the lowest preference IS-IS or OSPF instance. The SR IS-IS or SR-OSPF tunnel is selected at the time of the binding, according to the tunnel selection rules. If a more preferred tunnel is subsequently added to the TTM, the SDP does not automatically switch to the new tunnel until the next time the SDP is being re-resolved.

The **tunnel-far-end** option is not supported. In addition, the **mixed-lsp-mode** option does not support the **sr-isis** and **sr-ospf** tunnel types.

The signaling protocol for the service labels for an SDP using a segment routing tunnel can be configured to static (**off**), T-LDP (**tldp**), or BGP (**bgp**).

SR tunnels can be used in VPRN and BGP EVPN with the **auto-bind-tunnel** command. See "Next-hop resolution" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide* for more information.

Both VPN-IPv4 and VPN-IPv6 (6VPE) are supported in a VPRN or BGP EVPN service using segment routing transport tunnels with the **auto-bind-tunnel** command.

See "BGP" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide* and the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 3 Services Guide: IES and VPRN* for more information about the VPRN **auto-bind-tunnel** CLI command.

2.1.14 Mirror services and Lawful Intercept

The user can configure a spoke-SDP bound to an SR tunnel to forward mirrored packets from a mirror source to a remote mirror destination. In the configuration of the mirror destination service at the destination node, the **remote-source** command must use a spoke-sdp with VC-ID which matches the one the user configured in the mirror destination service at the mirror source node. The far-end option is not supported with an SR tunnel.

This also applies to the configuration of the mirror destination for an LI source.

Configuration at mirror source node:

```
config mirror mirror-dest 10
        – no spoke-sdp sdp-id:vc-id
        – spoke-sdp sdp-id:vc-id [create]
            – egress
                – vc-label egress-vc-label
```

Note:

- *sdp-id* matches an SDP which uses an SR tunnel
- for vc-label, both static and t-ldp egress vc labels are supported

Configuration at mirror destination node:

```
*A:7950 XRS-20# configure mirror mirror-dest 10 remote-source
        – spoke-sdp <SDP-ID>:<VC-ID> create <-- VC-ID matching that of spoke-sdp configured in
 mirror destination context at mirror source node.
            – ingress
                – vc-label <ingress-vc-label> <--- optional: both static and t-ldp ingress vc
 label are supported.
            – exit
            – no shutdown
        – exit
    – exit
```

Note:

- the **far-end** command is not supported with SR tunnel at mirror destination node; user must reference a spoke-SDP using a segment routing SDP coming from mirror source node:
  - **far-end** *ip-address* [**vc-id** *vc-id*] [**ing-svc-label** *ingress-vc-label* | **tldp**] [**icb**]
  - **no far-end** *ip-address*
- for vc-label, both static and t-ldp ingress vc labels are supported

Mirroring and LI are also supported with the PW redundancy feature when the endpoint spoke-sdp, including the ICB, is using an SR tunnel. Routable Lawful Intercept Encapsulation (**config>mirror>mirror-dest>encap# layer-3-encap**) when the remote L3 destination is reachable over an SR tunnel is also supported.

2.1.15 Class-based forwarding for SR-ISIS over RSVP-TE LSPs

To enable CBF+ECMP for SR-ISIS over RSVP-TE:

- Configure the resolution of SR over RSVP-TE LSPs as IGP shortcuts.
- Configure class based forwarding parameters in the MPLS context (a class forwarding policy, forwarding classes to sets associations, and RSVP-TE LSPs to forwarding sets associations).
- Enable class forwarding in the segment routing context.

When SR-ISIS resolves to an ECMP set of RSVP-TE LSPs and class forwarding is enabled in the segment routing context, the following behaviors apply:

- If no LSP in the full ECMP set, has been assigned with a class forwarding policy configuration, the set is considered as inconsistent from a CBF perspective. The system programs, in the forwarding path, the whole ECMP set and regular ECMP spraying occurs over the full set.
- If the ECMP set refers to more than one class forwarding policy, the set is inconsistent from a CBF perspective. The system programs, in the forwarding path, the whole ECMP set without any CBF information, and regular ECMP spraying occurs over the full set.
- In all other cases the ECMP set is considered consistent from a CBF perspective and the following rules apply:
  - If there is no default set (either user-defined or implicit) referenced in a CBF-consistent ECMP set, the system automatically selects one set as the default one. The selected set is the non-empty one with the lowest ID amongst those referenced by the LSPs of the ECMP set.
  - The system programs the data-path such that a packet which has been classified to a particular forwarding class is forwarded using the LSPs associated with the forwarding set which itself is associated with that forwarding class. In the event where the forwarding set is composed of multiple LSPs, the system performs ECMP over these LSPs.
  - Forwarding classes which are either not explicitly mapped to a set or which are mapped to a set for which all LSPs are down are forwarded using the default-set. The system re-elects a default set in cases where all the LSPs of the current default-set become inactive. The system also adapts (updates data-path programming) to configuration or state changes.
  - The CBF capability is available with any system profile. The number of sets is limited to four with system profile None or A, and to six with system profile B.

2.1.16 Segment routing traffic statistics

This section describes capabilities and procedures applicable to IS-IS, OSPFv2, and OSPFv3.

SR OS can enable and collect SID traffic statistics on the ingress and egress datapaths. Statistics can also be shown, monitored, and cleared, as well as accessed using telemetry.

IS-IS and OSPFv2 support Node SID, Adjacency SID, and Adjacency Set statistics. OSPFv3 supports Node SID and Adjacency SID statistics. The following commands are used to enter the context that allows for configuring the types of SIDs for which to collect traffic statistics:

- **configure router isis segment-routing egress-statistics**
- **configure router ospf segment-routing egress-statistics**
- **configure router ospf3 segment-routing egress-statistics**
- **configure router isis segment-routing ingress-statistics**
- **configure router ospf segment-routing ingress-statistics**
- **configure router ospf3 segment-routing ingress-statistics**

By default, statistics collection is disabled on all types of SIDs. If statistics are disabled after having been enabled, the statistics indexes that were allocated are released and the counter values are cleared.

On ingress, depending on which types of SIDs have statistics enabled, the system allocates a statistic index to each programmed ILM, corresponding to the following:

- the local node SID (including backup node SID) and the local adjacency SIDs (including adjacencies advertised as set members)
- the received node SID advertisements

On egress, depending on which types of SIDs have statistics enabled, the following apply:

- The system allocates a statistic index shared by the programmed NHLFEs (primary, and backup if any) corresponding to the local Adjacency SIDs and to the received Adjacency SIDs advertisements, and a statistic index shared by the primary NHLFEs (as many as members) of each adjacency set.
- The system allocates a statistic index shared by the programmed NHLFEs (one or more primaries, and backup if any) corresponding to each of the received node SID advertisements.

Note: The statistic indexes constitute a finite resource. The system may not be able to allocate as many indexes as needed. In this case, the system issues a notification and automatically retries to allocate statistic indexes, but does not issue further notifications in case it still fails to allocate the needed statistic indexes. If the system successfully allocates all the required statistic indexes to IGP SIDs, then a second notification is issued to inform the user. A state variable records whether a SID has an index allocated.

Note: The allocation of statistic indexes is non-deterministic. If more statistic indexes are required system-wide, for example, upon a reboot, the system may not be able to re-allocate the statistic indexes to the same entities as before the reboot.

2.1.17 Microloop avoidance using loop-free SR tunnels for IS-IS

Transient forwarding loops, or microloops, occur during IGP convergence as a result of the transient inconsistency among forwarding states of the nodes of the network. The microloop avoidance feature supports the use of loop-free SR paths and a configurable time as a solution to avoid microloops in SR IS-IS SID tunnels.

2.1.17.1 Configuring microloop avoidance

The following command enables the microloop avoidance feature within each IGP instance:

```
configure router isis segment-routing micro-loop-avoidance [fib-delay fib-delay]
```

Configure the *fib-delay* timer to a value that corresponds to the worst-case IGP convergence in a network domain. The default value of 1.5 seconds (1500 ms) corresponds to a network with a nominal convergence time.

When this feature is disabled using the following command, any active FIB delay timers are forced to expire immediately and the new next hops are programmed for all impacted node SIDs. The feature is disabled for the next SPF runs.

- **MD-CLI**

```
configure router isis segment-routing delete micro-loop-avoidance
```

- **classic CLI**

```
configure router isis segment-routing no micro-loop-avoidance
```

When this feature is enabled, the following scenarios apply:

- IS-IS MT=0 for an SR IS-IS IPv4/IPv6 tunnel (node SID)
- IPv4 and IPv6 SR-TE LSPs that use a node SID in their segment list
- IPv4 and IPv6 SR policy that use a node SID in their segment list

2.1.17.2 Microloop avoidance algorithm process

The SR OS microloop avoidance algorithm provides a loop-free mechanism in accordance with IETF *draft-bashandy-rtgwg-segment-routing-uloop*. The algorithm supports a single event on a P2P link or broadcast link with two neighbors only for the following cases:

- link addition or restoration
- link removal or failure
- link metric change

Using the algorithm, the router applies the following microloop avoidance process.

1. After it receives the topology updates and before the new SPF is started, the router verifies that the update corresponds to a single link event. Updates for the two directions of the link are treated as a single link event.

   If two or more link events are detected, the microloop avoidance procedure is aborted for this SPF and the existing behavior is maintained.

   Note: The microloop avoidance procedure is aborted if the subsequent link event received by an ABR is from a different area than the one that triggered the event initially. However, if the received event comes from a different IGP instance, the ABR handles it independently and triggers the microloop avoidance procedure, as long as it is a single event in that IGP instance.

2. The main SPF and LFA SPF algorithms (base LFA, remote LFA or TI-LFA, based on the user configuration in that IGP instance) are run.
3. No action is performed for a node or a prefix if the SPF calculation has resulted in no change to its next hops and metrics.
4. No action is performed for a node or a prefix if the SPF has resulted in a change to its next-hops or metrics, or both, and the new next hops are resolved over RSVP-TE LSPs used as IGP shortcuts.

   Note: Nokia strongly recommends enabling CSPF for the RSVP-TE LSP used in IGP shortcut application. This avoids IGP churn and ensures microloop avoidance in the path of the RSVP control plane messages which would otherwise be generated following the convergence of IGP because the next hop in the ERO is looked up in the routing table.

5. The route is marked as microloop avoidance eligible for a node or a prefix if the SPF has resulted in a change to its next hops or metrics. The router performs the following:
   - For each SR node SID that uses a microloop avoidance eligible route with ECMP next hops, the router activates the common set of next hops between the previous and new SPF.
   - For each SR node SID that uses a microloop avoidance eligible route with a single next hop, the router computes and activates a loop-free SR tunnel applicable to the specific link event.

     This tunnel acts as the microloop avoidance primary path for the route and uses the same outgoing interface as the newly computed primary next hop.

     See Microloop avoidance for link addition, restoration, or metric decrease and Microloop avoidance for link removal, failure, or metric increase for more information.
   - The router programs the TI-LFA, base LFA, or remote LFA backup path that protects the new primary next hop of the node SID.
6. The **fib-delay** timer is started to delay the programming of the new main and LFA SPF results into the FIB.
7. After the **fib-delay** expires, the new primary next hops are programmed for node SID routes that are marked as eligible for the microloop avoidance procedure.

Note:
If a new SPF is scheduled while the **fib-delay** timer is running, the timer is forced to expire and the entire procedure is aborted.

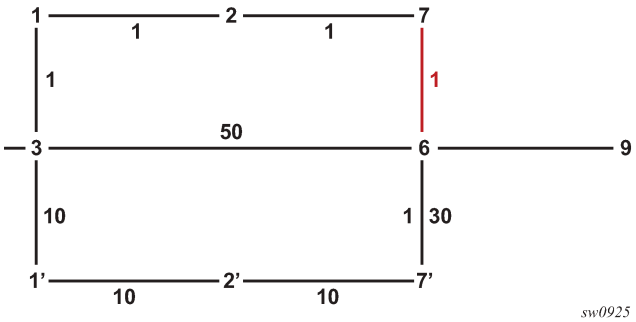
If a CPM switchover is triggered while the **fib-delay** timer is running, the timer is forced to expire and the entire procedure is aborted.

In both cases, the next hops from the most recently run SPF are programmed for all impacted node SIDs. A subsequent event restarts the procedure at step 1.

2.1.17.3 Microloop avoidance for link addition, restoration, or metric decrease

The following figure shows an example of link addition or restoration in a network topology.

Figure 19: Microloop avoidance in link addition or restoration



The microloop avoidance algorithm performs the following steps in the preceding network topology example.

1. Link 7-6 is added to the topology.
2. Router 3 detects a single link addition between remote nodes 7 and 6.

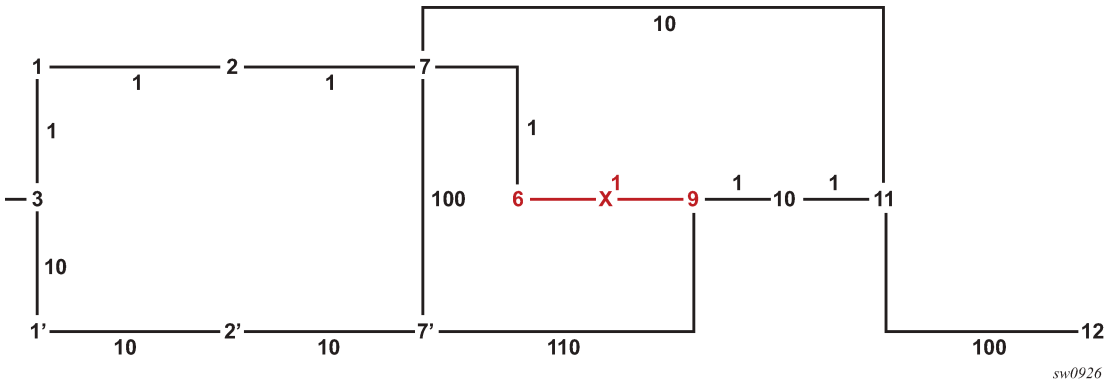
3. Router 3 runs the main and LFA SPFs.
   - All nodes downstream of the added link in Dijkstra tree (in this case, nodes 6 and 9) see a next-hop change.
   - All nodes upstream of the added link (in this case, nodes 1, 2, and 7) see no route change.
   - Nodes 1', 2', and 7' are not using node 6 or 7 as parent nodes and are not impacted by the link addition event.
4. For all nodes downstream from the added link, the algorithm computes and activates an SR tunnel that forces traffic to remote endpoint 6 of the added link.

   The algorithm pushes node SID 7 and adjacency SID of link 7-6 in the SR IS-IS tunnel for these nodes.
5. The use of the adjacency SID of link 7-6 skips the FIB state on node 7 and traffic to all nodes downstream of 6 are not impacted by microloop convergence.
6. The same method applies to metric decrease of link 7-6 that causes traffic to be attracted to that link.

2.1.17.4 Microloop avoidance for link removal, failure, or metric increase

The following figure depicts an example of link removal or failure in a network topology.

Figure 20: Microloop avoidance in link removal or failure



The microloop avoidance algorithm performs the following steps in the preceding network topology example.

1. Link 6-9 is removed or fails.
2. Router 3 detects a single link event and runs main and LFA SPFs.
   - All nodes downstream of the removed link in the Dijkstra tree (in this case, nodes 9, 10, 11, and 12) see a next-hop change.
   - Nodes 10, 11, and 12 are no longer downstream of node 9.
   - All nodes upstream of the removed link (in this case, nodes 1, 2, 7, and 6) see no route change.
   - Nodes 1', 2', and 7' are not using node 6 or 9 as parent nodes and are not impacted by the link removal event.
3. For each impacted node, the algorithm computes and activates a loop-free SR tunnel to the farthest node in the shortest path that did not see a next hop change, then uses the adjacency SIDs to reach destination node.

- For the SR IS-IS tunnel of node 12, push the SID of node 7, then the SIDs of adjacencies 7-11 and 11-12.
- This loop-free SR tunnel computation is similar to the P-Q set calculation in TI-LFA (see Topology-independent LFA), but the P node is defined as the farthest node in the shortest path to the destination in the new topology with no next-hop change.
- The maximum number of labels used for the P-Q set is determined as follows:
  - If TI-LFA is enabled, use the configured value of the **max-sr-frr-labels** parameter.
  - If TI-LFA is disabled, use the value of 3, which matches the maximum value of TI-LFA **max-sr-frr-labels** parameter.

  In both cases, this value is passed to MPLS for checking against the **max-sr-labels** [**additional-frr-labels**] parameter for all configured SR-TE LSPs and SR-TE LSP templates.
- The path to the P node may travel over an RSVP-TE LSP used as an IGP shortcut. In this case, the RSVP-TE LSP must have CSPF enabled. This is to avoid churn in IGP and to avoid microloops in the path of the RSVP control plane messages that are generated following the convergence of IGP, because the next hop in the ERO is looked up in the routing table.
- When SR-LDP stitching is enabled and the path to the P node or the path between the P and Q nodes is partly on the LDP domain, no loop-free SR tunnel is programmed and IGP programs the new next hop or hops.

4. The same method (steps 2 and 3) applies to a metric increase of link 6-9 that causes traffic to move away from that link; for example, a metric change from 1 to 200.

2.1.18 Microloop avoidance using loop-free SR tunnels for OSPF

The microloop avoidance feature supports the use of loop-free SR paths and a configurable time to avoid microloops in SR OSPFv2 and SR OSPFv2 flexible algorithms SID tunnels.

Transient forwarding loops, or microloops, occur during IGP convergence as a result of the transient inconsistency among forwarding states of the nodes of the network. Microloops can occur because of a network maintenance action, including the:

- addition of a new network component
- removal of a network component
- modification of a link cost

See section Microloop avoidance using loop-free SR tunnels for IS-IS for more information about the microloop avoidance algorithm.

2.1.18.1 Configuring microloop avoidance

Use the following command to enable microloop avoidance within each IGP instance.

```
configure router ospf segment-routing micro-loop-avoidance fib-delay
```

First enable microloop avoidance in the OSPF segment-routing context using the preceding command, before enabling microloop avoidance for each SR-OSPF flexible algorithm. When microloop avoidance is

enabled for a flexible algorithm, the FIB delay timer for the flexible algorithm is inherited from the OSPF segment-routing context.

Use the following command to enable microloop avoidance for an SR-OSPF flexible algorithm.

```
configure router ospf flexible-algorithms flex-algo micro-loop-avoidance
```

2.1.19 Configuring IS-IS for flexible algorithms for SR-MPLS

IGP protocols traditionally compute best paths over the network based on the IGP metric assigned to the links. Many network deployments use RSVP-TE based or SR-based TE to enforce traffic over a path that is computed using different metrics or constraints than the shortest IGP path. The SR Flexible Algorithm (Flex-Algorithm) solution allows IGPs to compute constraint-based paths over the network. This section describes the use of SR prefix SIDs to compute a constraint topology and send packets along the constraint-based paths.

Using Flex-Algorithms can reduce the number of SR SIDs that must be imposed to send packets along a constrained path; this implementation simplifies the hardware capabilities of the head-end devices of SR routing tunnels.

The supported depth of the label stack is considered in an SR network when SR-TE tunnels or SR policies are deployed. In such tunnel policies, the packet source routing is based on the SR label stack pushed on the packet. The depth of the label stack that a router can push on a packet determines the complexity of the SR-TE tunnel construction that the router can support.

The SR Flex-Algorithm solution allows the creation of composed metrics based upon arbitrary parameters (for example, delay, link administrative group, cost, and so on) when using Flex-Algorithms. A network-wide set of composed topology constraints, known as the Flexible Algorithm Definition (FAD), creates an SR Flex-Algorithm topology. The IGP calculates the best path using constraint-based SPF and the FAD to create the best paths through the Flex-Algorithm topology.

With Flex-Algorithms, each Flex-Algorithm topology can send data flows along the most optimal constrained path toward its destination using a single label, which reduces the imposed label stack along.

Using this solution, backup path calculations (for example, Loop Free Alternate (LFA), Remote LFA (R-LFA) and Topology Independent LFA (TI-LFA)) can be constrained to the SR Flex-Algorithm topology during link failure.

Perform the following tasks to configure Flex-Algorithms using IS-IS.

1. Configuring the flexible algorithm definition
2. Configuring IS-IS Flex-Algorithm participation
3. Configuring IS-IS Flex-Algorithm prefix node SID
4. Verifying basic Flex-Algorithm behavior

2.1.19.1 Configuring the flexible algorithm definition

To guarantee loop-free forwarding for paths that are computed for a specific Flex-Algorithm, all routers configured to participate in that Flex-Algorithm must agree on the FAD. The agreement ensures that routing loops and inconsistent forwarding behavior is avoided.

Each router that is configured to participate in a specific Flex-Algorithm must select the FAD based on standardized tie-breaking rules. This ensures consistent FAD selection in cases where different routers advertise different definitions for a specific Flex-Algorithm. The following tie-breaking rules apply:

- From the FAD advertisements in the area (including both locally generated advertisements and received advertisements), the router selects the one with the highest priority value.
- If there are multiple FAD advertisements with the same priority, the router selects one that originated from the router with the highest system ID.

A router that is not participating in a specific Flex-Algorithm is allowed to advertise the FAD for that specific Flex-Algorithm. Any change in the FAD may result in temporary disruption of traffic that is forwarded based on those Flex-Algorithm paths. The impact is similar to any other event that requires network-wide convergence.

If a node is configured to participate in a Flex-Algorithm but the selected FAD includes a calculation type, metric type, constraint, flag, or sub-TLV that is not supported by the node, the node stops participation and removes any forwarding state associated with the Flex-Algorithm.

Use the following syntax to configure FADs:

```
config>router
        – flexible-algorithm-definitions
            – flex-algo <fad-name> [create]
            – no flex-algo <fad-name>
                – description <description-string>
                – [no] description
                – exclude
                    – admin-group <admin-group>
                    – [no] admin-group <admin-group>
                – flags-tlv
                – [no] flags-tlv
                – include-all
                    – admin-group <admin-group>
                    – [no] admin-group <admin-group>
                – include-any
                    – admin-group <admin-group>
                    – [no] admin-group <admin-group>
                – metric-type {igp | te-metric | delay}
                – [no] metric-type
                – priority <[0..255]>
                – [no] priority
                – shutdown
                – [no] shutdown
```

Example: Configuration output for a basic FAD

```
router
  flexible-algorithm-definitions
    flex-algo "My128" create
      description "This-is-my-algo128"
        metric-type delay
        no shutdown
    exit
  exit
```

2.1.19.2 Configuring IS-IS Flex-Algorithm participation

Up to seven Flex-Algorithms in the range 128 to 255 can be configured for IS-IS. Use the **participate** command to configure a router to participate for the specific algorithm. If a locally configured FAD exists, use the **advertise** command to configure the router to advertise this definition. A router is not required to advertise a configured FAD to participate in a Flex-Algorithm. If a Flex-Algorithm is enabled to participate or advertise the FAD, it is configured and active for all configured IS-IS areas.

Use the following syntax to configure Flex-Algorithms for IS-IS.

```
config>router>isis
        – flexible-algorithms
            – [no] flex-algo flex-algo
                – advertise fad-name
                – no advertise
                – [no] loopfree-alternates
                – [no] participate
            – [no] shutdown
```

Note: When a router participates in Flex-Algorithms, it only advertises support for the Flex-Algorithm where the router can comply with the winning FAD, provided that at least one FAD exists for this algorithm.

Example: Configuration output for Flex-Algorithm participation

```
isis 0
    flexible-algorithms
      flex-algo 128
        advertise "My128"
        participate
      exit
      no shutdown
    exit
```

Example: IS-IS router capability when a FAD is advertised

```
*A:Dut-B# show router isis database Dut-B.00-00 detail level 2
===============================================================================
Rtr Base ISIS Instance 0 Database (detail)
===============================================================================
Displaying Level 2 database
-------------------------------------------------------------------------------
LSP ID    : Dut-B.00-00                                 Level     : L2
Sequence  : 0x94                  Checksum  : 0x4ae0    Lifetime  : 969
Version   : 1                     Pkt Type  : 20        Pkt Ver   : 1
Attributes: L1L2                  Max Area  : 3         Alloc Len : 1492
SYS ID    : 4900.0000.0002        SysID Len : 6         Used Len  : 223
TLVs :
  Supp Protocols:
    Protocols     : IPv4
  IS-Hostname   : Dut-B
  Router ID   :
    Router ID   : 10.20.1.2
Router Cap : 10.20.1.2, D:0, S:0
    TE Node Cap : B E M  P
    SR Cap: IPv4 MPLS-IPv6
       SRGB Base:20000, Range:10001
```

```
    SR Alg: metric based SPF, 128
    Node MSD Cap: BMI : 12 ERLD : 15
    FAD Sub-Tlv:
        Flex-Algorithm   : 128
        Metric-Type      : delay
        Calculation-Type : 0
        Priority         : 100
        Flags: M
```

2.1.19.3 Configuring Advertising Administrative Groups

Using Flex-Algorithms, a Nokia router can advertise and receive both Administrative Groups (AGs) and Extended Administrative Groups (EAGs) as defined in *draft-ietf-lsr-flex-algo-26-Internet-Draft*. Section 12 of *draft-ietf-lsr-flex-algo-26-Internet-Draft* states:

> ASLA Admin Group Advertisements to be used by the Flexible Algorithm application MAY use either the Administrative Group or Extended Administrative Group encodings.
>
> A receiver supporting this specification MUST accept both ASLA Administrative Group and Extended Administrative Group TLVs as defined in [RFC8919] or [RFC8920]. In the case of IS-IS, if the L-Flag is set in ASLA advertisement, as defined in [RFC8919] Section 4.2, then the receiver MUST be able to accept both Administrative Group TLV as defined in [RFC5305] and Extended Administrative Group TLV as defined in [RFC7308].

The user can configure the SR OS to support advanced backward compatibility and interoperability with legacy devices that only support legacy AGs.

Before an AG can be used it must have an affinity bit index associated that represents a Flex-Algorithm link color.

Example: Configuration of an AG link color (MD-CLI)

```
[ex:/configure routing-options]
A:node-2# info
    if-attribute {
        admin-group "blue" {
            value 100
        }
        admin-group "red" {
            value 10
        }
        admin-group "yellow" {
            value 200
        }
    }
```

Example: Configuration of an AG link color (classic CLI)

```
A:node-2>config>router>if-attr# info
----------------------------------------------
            admin-group "blue" value 100
            admin-group "red" value 10
            admin-group "yellow" value 200
----------------------------------------------
```

After creating an AG, the user can assign the AG for Flex-Algorithms to a link. A user can assign a maximum of 32 different AGs to a single link. Only segment routing Flex-Algorithms can use AGs from the extended EAG range.

Example: Configuration of an AG membership on an interface (MD-CLI)

```
[ex:/configure router "Base" interface "test1"]
A:node-2# info
    if-attribute {
        admin-group ["blue"]
    }
```

Example: Configuration of an AG membership on an interface (classic CLI)

```
A:node-2>config>router>if$ info
----------------------------------------------
            if-attribute
                admin-group "blue"
            exit
            no shutdown
----------------------------------------------
```

SR OS receives and processes both legacy AGs and newer EAGs. When SR OS receives conflicting AG and EAG link properties, SR OS respects the rules prescribed in section 2.3 of RFC 7308. For example, if both a legacy AG and EAG are present, a receiving device must use the legacy AG as the first 32 bits (0-31) of an administrative color and use the EAG for bits 32 and higher, if the higher bits are present.

Additionally, SR OS can select what AG or EAG to advertise for optimal backwards compatibility and IGP link-state database optimization. After an AG or EAG is advertised, it can also be exported into a BGP-LS.

Use the following commands to configure EAG per IGP instance. The default is **prefer-ag**.

```
configure router isis flexible-algorithms advertise-admin-group {prefer-ag|eag-only|ag-eag}
configure router ospf flexible-algorithms advertise-admin-group {prefer-ag|eag-only|ag-eag}
```

From the advertising side, when configured, SR OS supports any of the following:

eag-only

This option supports only the advertisement of the EAG TLV.

ag-eag

This option supports advertisement of both the AG and the EAG TLVs at the same time.

pref-ag

This option supports dynamic behavior, where the router IGP instance advertises only the legacy AG TLV when the legacy range is used or advertises the EAG TLVs when an extended range is configured. If no bits are set in the AG range, no AG TLV is sent; only the EAG TLV is sent.

2.1.19.4 Configuring IS-IS Flex-Algorithm prefix node SID

A prefix node SID (IPv4 or IPv6) must be assigned for each participating Flex-Algorithm.

The Flex-Algorithm SIDs are allocated from the label block assigned to SR configuring a special range is not required.

Note:
Flex-Algorithm node SIDs can be configured for IPv4 and IPv6 prefixes.

Use the following syntax to configure the prefix node SIDs for IS-IS Flex-Algorithms:

```
config>router>isis>interface
        – ipv4-node-sid
        – flex-algo
            – ipv4-node-sid index <value>
            – ipv4-node-sid label <value>
            – no ipv4-node-sid
            – ipv6-node-sid index <value>
            – ipv6-node-sid label <value>
            – no ipv6-node-sid
```

Example: Configuration output for Flex-Algorithm prefix node SIDs

```
router
  mpls-labels
    sr-labels start 20000 end 30000
  exit
  interface "Loopback0"
    address 10.20.1.2/32
    loopback
    no shutdown
  exit
  isis 0
    segment-routing
      prefix-sid-range global
      no shutdown
    exit
    interface "Loopback0"
      ipv4-node-sid index 2
      passive
      flex-algo 128
        ipv4-node-sid index 12
      exit
      no shutdown
    exit
```

Example: Level 2 database of an advertised IS-IS

```
A:Dut-B# show router isis database Dut-B.00-00 detail level 2
===============================================================================
Rtr Base ISIS Instance 0 Database (detail)
===============================================================================
Displaying Level 2 database
-------------------------------------------------------------------------------
LSP ID    : Dut-B.00-00                                  Level     : L2
Sequence  : 0x9d                  Checksum  : 0x38e9     Lifetime  : 626
Version   : 1                     Pkt Type  : 20         Pkt Ver   : 1
Attributes: L1L2                  Max Area  : 3          Alloc Len : 1492
SYS ID    : 4900.0000.0002        SysID Len : 6          Used Len  : 223
……<snip>……
  TE IP Reach   :
    Default Metric  : 10
    Control Info:    , prefLen 30
    Prefix   : 10.10.10.0
    Default Metric  : 0
    Control Info:   S, prefLen 32
    Prefix   : 10.20.1.2
```

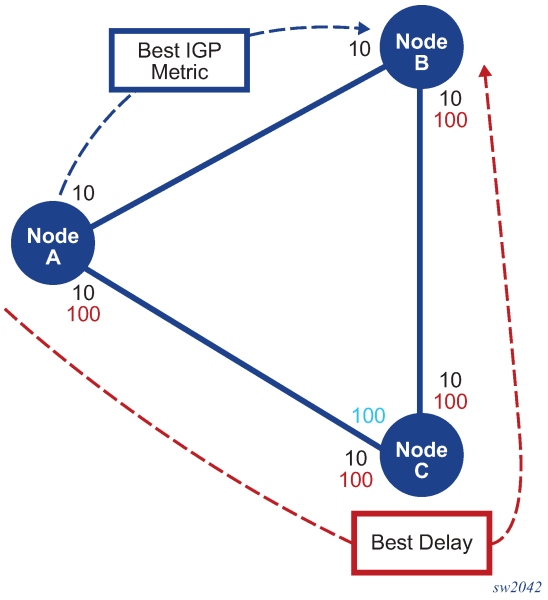
```
    Sub TLV   :
      Prefix-SID Index:2, Algo:0, Flags:NnP
      Prefix-SID Index:12, Algo:128, Flags:NnP
    Default Metric  : 10
    Control Info:    , prefLen 30
    Prefix   : 10.10.10.8
...<snip>...
```

2.1.19.5 Verifying basic Flex-Algorithm behavior

The creation of the segment routing Flex-Algorithm forwarding information results in the label forwarding tables on the router. On a Nokia router, it is possible to look both at the tunnel table and the routing table to understand the Flex-Algorithm path toward a destination prefix.

For example, algorithm 128 has been configured to use the delay metric and consequently forwards traffic using the lowest delay through the network. In the following figure, Node B is configured with IP address 10.20.1.2/32, the A-B path has the best default IGP metric, and the A-C-B path has the best delay.

Figure 21: Selecting the lowest delay path



Example: tunnel-table command output

```
A:Dut-A# show router tunnel-table
===============================================================================
IPv4 Tunnel Table (Router: Base)
===============================================================================
Destination           Owner     Encap TunnelId  Pref   Nexthop        Metric
   Color
-------------------------------------------------------------------------------
10.10.10.2/32         isis (0)  MPLS  524298    11     10.10.10.2     0
10.10.10.6/32         isis (0)  MPLS  524292    11     10.10.10.6     0
10.20.1.2/32          isis (0)  MPLS  524296    11     10.10.10.2     10
10.20.1.2/32          isis (0)  MPLS  524306    11     10.10.10.6     200
10.20.1.3/32          isis (0)  MPLS  524294    11     10.10.10.6     10
10.20.1.3/32          isis (0)  MPLS  524307    11     10.10.10.6     100
```

```
-------------------------------------------------------------------------------
Flags: B = BGP or MPLS backup hop available
       L = Loop-Free Alternate (LFA) hop available
       E = Inactive best-external BGP route
       k = RIB-API or Forwarding Policy backup hop
===============================================================================
A:Dut-A#
```

Example: Detailed tunnel-table command output

```
A:Dut-A# show router tunnel-table 10.20.1.2/32 detail
===============================================================================
Tunnel Table (Router: Base)
===============================================================================
Destination      : 10.20.1.2/32
NextHop          : 10.10.10.2
Tunnel Flags     : entropy-label-capable
Age              : 18h21m35s
CBF Classes      : (Not Specified)
Owner            : isis (0)             Encap            : MPLS
Tunnel ID        : 524296               Preference       : 11
Tunnel Label     : 20002                Tunnel Metric    : 10
Tunnel MTU       : 1560                 Max Label Stack  : 1
-------------------------------------------------------------------------------
Destination      : 10.20.1.2/32
NextHop          : 10.10.10.6
Tunnel Flags     : entropy-label-capable
Age              : 02h01m32s
CBF Classes      : (Not Specified)
Owner            : isis (0)             Encap            : MPLS
Algorithm        : 128
Tunnel ID        : 524306               Preference       : 11
Tunnel Label     : 20012                Tunnel Metric    : 200
Tunnel MTU       : 1560                 Max Label Stack  : 1
-------------------------------------------------------------------------------
Number of tunnel-table entries          : 2
Number of tunnel-table entries with LFA : 0
===============================================================================
A:Dut-A#
```

Example: Route table output with and without the Flex-Algorithm context

```
A:Dut-A# show router isis routes
===============================================================================
Rtr Base ISIS Instance 0 Route Table
===============================================================================
Prefix[Flags]                    Metric     Lvl/Typ     Ver.  SysID/Hostname
  NextHop                                               MT     AdminTag/SID[F]
-------------------------------------------------------------------------------
10.10.10.0/30                    10         1/Int.      65    Dut-A
   0.0.0.0                                                0       0
10.10.10.4/30                    10         1/Int.      42    Dut-A
0.0.0.0                                                 0       0
10.10.10.8/30                    20         2/Int.      65    Dut-B
   10.10.10.2                                             0       0
10.20.1.1/32                     0          1/Int.      42    Dut-A
   0.0.0.0                                                0       0/1[NnP]
10.20.1.2/32                     10         2/Int.      65    Dut-B
   10.10.10.2                                             0       0/2[NnP]
10.20.1.3/32                     10         2/Int.      42    Dut-C
   10.10.10.6                                             0       0/3[NnP]
-------------------------------------------------------------------------------
```

```
No. of Routes: 6 (6 paths)
-------------------------------------------------------------------------------
Flags        : L = LFA nexthop available
SID[F]       : R  = Re-advertisement
               N  = Node-SID
               nP = no penultimate hop POP
               E  = Explicit-Null
               V  = Prefix-SID carries a value
               L  = value/index has local significance
===============================================================================
A:Dut-A#
A:Dut-A# show router isis routes flex-algo 128
===============================================================================
Rtr Base ISIS Instance 0 Flex-Algo 128 Route Table
===============================================================================
Prefix[Flags]                    Metric     Lvl/Typ     Ver.  SysID/Hostname
  NextHop                                                MT     AdminTag/SID[F]
-------------------------------------------------------------------------------
10.20.1.2/32                     200        2/Int.      82    Dut-C
   10.10.10.6                                              0       0/12[NnP]
10.20.1.3/32                     100        2/Int.      82    Dut-C
   10.10.10.6                                              0       0/13[NnP]
-------------------------------------------------------------------------------
No. of Routes: 2 (2 paths)
-------------------------------------------------------------------------------
Flags        : L = LFA nexthop available
SID[F]       : R  = Re-advertisement
               N  = Node-SID
               nP = no penultimate hop POP
               E  = Explicit-Null
               V  = Prefix-SID carries a value
               L  = value/index has local significance
===============================================================================
A:Dut-A#
```

Example: Detailed route table output, with and without the Flex-Algorithm context

```
A:Dut-A# show router isis routes 10.20.1.2 detail
===============================================================================
Rtr Base ISIS Instance 0 Route Table (detail)
===============================================================================
Prefix           : 10.20.1.2/32
Status           : Active               Level              : 2
NextHop          : 10.10.10.2
Metric           : 10                   Type               : Internal
SPF Version      : 65                   SysID/Hostname     : Dut-B
MT               : 0                    AdminTag           : 0
SID              : 2                    SID-Flags          : NnP
-------------------------------------------------------------------------------
No. of Routes: 1 (1 path)
-------------------------------------------------------------------------------
SID[F]       : R  = Re-advertisement
               N  = Node-SID
               nP = no penultimate hop POP
               E  = Explicit-Null
               V  = Prefix-SID carries a value
               L  = value/index has local significance
===============================================================================
A:Dut-A#

A:Dut-A# show router isis routes 10.20.1.2 flex-algo 128 detail
===============================================================================
Rtr Base ISIS Instance 0 Flex-Algo 128 Route Table (detail)
```

```
===============================================================================
Prefix           : 10.20.1.2/32
Status           : Active               Level             : 2
NextHop          : 10.10.10.6
Metric           : 200                  Type              : Internal
SPF Version      : 82                   SysID/Hostname    : Dut-C
MT               : 0                    AdminTag          : 0
SID              : 12                   SID-Flags         : NnP
-------------------------------------------------------------------------------
No. of Routes: 1 (1 path)
-------------------------------------------------------------------------------
SID[F]       : R  = Re-advertisement
               N  = Node-SID
               nP = no penultimate hop POP
               E  = Explicit-Null
               V  = Prefix-SID carries a value
               L  = value/index has local significance
===============================================================================
A:Dut-A#
```

2.1.19.6 Configuration and usage considerations for Flex-Algorithms

Take the following considerations into account when configuring and using Flex-Algorithms:

- IS-IS algorithms 128 to 255 can program only the tunnel table, while IS-IS for algorithm 0 can program both the tunnel and the IP routing tables. For operational simplicity, the **show router isis routes** command displays the correct egress interface.
- The user can only configure a maximum of 256 FADs on a router to prevent the accidental creation of an overload of local FADs.
- A router can participate in a maximum of seven Flex-Algorithms. Each algorithm has the capability to advertise a single locally configured FAD.
- The SR OS implementation assumes that the participation of a specific Flex-Algorithm is valid for all IGP areas. For example, if a user enables FAD advertisement for an IS-IS instance with Level 1 and Level 2 capability, the algorithm is enabled and the FAD is advertised in both levels.
- All nodes participating in a Flex-Algorithm configuration must advertise the locally used FADs when configured and optionally advertise node participation when the winning FAD is supported.
- The winning FAD on a router is selected based on the following tiebreaker:

  1. Select the FAD with the highest priority.
  2. Select the FAD advertised by the highest IGP system ID.

- If redundant, the local router does not support the winning FAD (because it is locally advertised or it is advertised by a remote router). The router should remove itself from the Flex-Algorithm topology by not advertising algorithm participation in the IS-IS router TLV capability. In such a case, no path delay is computed and any prefix SID of that Flex-Algorithm is removed from the associated routing and tunnel tables.
- When the FAD selects a metric type, only links that include the same metric type in their attributes are considered for the **flex-algo** topology.
- Leaking of a FAD on an ABR is not supported.
- When advertising the FAD flags TLV, the SR OS router always sets the M-flag, which forces the IS-IS routers to use Flex-Algorithm-aware metrics for inter-area routing. The enforced M-flag ensures that the best ABR, according to the Flex-Algorithm, is selected to exit the local IGP area. Without the M-flag,

the wrong ABR may be selected and cause routing loops or a traffic blackhole. This handling assumes that an ABR must advertise the IS-IS Flex-Algorithm prefix metric sub-TLV when leaking prefixes and associated SIDs.

Advertising the flags TLV is optional. Use the following command to disable the flags TLV configuration within the Flex-Algorithm definition.

– **MD-CLI**

```
configure routing-options flexible-algorithm-definitions flex-algo flags-tlv false
```

– **classic CLI**

```
configure router flexible-algorithm-definitions flex-algo no flags-tlv
```

- SR OS supports the AGs as defined in RFC 5305 and EAGs as defined in RFC 7308 with the following considerations:
  – Up to 32 extended AGs (also known as link colors) can be used on a single link.
  – SR OS provides the following AG and EAG support for Flex-Algorithms:

    Note: Nokia recommends you avoid using AGs for both Flex-Algorithm and LFA policies.

    - SR OS does not support EAG bits beyond the first 32 octets. If the router receives a FAD with the EAG TLV length greater than 32 octets and the bits set to 1 beyond the first 32 bits, the router blocks the FAD.
    - For backward compatibility, vendors may use only the first 32 colors in the EAG.
    - If EAG is used to add a color on the links, the link attribute size can be 4 (or a multiple of 4) octets long.
    - The EAG for Flex-Algorithms is forwarded for appropriate ASLA encoding in accordance with RFC 8919, *IS-IS Application-Specific Link Attributes*.
    - When an EAG ASLA link attribute is received, the SR OS router handles it as follows:
      – SR OS provides limited EAG support and only parses EAGs that are 4 octets long. The EAG represents a traditional 4-octet AG to support backward compatibility.
      – SR OS treats the ASLA-encoded EAG as opaque information when the EAG size is a multiple of 4 octets long (that is, 4, 8, and so on).
      – Because of limited EAG support, a new trap is not sent if the AG and EAG link attributes are inconsistent. In such a case, the AG attributes are used in accordance with RFC 7308.
    - The receipt of a Flex-Algorithm FAD that contains an include or exclude EAG ASLA link attribute is handled as follows:
      – If the SR OS router receives a FAD where the AG TLV length is 4 octets, the FAD can be used for **flex-algo** and it is treated as an AG.
      – If the SR OS router receives a FAD where the AG TLV length is greater than 4 octets and bits are set to 1 in the first four octets only (the remaining bits are set to 0), the FAD participates assuming that the AGs have been configured as a result of EAG backward compatibility.
      – If the SR OS router receives a FAD where the length of the AG TLV is greater than 4 octets and has bits set to 1 beyond the first 32 bits, the router blocks this FAD. SR OS does not support EAG bits beyond the first 32 bits.

- Flex-Algorithm uses the IS-IS Min/Max Unidirectional Link Delay sub-TLV as defined in RFC 8570. This delay is set through the static configuration.
- SR OS allows the user to enable and disable Flex-Algorithm LFA paths. The LFA type is inherited from the algorithm 0 base topology configuration.
- Users can protect links and nodes using the LFA fast-convergence technology. If the primary path is constrained by a specific Flex-Algorithm topology, the LFA SPF calculation is executed within the Flex-Algorithm topology. This calculation identifies the correct LFA, R-LFA or TI-LFA bounded by this topology. Consequently, the constraints of a specific Flex-Algorithm topology are respected even during failure scenarios:
  - Enabling or disabling the Flex-Algorithm-dependent LFA, R-LFA, or TI-LFA is aligned with enabling the LFA within the router **flex-algo** context.
  - A new configuration node LFA is added in the IGP command options within the Flex-Algorithm configuration and this node configuration shows if the commands are administratively enabled or disabled.
  - The LFA command option allows the user to disable or enable loop-free alternates for this Flex-Algorithm. The RLFA and TLFA command options are inherited from algorithm 0.
  - The Flex-Algorithm LFA exclude policy configuration is copied from the **flex-algo** 0 configuration.
  - The Flex-Algorithm-aware LFA may cause additional resource consumption (for example, in memory and in CPU).
  - SR OS Flex-Algorithm support for LFA policies supported by algorithm 0, including SRLG, protection type, exclude and include groups.

Take the following considerations into account for the following situations:

- **Interaction with SR-LDP mapping server**

  Flex-Algorithms are not compatible with the SR-LDP mapping server. SR OS only supports the mapping server TLV with algorithm 0.

- **Interaction with SR-TE policy**

  Flex-Algorithms have no impact on how SR-TE LSPs are used. Applications that support the use of SR-TE LSPs continue to be supported. All SR-TE resolution mechanisms are supported.

  Flex-Algorithm support results in the following SR-TE changes:

  - When an SR-TE path is constructed through manual router configuration or received from the PCE, the sequence of SR-TE SIDs may include one or more Flex-Algorithm prefix node SIDs.
  - At the SR-TE head-end router, the sequenced SR-TE label stack (the sequence of SIDs) is imposed upon the payload and the packet is forwarded using the NHLFE from the top label or SID.

  Validity of a specific SR-TE LSP is the same as without Flex-Algorithm support.

- **Interaction with SR policies**

  Similar to SR-TE LSPs, SR policies are only influenced by Flex-Algorithms because of the construction of the segment list. The segment list may be constructed using one or more Flex-Algorithm prefix node label SIDs. All applications capable of using SR policies have opaque awareness if a segment list is constructed using Flex-Algorithm labels or SIDs.

- **Flex-Algorithm and adjacency SID protection**

  During the FRR process, local repair of the links to reach the Q-node from the P-node is determined by the sub-topology defined by the Flex-Algorithm. Therefore, the used link will include the correct AGs and so on.

However, the adjacency SID backup is based upon algorithm 0 because adjacency SIDs are not advertised using a Flex-Algorithm. Consequently, there is a risk to violate the Flex-Algorithm if the related link breaks while it is in use as backup for a Flex-Algorithm path. This Flex-Algorithm SLA break can be avoided when adjacency SIDs are configured with no backup capability.

- **Duplicate SID handling**

  IS-IS uses the first learned remote SID and generates a trap for duplicate entries.

- **Interaction with IGP shortcut and forwarding adjacency features**

  To select the optimal shortest path within a constrained topology, Flex-Algorithm paths are carefully crafted using the constraints specified in the FAD. If the constrained topology includes logical RSVP-TE links that conceal FAD constraints, the Flex-Algorithm may incorrectly send traffic over out-of-profile physical links.

  Flex-Algorithms do not support shortcut tunnels that hide physical link properties; the following features are not supported:

  – SR-LDP stitching

  – IGP shortcut

  – forwarding adjacency; forwarding adjacencies are not considered in the **flex-algo** topology.

- **Relationship between Flex-Algorithm and algorithm 0 configuration**

  A configured router with Flex-Algorithm does not have to advertise an algorithm 0 SID.

- **Interaction of Flex-Algorithm-aware nodes and FAD flags TLV**

  When Flex-Algorithms are enabled, SR OS advertises FAD flags TLV in IGP to signal the mandatory use of Flex-Algorithm-aware performance metrics for optimal SPF calculation. For correct Flex-Algorithm operation, it is expected that Flex-Algorithm-aware nodes support FAD flags TLV interpretation.

  For improved interoperability, when defining a Flex-Algorithm on SR OS use the following command to stop advertising the FAD flags TLV:

  – **MD-CLI**

  ```
  configure routing-options flexible-algorithm-definitions flex-algo flags-tlv false
  ```

  – **classic CLI**

  ```
  configure router flexible-algorithm-definitions flex-algo no flags-tlv
  ```

- **Flex-Algorithm for BGP services**

  For BGP, BGP LU, and VPN, use the import policy action **flex-algo** to automatically resolve BGP next hop over an IGP Flex-Algorithms topology.

  The BGP EVPN service does not support Flex-Algorithms aware autobind for BGP next-hop.

- **Flex-Algorithm and TLV encoding**

  Flex-Algorithms BGP-LS export and TLV encoding is supported.

2.1.20 OSPFv2 configuration for flexible algorithms for SR-MPLS

Basic flexible algorithms configuration tasks for an IGP are described in Configuring IS-IS for flexible algorithms for SR-MPLS.

Perform the following tasks to configure Flex-Algorithms using OSPFv2.

1. Configuring FAD for OSPFv2
2. Configuring OSPFv2 Flex-Algorithm participation
3. Configuring OSPFv2 Flex-Algorithm prefix node SID
4. Verifying basic Flex-Algorithm behavior for OSPFv2

2.1.20.1 Configuring FAD for OSPFv2

Configuration of OSPFv2 FAD is identical to the IS-IS configuration; see Configuring the flexible algorithm definition for more information.

2.1.20.2 Configuring OSPFv2 Flex-Algorithm participation

Up to seven flexible algorithms in the range 128 to 255 can be configured for OSPFv2. Use the **participate** command to configure a router to participate in the specific algorithm. If a locally configured FAD exists, use the **advertise** command to advertise the definition. A router is not required to advertise a configured FAD to participate in a Flex-Algorithm.

If a router with Flex-Algorithms is enabled to participate and enabled to advertise the FAD, the Flex-Algorithms are configured and active for all configured OSPFv2 areas, and the FAD is advertised in all OSPFv2 areas.

Use the following syntax to configure Flex-Algorithms for OSPFv2:

```
config>router>ospf
              +--flexible-algorithms
                 +--[no] flex-algo flex-algo-id
                    +--advertise fad-name
                    +--no advertise
                    +-- [no] loopfree-alternates
                    +-- [no] participate
                 +-- [no] shutdown
```

Note: When a router participates in Flex-Algorithms, it only advertises support for the Flex-Algorithm where the router can comply with the winning FAD, provided that at least one FAD exists for this algorithm.

Example: Configuration output for Flex-Algorithm participation

```
ospf 0
  flexible-algorithms
    flex-algo 128
    advertise "My128"
    participate
```

```
  exit
  no shutdown
exit
```

2.1.20.3 Configuring OSPFv2 Advertising Administrative Groups

Configuration of OSPFv2 Advertising AGs is identical to the IS-IS configuration; see Configuring Advertising Administrative Groups for more information.

2.1.20.4 Configuring OSPFv2 Flex-Algorithm prefix node SID

An IPv4 prefix node SID must be assigned for each participating Flex-Algorithm.

The Flex-Algorithm SIDs are allocated from the label block assigned to SR and configuring a special range for Flex-Algorithms is not required.

Use the following syntax to configure the prefix node SIDs for OSPFv2 Flex-Algorithms:

```
config>router>ospf>area>interface
                      +--node-sid
                      +--flex-algo flex-algo-id
                         +--node-sid index <[0..4294967295]>
                         +--node-sid label <[1..4294967295]>
                         +--no node-sid
```

Example: Configuration output for Flex-Algorithm prefix node SIDs

```
router
  mpls-labels
    sr-labels start 20000 end 30000
  exit
  interface "Loopback0"
    address 10.20.1.2/32
    loopback
    no shutdown
  exit
  ospf 0
    segment-routing
      prefix-sid-range global
      no shutdown
    exit
    area 0.0.0.0
      interface "Loopback0"
        node-sid index 2
        flex-algo 128
          node-sid index 12
          exit
          no shutdown
      exit
```

2.1.20.5 Verifying basic Flex-Algorithm behavior for OSPFv2

The basic Flex-Algorithm behavior verification is identical to the information provided for IS-IS in Verifying basic Flex-Algorithm behavior.

2.1.20.6 Configuration and usage considerations for Flex-Algorithms for OSPFv2

The considerations described in Configuration and usage considerations for Flex-Algorithms apply to OSPFv2 Flex-Algorithms.

The following configuration and usage considerations are specific to OSPFv2:

- OSPFv2 Flex-Algorithm supports IPv4 only.
- Virtual links are not supported for OSPFv2 Flex-Algorithms.
- Leaking of prefix SIDs and Flex-Algorithm-aware SIDs is not supported between OSPFv2 instances.
- When enabled, the OSPFv2 Flex-Algorithm is activated for all areas configured within the OSPFv2 routing instance.

2.1.21 Configuring BGP-based services for flexible algorithms

BGP-based network services (VPRN, EVPN and VPLS) can be automatically steered to a flexible algorithm using service-based import policies. To configure BGP automated steering, a policy-statement must first be defined. Within this policy-statement, all BGP criteria is identified to steer traffic towards certain prefixes towards a flexible algorithm topology. Often the BGP color community is used to identify which flex-algorithm is used for a BGP prefix, however, Nokia SR OS has the capability to match upon any existing policy-statement BGP attribute criteria.

The following example shows policy-statement configuration.

Example: MD-CLI

```
[ex: /configure router policy-options]
A:admin@node-2# info
policy-options {
        policy-statement "ExamplePolicy" {
            entry 10 {
                from {
                    color 128
                }
                action {
                    action-type accept
                    flex-algo 128
                }
            }
            default-action {
                action-type accept
                flex-algo 128
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config>router>policy-options# info
----------------------------------------------
            policy-statement "ExamplePolicy"
                entry 10
                    from
                        color 128
                    exit
                    action accept
                        flex-algo 128
                    exit
                exit
                default-action accept
                    flex-algo 128
                exit
            exit
```

After a policy-statement is created it can be applied to the BGP service using service-based import policies. BGP-based automated flexible algorithm steering for SR-MPLS based segment routing can be applied for VPRN, EVP and VPLS services.

This example provides the various places a policy-statement can be applied to initiate BGP-based automated steering to a flexible algorithm:

Example: MD-CLI

```
[ex: /configure service]
A:admin@node-2# info
vpn-apply-import true
    ebgp-default-reject-policy {
        import false
        export false
    }
    import {
        policy ["ExamplePolicy"]
    }
    next-hop-resolution {
        shortcut-tunnel {
            family ipv4 {
                allow-flex-algo-fallback true
            }
        }
        labeled-routes {
            transport-tunnel {
                family vpn {
                    allow-flex-algo-fallback true
                }
                family label-ipv4 {
                    allow-flex-algo-fallback true
                }
                family label-ipv6 {
                    allow-flex-algo-fallback true
                }
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config>router>bgp# info
----------------------------------------------
            vpn-apply-import
            import "ExamplePolicy"
            next-hop-resolution
                shortcut-tunnel
                    family ipv4
                        allow-flex-algo-fallback
                        resolution disabled
                    exit
                exit
                labeled-routes
                    transport-tunnel
                        family vpn
                            resolution filter
                            allow-flex-algo-fallback
                        exit
                        family label-ipv4
                            resolution filter
                            allow-flex-algo-fallback
                        exit
                        family label-ipv6
                            resolution filter
                            allow-flex-algo-fallback
                        exit
                    exit
                exit
            exit
            no shutdown
```

Example: MD-CLI

```
[ex: /configure service]
A:admin@node-2# info
    epipe "3" {
        customer "1"
        bgp 1 {
            vsi-import ["ExamplePolicy"]
        }
    }
    vpls "2" {
        customer "1"
        bgp 1 {
            vsi-import ["ExamplePolicy"]
        }
    }
    vprn "1" {
        customer "1"
        bgp-ipvpn {
            mpls {
                vrf-import {
                    policy ["ExamplePolicy"]
                }
                auto-bind-tunnel {
                    allow-flex-algo-fallback true
                }
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config>service# info
----------------------------------------------
        customer 1 name "1" create
            description "Default customer"
        exit
        vprn 1 name "1" customer 1 create
            shutdown
            bgp-ipvpn
                mpls
                    shutdown
                    auto-bind-tunnel
                        allow-flex-algo-fallback
                    exit
                    vrf-import "ExamplePolicy"
                exit
            exit
        exit
        vpls 2 name "2" customer 1 create
            shutdown
            bgp
                vsi-import "ExamplePolicy"
            exit
            stp
                shutdown
            exit
        exit
        epipe 3 name "3" customer 1 create
            shutdown
            bgp
                vsi-import "ExamplePolicy"
            exit
        exit
```

One of the side effects of using a flex-algorithm import policy-statement is packets are dropped when no igp-shortcut exists to the BGP next-hop. This behavior ensures that committed Service Level Agreements (SLA) are kept when flexible algorithms are used. However, for some use-cases, dropping packets may be too strict of a behavior, and therefore Nokia SR OS allows through the configuration of the **allow-flex-algo-fallback** command a relaxation of the requirement for a matching flexible algorithm igp-shortcut. When the **allow-flex-algo-fallback** command is configured and no matching shortcut exists, an igp-shortcut from algorithm uses a lower preferred alternative, and therefore, may be breaking strictly committed SLAs.

2.2 Establishing segment routing TE LSPs

When segment routing is used together with MPLS data plane, the SID is a standard MPLS label. A router forwarding a packet using segment routing therefore pushes one or more MPLS labels.

Segment routing using MPLS labels can be used in both shortest path routing applications (see Segment routing in shortest path forwarding for more information) and in traffic engineering (TE) applications, as described in this section.

An SR-TE LSP supports a primary path, with FRR backup, and one or more secondary paths. A secondary path can be configured as standby.

SR OS implements the following computation methods for the paths of an SR-TE LSP:

- **hop-to-label translation**

The TE-DB converts the list of hops, destination of the LSP, and the strict or loose hops in the path definition to a list of SIDs by searching the IGP instances with segment routing enabled. This method does not support TE constraints except for loose or strict hops.

See SR-TE LSP path computation using hop-to-label translation for more information.

- **local CSPF**

  The LSP path TE constraints are considered in the path computation. This method implements most of the CSPF capabilities supported with RSVP-TE LSP with very few exceptions, such as the bandwidth constraint which cannot be reserved with SR-TE LSP because of the lack of a signaling protocol to establish the LSP path.

  See SR-TE LSP path computation using local CSPF for more details.

- **Path Computation Element (PCE)**

  The router acting as a PCE client (PCC) requests a computation of the path of an SR-TE LSP from the PCE using the PCEP.

  See Path Computation Element Protocol (PCEP) for more details.

- **user-specified SID list**

  The SR-LSP feature provides the option for the user to manually configure each path of the LSP using an explicit list of SID values.

  See SR-TE LSP paths using explicit SIDs for more details.

The configured or computed path of an SR-TE LSP can use a combination of node SIDs and adjacency SIDs.

2.2.1 SR-TE MPLS support

The following MPLS commands and nodes are supported in Segment Routing-Traffic Engineering (SR-TE):

- global MPLS-level commands and nodes

  interface, **lsp**, **path**, **shutdown**

- LSP-level commands and nodes

  bfd, **bgp-shortcut**, **bgp-transport-tunnel**, **cspf, exclude**, **hop-limit**, **igp-shortcut, include**, **metric**, **metric-type**, **path-computation-method**, **primary**, **retry-limit**, **retry-timer**, **revert-timer**, **shutdown**, **to**, **from**, **vprn-auto-bind**

- Both primary and secondary paths are supported with an SR-TE LSP. The following primary path level commands and nodes are supported with SR-TE LSP:

  bandwidth, **bfd**, **exclude**, **hop-limit**, **include**, **priority**, **shutdown**

  The following secondary path level commands and nodes are supported with SR-TE LSP:

  bandwidth, **bfd**, **exclude**, **hop-limit**, **include**, **path-preference**, **priority**, **shutdown**, **srlg**, **standby**

The following MPLS commands and nodes are not supported with SR-TE LSP:

- global MPLS-level commands and nodes (configuration ignored)

  admin-group-frr, **auto-bandwidth-multipliers**, **auto-lsp**, **bypass-resignal-timer**, **cspf-on-loose-hop**, **dynamic-bypass**, **exponential-backoff-retry**, **frr-object**, **hold-timer**, **ingress-statistics**, **least-fill-min-thd**, **least-fill-reoptim-thd**, **logger-event-bundling**, **lsp-init-retry-timeout**, **lsp-template**,

max-bypass-associations, **mbb-prefer-current-hops**, **mpls-tp**, **p2mp-resignal-timer**, **p2mp-s2l-fast-retry**, **p2p-active-path-fast-retry**, **retry-on-igp-overload**, **secondary-fast-retry-timer**, **shortcut-local-ttl-propagate**, **shortcut-transit-ttl-propagate**, **srlg-database**, **srlg-frr**, **static-lsp**, **static-lsp-fast-retry**, **user-srlg-db**

- LSP-level commands and nodes not supported (configuration blocked)

  adaptive, **adspec**, **auto-bandwidth**, **class-type**, **dest-global-id**, **dest-tunnel-number**, **exclude-node**, **fast-reroute**, **ldp-over-rsvp**, **least-fill**, **main-ct-retry-limit, p2mp-id, primary-p2mp-instance, propagate-admin-group, protect-tp-path, rsvp-resv-style, working-tp-path**

- primary path level commands and nodes not supported (configuration blocked)

  adaptive, **backup-class-type**, **class-type**, **record**, **record-label**

- secondary path level commands and nodes not supported (configuration blocked)

  adaptive, **class-type**, **record**, **record-label**

The user can associate an empty path or a path with strict or loose explicit hops with SR-TE LSP paths using the **hop**, **primary**, and **secondary** CLI commands.

A hop that corresponds to an adjacency SID must be identified with its far-end host IP address (next hop) on the subnet. If the local end host IP address is provided, this hop is ignored because this router can have multiple adjacencies (next hops) on the same subnet.

A hop that corresponds to a node SID is identified by the prefix address.

See SR-TE LSP instantiation for information about processing the user-configured path hops.

2.2.2 SR-TE LSP instantiation

If an SR-TE LSP is configured on the router, its path can be computed by the router or by an external TE controller, which is referred to as a PCE. This feature works with the Nokia stateful PCE, which is part of the NSP. SR OS supports the following modes of operation, configurable on a per-SR-TE LSP basis.

- **PCC-initiated and PCC-controlled SR-TE LSP**

  In this mode of operation, the path of the LSP is computed by the router acting as a PCE Client (PCC).

  A PCC-initiated and PCC-controlled SR-TE LSP has the following characteristics:

  – can contain strict or loose hops, or a combination of both

  – supports both a basic hop-to-label translation and a full CSPF as a path computation method

  – capability exists to report an SR-TE LSP to synchronize the LSP database of a stateful PCE server using the **pce-report** command, but the PCE cannot update the LSP path; the PCC maintains control of the LSP

- **PCC-initiated and PCE-computed SR-TE LSP**

  In this mode of operation, the path of the LSP is computed by the PCE at the request of the PCC.

  A PCC-initiated and PCE-computed SR-TE LSP supports the Passive Stateful Mode, which enables the **path-computation-method pce** option for the SR-TE LSP. This allows the PCE to perform path computation only at the request of the PCC; the PCC retains control.

  The capability exists to report an SR-TE LSP to synchronize the LSP database of a stateful PCE server using the **pce-report** command.

- **PCC-initiated and PCE-controlled SR-TE LSP**

In this mode of operation, the path of the LSP is computed and updated by the PCE following a delegation from the PCC.

A PCC-initiated and PCE-controlled SR-TE LSP allows Active Stateful Mode, which enables the **pce-control** command for the SR-TE LSP. This allows the PCE to perform path computation and updates following a network event without the explicit request from the PCC; the PCC delegates full control.

The user can configure the path computation requests only (PCE-computed) or both path computation requests and path updates (PCE-controlled) to PCE for a specific LSP using the **path-computation-method pce** option and **pce-control** CLI command.

The **path-computation-method pce** option sends the path computation request to the PCE instead of the local CSPF. If this command is enabled, the PCE acts in Passive Stateful mode for this LSP and performs path computations for the LSP only at the request of the router. This command configures the router to use the PCE-specific path computation algorithm instead of the local router CSPF algorithm.

The default value is **no path-computation-method**.

The user can also enable the router's full CSPF path computation method. See SR-TE LSP path computation using local CSPF for more details.

The **pce-control** command allows the router to delegate full control of the LSP to the PCE (PCE-controlled). If this command is enabled, the PCE acts in Active Stateful mode for this LSP, which allows PCE to reroute the path following a failure or to reoptimize the path and update the router without requiring the router to request these actions.

Note:

- The user can delegate LSPs computed by either the local CSPF or the hop-to-label translation path computation methods.
- The user can delegate LSPs that have the **path-computation-method pce** option enabled or disabled. The LSP maintains its latest active path computed by the PCE or the router at the time it was delegated. The PCE only makes an update to the path at the next network event or reoptimization. The default value is **no pce-control**.
- A PCE report is supported for SR-TE LSPs with more than one path. However, PCE computation and PCE control are not supported in such cases. PCE computation and PCE control are supported for SR-TE LSPs with only one path that is either primary or secondary.

In all cases, the PCC LSP database is synchronized with the PCE LSP database using the PCEP Path Computation State Report (PCRpt) message for LSPs that have the **pce-report** command enabled.

The global MPLS-level **pce-report** command can be used to enable or disable PCE reporting for all SR-TE LSPs for the purpose of LSP database synchronization. This configuration is inherited by all LSPs of the specified type. The PCC reports both CSPF and non-CSPF LSP. The default value (**no pce-report**) controls the introduction of the PCE into an existing network and allows the operator to decide if all LSP types need to be reported.

The LSP-level **pce-report** command overrides the global configuration for reporting LSPs to the PCE. The default value is to inherit the global MPLS-level value. The **inherit** value returns the LSP to inherit the global configuration for that LSP type.

Note:
If PCE reporting is disabled for the LSP, either because of inheritance or because of LSP-level configuration, enabling the **pce-control** command for the LSP has no effect. To help

troubleshoot this situation, both the **pce-report** and **pce-control** operational values are added to the output of the LSP **show** commands.

For more information about configuring PCC-Initiated and PCC-Controlled LSPs, see Configuring PCC-controlled, PCE-computed, and PCE-controlled SR-TE LSPs.

2.2.2.1 PCC-initiated and PCC-controlled LSPs

In this mode of operation, the user configures the LSP name and the primary and (optionally) secondary path name with the path information in the referenced path name, entering a full or partial explicit path with all or some hops to the destination of the LSP. Each hop is specified as an address of a node or an address of the next hop of a TE link. Optionally, each hop may be specified as a SID value corresponding to the MPLS label to use on a specific hop. In this case, the whole path must consist of SIDs.

To configure a primary or secondary path to always use a specific link whenever it is up, the strict hop must be entered as an address corresponding to the next hop of an adjacency SID, or the path must consist of SID values for every hop. If the strict hop corresponds to a loopback address, it is translated into an adjacency SID, and therefore does not guarantee that the same specific TE link is chosen.

To use an SR-TE path that consists of unprotected adjacency SIDs, each hop of the path must be configured as a strict hop with the address matching the next hop of the adjacency SID. Protection on each of these adjacencies must be disabled. See SR-TE LSP path computation for more information.

MPLS assigns a tunnel ID to the SR-TE LSP and a path ID to each new instantiation of the primary path, as in an RSVP-TE LSP. These IDs are useful for representing the MBB path of the same SR-TE LSP, which need to coexist during the update of the primary path.

Note:
The concept of MBB is used broadly in the context of an SR-TE LSP because there is no signaling involved, so the new path information immediately overrides the older one.

The router retains full control of the path of the LSP. CSPF is not supported and, as a result, the full or partially explicit path is instantiated as-is and no other constraint (such as SRLG, **admin-group**, hop count, or bandwidth) is checked. Only the LSP path label stack size is checked by MPLS against the maximum value configured for the LSP after the TE database (TE-DB) hop-to-label translation returns the label stack. See SR-TE LSP path computation for more information about this check.

The ingress LER performs the following steps to resolve the user-entered path before programming it in the datapath.

1. MPLS passes the path information to the TE-DB, which uses the hop-to-label translation or the full CSPF method to convert the list of hops into a label stack. The TE database returns the actual selected hop SIDs plus labels as well the configured path hop addresses which were used as the input for this conversion.
2. The ingress LER validates the first hop of the path to determine the next hop and the associated egress interfaces and programs the datapath according to the following conditions.
   - If the first hop corresponds to an adjacency SID (host address of the next hop on the subnet of the link), the adjacency SID label is not pushed. The ingress LER treats forwarding to a local interface as a push of an implicit-null label.
   - If the first hop is a node SID of a downstream router, the node SID label is pushed.

   In both cases, the SR-TE LSP tracks and rides the SR shortest path tunnel of the SID of the first hop.

3. In the case where the router is configured as a PCC and has a PCEP session to a PCE, the router sends a PCRpt message to update PCE with the state of UP and the RRO object for each LSP where the **pce-report** command is enabled. The PE router does not set the delegation control flag to keep LSP control. The state of the LSP is now synchronized between the router and the PCE.

2.2.2.1.1 Guidelines for PCC-initiated and PCC-controlled LSPs

The router supports both a full CSPF and a basic hop-to-label translation path computation methods for a SR-TE LSP. In addition, the user can configure a path for the SR-TE LSP by explicitly entering SID label values.

The ingress LER has a few ways to detect a path is down or is not optimal and take immediate action:

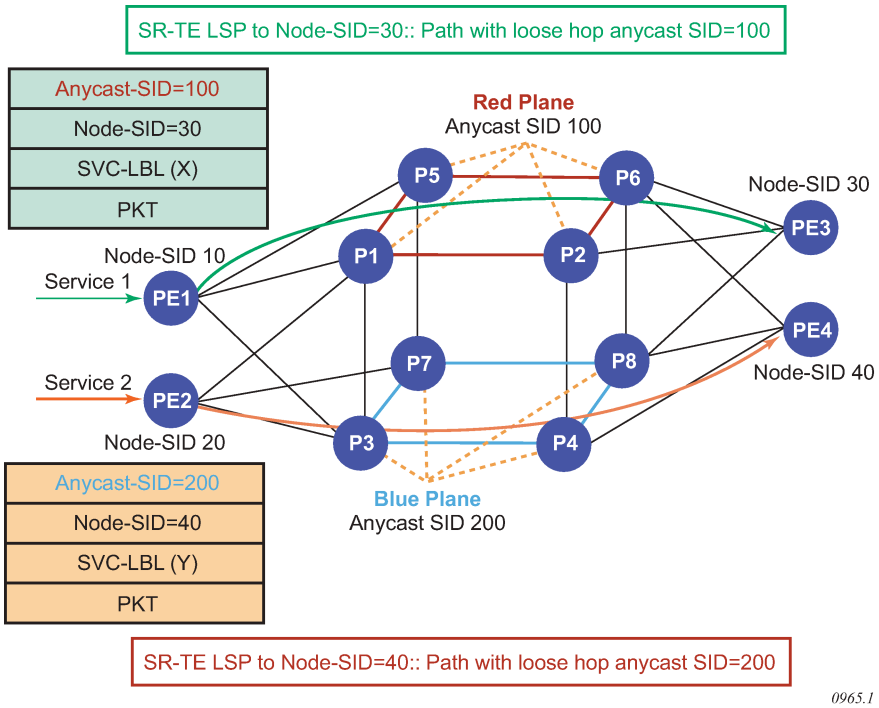
- Failure of the top SID detected via a local failure or an IGP network event. In this case, the LSP path goes down and MPLS retries it.
- Timeout of the seamless BFD session when enabled on the LSP and the **failure-action** is set to the value of **failover-or-down**. In this case, the path goes down and MPLS retries it.
- Receipt of an IGP link event in the TE database. In this case, MPLS performs an ad-hoc re-optimization of the paths of all SR-TE LSPs if the user enabled the MPLS level command **sr-te-resignal resignal-on-igp-event**. This capability only works when the path computation method is the local CSPF. It allows the ingress LER not only to detect a single remote failure event which causes packets to drop but also a network event which causes a node SID to reroute and therefore forwarding packets on a potentially sub-optimal TE path.
- Performing a manual or timer based resignal of the SR-TE LSP. This applies only when the path computation method is the local CSPF. In this case, MPLS re-optimizes the paths of all SR-TE LSPs.

With both the hop-to-labeled path computation method and the user configured SID labels, the ingress LER does not monitor network events which affect the reachability of the adjacency SID or node SID used in the label stack of the LSP, except for the top SID. As a result, the label stack may not be updated to reflect changes in the path except when seamless BFD is used to detect failure of the path. It is therefore recommended to use this type of SR-TE LSP in the following configurations only:

- empty path
- path with a single node-SID loose-hop
- path of an LSP to a directly-connected router (single-hop LSP) with an adjacency-SID or a node-SID loose/strict hop
- strict path with hops consisting of adjacencies explicitly configured in the path as IP addresses or SID labels.

The user can also configure a SR-TE LSP with a single loose-hop using the anycast SID concept to provide LSR node protection within a plane of the network TE topology. This is illustrated in Figure 22: Multiplane TE with node protection. The user configures all LSRs in a plane with the same loopback interface address, which must be different from that of the system interface and the router ID of the router, and assigns them the same node-SID index value. All routers must use the same SRGB.

Figure 22: Multiplane TE with node protection



Then user configures in a LER a SR-TE LSP to some destination and adds to its path either a loose-hop matching the anycast loopback address or the explicit label value of the anycast SID. The SR-TE LSP to any destination hops over the closest of the LSRs owning the anycast SID because the resolution of the node-SID for that anycast loopback address uses the closest router. When that router fails, the resolution is updated to the next closest router owning the anycast SID without changing the label stack of the SR-TE LSP.

2.2.2.2 PCC-initiated and PCE-computed or -controlled LSP

In this mode of operation, the ingress LER uses PCEP to communicate with a PCE-based external TE controller (called the PCE). Acting as the PCC, the router instantiates a PCEP session to the PCE.

The following PCE control modes are supported:

- **passive control mode**

  In this mode, the user enables the **path-computation-method pce** option for one or more SR-TE LSPs. A PCE performs path computations at the request of the PCC.

- **active control mode**

  In this mode, the user enables the **pce-control** command for an LSP, which allows the PCE to perform both path computation and periodic reoptimization of the LSP path without an explicit request from the PCC.

For the PCC to communicate with a PCE about the management of the path of an SR-TE LSP, the router implements the extensions to PCEP in support of segment routing. This feature works with the Nokia stateful PCE, which is part of the NSP.

The following sequence describes configuring and programming a PCC-initiated SR-TE LSP when passive or active control is assigned to the PCE.

Note:
The subsequent steps are followed when the user performs a **no shutdown** command on a PCE-controlled or PCE-computed LSP. The starting point is an administratively down LSP with no active paths.

1. The SR-TE LSP configuration is created on the PE router using the CLI or OSS/SAM.

   The configuration dictates the PCE control mode: active (**pce-control** command enabled) or passive (**path-computation-method pce** enabled and **pce-control** disabled).
2. The PCC assigns a unique PLSP-ID to the LSP. The PLSP-ID identifies the LSP on a PCEP session and must remain constant during its lifetime. The PCC on the router tracks the association of {PLSP-ID, SRP-ID} to {Tunnel-ID, Path-ID} and uses the latter to communicate with MPLS about a specific path of the LSP.
3. The PE router does not validate the entered path. While the PCC can include the IRO objects for any loose or strict hop in the configured LSP path in the Path Computation Request (PCReq) message to the PCE, the PCE ignores them and computes the path with the other constraints, excepting the IRO.
4. The PE router sends a PCReq message to the PCE to request a path for the LSP and includes the LSP parameters in the METRIC object, the LSPA object, and the Bandwidth object. It also includes the LSP object with the assigned PLSP-ID. At this point, the PCC does not delegate control of the LSP to the PCE.
5. The PCE computes a new path, reserves the bandwidth, and returns the path in a Path Computation Reply (PCRep) message with the computed ERO in the ERO object. It also includes the LSP object with the unique PLSP-ID, the METRIC object with the computed metric value (if any), and the Bandwidth object.

   Note:
   For the PCE to use the SRLG path diversity and admin-group constraints in the path computation, the user must configure the SRLG and admin-group membership against the MPLS interface and verify that the **traffic-engineering** option is enabled in IGP. This causes IGP to flood the link SRLG and admin-group membership in its participating area and for the PCE to learn it in its TE-DB.

6. The PE router updates the Control Processor Module (CPM) and the datapath with the new path.

   Note:
   Up to step 6, the PCC and PCE are using passive stateful PCE procedures. The following steps synchronize the LSP database of the PCC and PCE for both PCE-computed and PCE-controlled LSPs. The steps also initiate the active PCE stateful procedures for the PCE-controlled LSP only.

7. The PE router sends a PCRpt message to update PCE with the state of UP and the RRO as confirmation, including the LSP object with the unique PLSP-ID. For a PCE-controlled LSP, the PE router also sets a delegation control flag to delegate control to the PCE. The state of the LSP is now synchronized between the router and the PCE.
8. Following a network event or reoptimization, the PCE computes a new path for a PCE-controlled LSP and returns it in a Path Computation Update (PCUpd) message with the new ERO. It includes the LSP object with the same unique PLSP-ID assigned by the PCC and the Stateful Request Parameter

(SRP) object with a unique SRP-ID number to track error and state messages specific to this new path.

Note: If the **no pce-control** command is performed while a PCUpdate MBB is in progress on the LSP, the router aborts and removes the information and state related to the in-progress PCUpdate MBB. As the LSP is no longer controlled by the PCE, the router may take further actions depending on the state of the LSP. For example, if the LSP is up, and has FRR active or pre-emption, then the router starts a GlobalRevert or pre-emption MBB. If the LSP is down, the router starts the retry-timer to trigger setup.

9. The PE router updates the CPM and the datapath with the new path.
10. The PE router sends a new PCRpt message to update the PCE with the state of UP and the RRO as confirmation. The state of the LSP is now synchronized between the router and the PCE.
11. If the user makes a configuration change to the PCE-computed or PCE-controlled LSP, MPLS requests PCC to first revoke delegation in a PCRpt message (PCE-controlled only). Next, MPLS and PCC follow steps 1 to 10 to convey the changed constraint to the PCE, which results in a new path programmed into the datapath, the LSP databases of the PCC and the PCE to be synchronized, and the delegation to be returned to the PCE.

    In the case of an SR-TE LSP, MBB is not supported. Therefore, the PCC takes down the LSP and sends a PCRpt message to the PCE with the Remove flag set to 1 before following the preceding configuration change procedure.

For an LSP with an active path, the following applies:

- If the **path-computation-method pce** option is enabled on a PCC-controlled LSP that has an active path, no action is performed until the router needs a path for the LSP following a network event of an LSP parameter change. At this point, the preceding procedure is followed.
- If the **pce-control** command is enabled on a PCC-controlled or PCE-computed LSP that has an active path, the PCC issues a PCRpt message to the PCE with the state of UP and the RRO of the active path. It sets the delegation control flag to delegate control to the PCE. The PCE keeps the active path of the LSP and does not update until the next network event or reoptimization. At this point, the preceding procedure is followed.

The PCE supports the computation of disjointed paths for two different LSPs originating or terminating on the same or different PE routers. To indicate this constraint to the PCE, the user must configure the PCE path profile ID and path group ID to which the LSP belongs. These parameters are passed transparently by the PCC to the PCE and are opaque data to the router. Use the **path-profile** *profile-id* [**path-group** *group-id*] CLI command to configure the path profile and path group.

The association of the optional path group ID allows the PCE to determine which profile ID this path group ID must be used with. One path group ID is allowed per profile ID. However, the user can run the **path-profile** command multiple times to enter the same path group ID with multiple profile IDs. A maximum of five entries of the **path-profile** command can be associated with the same LSP. See Path Computation Element Protocol (PCEP) for more information about the operation of the PCE path profile.

2.2.3 SR-TE LSP path computation

For PCC-controlled SR-TE LSPs, CSPF is supported on the router using the **path-computation-method local-cspf** command. See SR-TE LSP path computation using local CSPF for details about the full CSPF path computation method. By default, the path is computed using the hop-to-label translation method. In the latter case, MPLS makes a request to the TE-DB to get the label corresponding to each hop entered

by the user in the primary path of the SR-TE LSP. See PCC-initiated and PCC-controlled LSPs for details about the hop-to-label translation.

The user can enable the **path-computation-method pce** option and forward the path computation request of a CSPF-enabled SR-TE LSP to a PCE instead of the local router CSPF. See SR-TE LSP instantiation for more information. To further delegate the reoptimization of the LSP to the PCE, the user can enable the **pce-control** command. In both cases, PCE is responsible for determining the label required for each returned explicit hop and includes this in the SR-ERO.

In all cases, the user can configure the maximum number of labels that the ingress LER can push for a specified SR-TE LSP by using the **max-sr-labels** command. This command sets a limit on the maximum label stack size of the SR-TE LSP primary path, which allows room to insert additional transport, service, and other labels when packets are forwarded in a specific context.

Use the **config**>**router**>**mpls**>**lsp**>**max-sr-labels** *label-stack-size* [**additional-frr-labels** *labels* CLI command to configure the maximum number of labels.

Set the **max-sr-labels** *label-stack-size* value to account for the required maximum label stack of the primary path of the SR-TE LSP.

Set the **additional-frr-labels** *labels* value to account for additional labels inserted by remote LFA or Topology Independent LFA (TI-LFA) for the backup next-hop of the SR-TE LSP. The supported range is 0 to 3 labels with a default value of 1.

The sum of the value of both labels represents the worst-case transport of the SR label stack size for this SR-TE LSP and is populated by MPLS in the TTM. Services can check the value to decide if a service can be bound or a route can be resolved to this SR-TE LSP. See SR-TE label stack check for services and shortcuts for more information about the label stack size check and requirements for services and shortcut applications.

The Maximum Stack Depth (MSD), which is the maximum label stack supported by the router, is always signaled by the PCC in the PCEP Open object as part of the SR-PCE-CAPABILITY TLV. See Datapath support for more information about the MSD.

In addition, the per-LSP value for the **max-sr-labels** *label-stack-size* option, if configured, is signaled by the PCC to the PCE in the SID depth value in a METRIC object for both a PCE-computed LSP and a PCE-controlled LSP. PCE computes and provides the full explicit path with TE-links specified. If there is no path with the number of hops lower than the MSD value, or the SID depth value is signaled, a reply with no path is returned to the PCC.

For a PCC-controlled LSP, if the label stack returned by the TE-DB hop-to-label translation exceeds the per-LSP maximum SR label stack size, the LSP is brought down.

2.2.4 SR-TE LSP path computation using hop-to-label translation

MPLS passes the path information to the TE-DB, which converts the list of hops into a label stack as follows:

- A loose hop with an address matching any interface (loopback or not) of a router (identified by the router ID) is always translated to a node SID. If the prefix matching the hop address has a node SID in the TE database, it is selected by preference. If not, the node SID of any loopback interface of the same router that owns the hop address is selected. In the latter case, the lowest IP-address of that router that has a /32 prefix SID is selected.
- A strict hop with an address matching any interface (loopback or not) of a router (identified by the router ID) is always translated to an adjacency SID. If the hop address matches the host address reachable in a local subnet from the previous hop, then the adjacency SID of that adjacency is selected. If the hop

address matches a loopback interface, it is translated to the adjacency SID of any link from the previous hop which terminates on the router owning the loopback. The adjacency SID label of the selected link is used.

In both cases, it is possible to have multiple matching previous hops in the case of a LAN interface. In this case, the adjacency-SID with the lowest interface address is selected.

- In addition to the IGP instance that resolved the prefix of the destination address of the LSP in the RTM, all IGP instances are scanned from the lowest to the highest instance ID, beginning with IS-IS instances and then OSPF instances. For the first instance via which all specified path hop addresses can be translated, the label stack is selected. The hop-to-label translation tool does not support paths that cross area boundaries. All SIDs and labels of a path are therefore taken from the same IGP area and instance.
- Unnumbered network IP interfaces, which are supported in the router's TE database, can be selected when converting the hops into an adjacency SID label when the user has entered the address of a loopback interface as a strict hop; however, the user cannot configure an unnumbered interface as a hop in the path definition.

Note: For the hop-to-label translation to operate, the user must enable TE on the network links, which means adding the network interfaces to MPLS and RSVP. The user must also enable the **traffic-engineering** option on all participating router IGP instances. If any router has the **database-export** option enabled in the participating IGP instances to populate the learned IGP link state information into the TE-DB, then enabling the **traffic-engineering** option is not required. For consistency purposes, Nokia recommends that the **traffic-engineering** option is always enabled.

2.2.5 SR-TE LSP path computation using local CSPF

This section describes full CSPF path computation for SR-TE LSP paths.

The **path-computation-method** [**local-cspf** | **pce**] command configures the path computation method for SR-TE LSPs. The **no** form of this command enables the user to select the computation method for the SR-TE LSP and set it to hop-to-label translation, local CSPF, or the PCE path computation method. The **no** form of this command, which is the default value, sets the computation method to the hop-to-label translation method. The **pce** option is not supported with the SR-TE LSP template.

2.2.5.1 Extending MPLS and TE database CSPF support to SR-TE LSP

The following MPLS and TE database features extend CSPF support to SR-TE LSP:

- IPv4 SR-TE LSP
- local CSPF on both primary and secondary standby paths of an IPv4 SR-TE LSP
- local CSPF in LSP templates of types **mesh-p2p-srte** and **one-hop-p2p-srte** of SR-TE auto-LSP
- support path computation in single area OSPFv2 and IS-IS IGP instances
- computes full explicit TE paths using TE links as hops and returning a list of SIDs consisting of adjacency SIDs and parallel adjacency set SIDs. SIDs of a non-parallel adjacency set are not used in CSPF. The details of the CSPF path computation are provided in SR-TE specific TE-DB changes. Loose-hop paths, using a combination of node SID and adjacency SID, are not required.

- use random path selection in the presence of ECMP paths that satisfy the LSP and path constraints. Least-fill path selection is not required.
- provide an option to reduce or compress the label stack such that the adjacency SIDs corresponding to a segment of the explicit path are replaced with a node SID whenever the constraints of the path are met by all the ECMP paths to that node SID. The details of the label reduction are provided in SR-TE LSP path label stack reduction.
- use legacy TE link attributes as in RSVP-TE LSP CSPF
- Uses timer re-optimization of all paths of the SR-TE LSP that are in the operational up state. This differs from RSVP-TE LSP resignal timer feature which re-optimizes the active path of the LSP only.

  MPLS provides the current path of the SR-TE LSP and TE-DB updates the total IGP or TE metric of the path, checking the validity of the hops and labels as per current TE-DB link information. CSPF then calculates a new path and provides both the new and metric updated current path back to MPLS. MPLS programs the new path only if the total metric of the new computed path is different than the updated metric of the current path, or if one or more hops or labels of the current path are invalid. Otherwise, the current path is considered one of the most optimal ECMP paths and is not updated in the datapath.

  Timer resignal applies only to the CSPF computation method and not to the ip-to-label computation method.
- use manual re-optimization of a path of the SR-TE LSP. In this case, the new computed path is always programmed even if the metric or SID list is the same.
- Supports ad-hoc re-optimization. This SR-TE LSP feature for SR-TE LSP triggers the ad-hoc resignaling of all SR-TE LSPs if one or more IGP link down events are received in TE-DB.

  After the re-optimization is triggered, the behavior is the same as the timer-based resignal or the delay option of the manual resignal. MPLS forces the expiry of the resignal timer and asks TE-DB to re-evaluate the active paths of all SR-TE LSPs. The re-evaluation consists of updating the total IGP or TE metric of the current path, checking the validity of the hops and labels, and computing a new CSPF for each SR-TE LSP. MPLS programs the new path only if the total metric of the new computed path is different than the updated metric of the current path, or if one or more hops or labels of the current path are invalid. Otherwise, the current path is considered one of the most optimal ECMP paths and is not updated in the datapath.
- support unnumbered interfaces in the path computation. There is no support for configuring an unnumbered interface as a hop in the path of the LSP. The path can be empty or include hops with the address of a system or loopback interface, but path computation can return a path that uses TE links corresponding to unnumbered interfaces.
- support **admin-group**, **hop-count**, IGP metric, and TE-metric constraints
- Bandwidth constraint is not supported because SR-TE LSP does not have an LSR state to book bandwidth. The **bandwidth** parameter, when enabled on the LSP path, has no impact on local CSPF path calculation. However, the **bandwidth** parameter is passed to PCE when it is the selected path computation method. PCE reserves bandwidth for the SR-TE LSP path accordingly.

2.2.5.2 SR-TE specific TE-DB changes

When the **traffic-engineering** command is enabled in an OSPFv2 instance, only local and remote TE-enabled links are added into the TE-DB. A TE-link is a link that has one or more TE attributes added to it in the MPLS interface context. Link TE attributes are TE metric, bandwidth, and membership in an SRLG or an Admin-Group.

To allow the SR-TE LSP path computation to use SR-enabled links that do not have TE attributes, the following changes are made:

- OSPFv2 passes all links, whether they are TE-enabled or SR-enabled, to the TE-DB, as currently performed by IS-IS.
- TE-DB relaxes the link back-check when performing a CSPF calculation to ensure that there is at least one link from the remote router to the local router. Because OSPFv2 advertises the remote link IP address or remote link identifier only when a link is TE-enabled, the strict check about the reverse direction of a TE-link cannot be performed if the link is SR-enabled but not TE-enabled.

As a consequence of this implementation, CSPF can compute an SR-TE LSP with SR-enabled links that do not have TE attributes. This means that if the user admin shuts down an interface in MPLS, an SR-TE LSP path that uses this interface does not go operationally down.

2.2.5.3 SR-TE LSP and auto-LSP-specific CSPF changes

The local CSPF for an SR-TE LSP is performed in two phases. Phase 1 computes a fully explicit path with all TE links to the destination specified, as in the case of an RSVP-TE LSP. If the user has enabled label stack reduction or compression for this LSP, Phase 2 is applied to reduce the label stack so that adjacency SIDs corresponding to a segment of the explicit path are replaced with a node SID whenever the constraints of the path are met by all the ECMP paths to that node SID. The details of the label reduction are provided in SR-TE LSP path label stack reduction.

The CSPF computation algorithm for the fully explicit path in Phase 1 remains mostly unchanged from its behavior in RSVP-TE LSP.

The meaning of a strict and loose hop in the path of the LSP is the same as in CSPF for RSVP-TE LSP. A strict hop means that the path from the previous hop must be a direct link. A loose hop means the path from the previous hop can traverse intermediate routers.

A loose hop may be represented by a set of back-to-back adjacency SIDs if not all paths to the node SID of that loose hop satisfy the path TE constraints. This is different from the IP-to-labeled path computation method where a loose hop always matches a node SID because no TE constraints are checked in the path to that loose hop.

When the label stack of the path is reduced or compressed, a strict hop may be represented by a node SID if all the links from the previous hop satisfy the path TE constraints. This is different from the IP-to-labeled path computation method where a strict hop always matches an adjacency SID or a parallel adjacency set SID.

The first phase of CSPF returns a full explicit path with each TE link specified all the way to the destination. The label stack may contain protected adjacency SIDs, unprotected adjacency SIDs, and adjacency set SIDs. The user can configure the type of adjacency protection for the SR-TE LSP using a CLI command as described in SR-TE LSP path protection.

SR OS does not support the origination of a global adjacency SID. If received from a third-party router implementation, it is added into the TE database but is not used in any CSPF path computation.

2.2.5.3.1 SR-TE LSP path protection

SR-TE LSP allows the user to configure whether the path of the LSP must use protected or unprotected adjacencies exclusively for all links of the path.

When SR OS routers form an IGP adjacency over a link and segment-routing context is enabled in the IGP instance, the static or dynamic label assigned to the adjacency is advertised in the link adjacency SID sub-TLV. By default, an adjacency is always eligible for LFA/RLFA/TI-LFA protection and the B-flag in the sub-TLV is set. The presence of a B-flag does not reflect the instant state of the availability of the adjacency LFA backup; it reflects that the adjacency is eligible for protection. The SR-TE LSP using the adjacency in its path still comes up if the adjacency does not have a backup programmed in the datapath at that instant. Use the **configure>router>isis>interface> no sid-protection** command to disable protection. When protection is disabled, the B-flag is cleared and the adjacency is not eligible for protection by LFA/RLFA/TI-LFA.

SR OS also supports the adjacency set feature that treats a set of adjacencies as a single object and advertises a link adjacency sub-TLV for it with the S-flag (SET flag) set to 1. The adjacency set in the SR OS implementation is always unprotected, even if there is a single member link in it and therefore the B-flag is always clear. Only a parallel adjacency set, meaning that all links terminate on the same downstream router, is used by the local CSPF feature.

The same P2P link can participate in a single adjacency and in one or more adjacency sets. Therefore, multiple SIDs can be advertised for the same link.

Third party implementations of Segment Routing may advertise two SIDs for the same adjacency when LFA is enabled in the IS-IS or OSPF instance: one protected with the B-flag set and one unprotected with the B-flag clear. SR OS can achieve the same behavior using one of the following two options:

- Enabling the allocation of dual SIDs using the following command for IS-IS or OSPF respectively:
  - **MD-CLI**

    ```
    configure router isis segment-routing adjacency-sid allocate-dual-sids true
    configure router ospf segment-routing adjacency-sid allocate-dual-sids true
    ```

  - **classic CLI**

    ```
    configure router isis segment-routing adjacency-sid allocate-dual-sids
    configure router ospf segment-routing adjacency-sid allocate-dual-sids
    ```

- Adding a link to a single-member adjacency SET, in which case a separate SID is advertised for the SET and the B-flag is cleared while the SID for the regular adjacency over that link has its B-flag set by default

Note: LFA must be enabled under the IS-IS or OSPF instance for the previously mentioned cases.

In all cases, SR OS CSPF can use all local and remote SIDs to compute a path for an SR-TE LSP based on the needed local protection property.

The following different behaviors of CSPF are introduced with SR-TE LSP:

- If the **local-sr-protection** command is not enabled (**no local-sr-protection**) or is set to **preferred**, the local CSPF prefers a protected adjacency over an unprotected adjacency whenever both exist for a TE link. This is done on a link-by-link basis after the path is computed based on the LSP path constraints. This means that the protection state of the adjacency is not used as a constraint in the path computation. It is only used to select an SID among multiple SIDs after the path is selected. Thus, the computed path can combine both types of adjacencies.

  If a parallel adjacency set exists between two routers in a path and all the member links satisfy the constraints of the path, it is selected in preference to a single protected adjacency, which is selected in preference to a single unprotected adjacency.

If multiples ECMP paths satisfy the constraints of the LSP path, one path is selected randomly and then the SID selection above applies. There is no check if the selected path has the highest number of protected adjacencies.

- If the **local-sr-protection** command is set to a value of **mandatory**, CSPF uses it as an additional path constraint and selects protected adjacencies exclusively in computing the path of the SR-TE LSP. Adjacency sets cannot be used because they are always unprotected.

  If no path satisfies the other LSP path constraints and consists of all TE links with protected adjacencies, the path computation returns no path.

- If the **local-sr-protection** command is set to a value of **none**, CSPF uses it as an additional path constraint and selects unprotected adjacencies exclusively in computing the path of the SR-TE LSP.

  If a parallel adjacency set exists between two routers in a path and all the member links satisfy the constraints of the path, it is selected in preference to a single unprotected adjacency.

  If no path satisfies the other LSP path constraints and consists of all TE links with unprotected adjacencies, the path computation returns no path.

The **local-sr-protection** command impacts PCE-computed and PCE-controlled SR-TE LSPs. When the **local-sr-protection** command is set to the default value **preferred**, or to the explicit value of **mandatory**, the **local-protection-desired** flag (L-flag) in the LSPA object in the PCReq (Request) message or in the PCRpt (Report) message is set to a value of 1.

When the **local-sr-protection** command is set to **none**, the **local-protection-desired** flag (L-flag) in the LSPA object is cleared. The PCE path computation checks this flag to decide if protected adjacencies are used in preference to unprotected adjacencies (L-flag set) or must not be used at all (L-flag clear) in the computation of the SR-TE LSP path.

2.2.5.3.2 SR-TE LSP path label stack reduction

The objective of the label stack reduction is twofold:

- It reduces the label stack so ingress PE routers with a lower Maximum SID Depth (MSD) can still work.
- It provides the ability to spray packets over ECMP paths to an intermediate node SID when all these paths satisfy the constraints of the SR-TE LSP path. Even if the resulting label stack is not reduced, this aspect of the feature is still useful.

If the user enables the **label-stack-reduction** command for this LSP, a second phase is applied, attempting to reduce the label stack that resulted from the fully explicit path with adjacency SIDs and adjacency sets SIDs computed in the first phase.

This is to attempt a replacement of adjacency and adjacency set SIDs corresponding to a segment of the explicit path with a node SID whenever the constraints of the path are met by all the ECMP paths to that node SID.

The label stack reduction algorithm uses the following procedure.

1. Phase 1 of the CSPF returns up to three fully explicit ECMP paths that are eligible for label stack reduction. These paths are equal cost from the point of view of IGP metric or TE metric as configured for that SR-TE LSP.
2. Each fully explicit path of the SR-TE LSP that is computed in Phase 1 of the CSPF is split into a number of segments that are delimited by the user-configured loose or strict hops in the path of the LSP. Label stack reduction is applied to each segment separately.

3. Label stack reduction in Phase 2 consists of traversing the CSPF tree for each ECMP path returned in Phase 1, then attempting to find the farthest node SID in a path segment that can be used to summarize the entire path up to that node SID. This requires that all links of ECMP paths are able to reach the node SID from the current node on the CSPF tree to satisfy all the TE constraints of the SR-TE LSP paths. ECMP is based on the IGP metric, in this case, because this is what routers use in the datapath when forwarding a packet to the node SID.

   If the TE metric is enabled for the SR-TE LSP, one of the constraints is that the TE metric must be the same value for all the IGP metric ECMP paths to the node SID.

4. CSPF in Phase 2 selects the first candidate ECMP path from Phase 1, which reduced label stack that satisfies the constraint carried in the **max-sr-labels** command.

5. The CSPF path computation in Phase 1 always avoids a loop over the same hop, as is the case with the RSVP-TE LSP. In addition, the label stack reduction algorithm prevents a path from looping over the same hop because of the normal routing process. For example, it checks if the same node is involved in the ECMP paths of more than one segment of the LSP path and builds the label stack to avoid this situation.

6. During the MBB procedure of a timer or the manual re-optimization of an SR-TE LSP path, the TE-DB performs the following steps in addition to the initial path computation.

   - MPLS provides the TE-DB with the current working path of the SR-TE LSP.
   - The TE-DB updates the path's metric based on the IGP or TE link metric (if the TE metric enabled for the SR-TE LSP).
     - For each adjacency SID, the TE-DB verifies that the related link and SID are still in its database and that the link fulfills the LSP constraints. If so, it picks up the current metric.
     - For each node SID, the TE-DB verifies that the related prefix and SID are still available, and if so, checks that all the links on the shortest IGP path to the node owning the node SID fulfill the SR-TE LSP path constraints. This step reuses the same checks detailed in step 3 for the label compression algorithm.
   - CSPF computes a new path with or without label stack reduction as described in steps 1, 2, and 3.
   - The TE-DB returns both paths to MPLS. MPLS always programs the new path in the case of a manual re-optimization. MPLS compares the metric of the new path to the current path and if different, programs the new path in the case of a timer re-optimization.

7. The TE-DB sends the reduced path ERO and label stack to MPLS, along with the following information:

   - a list of SRLGs of each hop in the ERO, represented by a node SID, including the SRLGs used by links in all ECMP paths to reach that node SID from the previous hop
   - the cost of each hop in the ERO represented by an adjacency SID or adjacency set SID. The cost corresponds to the IGP metric or TE metric (if the TE metric is enabled for the SR-TE LSP) of that link or set of links. In the case of an adjacency set, all TE metrics of the links must be the same, otherwise CSPF does not select the set.
   - the cost of each hop in the ERO represented by a node SID, which corresponds to the cumulated IGP metric or TE metric (if the TE metric is enabled for the SR-TE LSP) to reach the node SID from the previous hop using the fully explicit path computed in Phase 1.
   - the total cost or computed metric of the SR-TE LSP path. This consists of the cumulated IGP metric or TE metric (if TE metric enabled for the SR-TE LSP) of all hops of the fully explicit path computed in Phase 1 of the CSPF.

8. If label stack reduction is disabled, the values of the **max-sr-labels** and the **hop-limit** commands are applied to the full explicit path in Phase 1.

   The minimum of the two values is used as a constraint in the full explicit path computation.

   If the resulting ECMP paths net hop-count in Phase 1 exceeds this minimum value, the TE-DB does not return a path to MPLS.

9. If label stack reduction is enabled, the values of the **max-sr-labels** and the **hop-limit** commands are both ignored in Phase 1 and only the value of the **max-sr-labels** command is used as a constraint in Phase 2.

   If the Phase 2 reduction of all candidate paths results in a net label stack size that exceeds the value of the **max-sr-labels** command, the TE-DB does not return a path to MPLS.

10. The label stack reduction uses a node SID to replace a segment of the SR-TE LSP path; using an anycast SID or a prefix SID with the N-flag clear is not supported.

2.2.5.3.3 Interaction with SR-TE LSP path protection

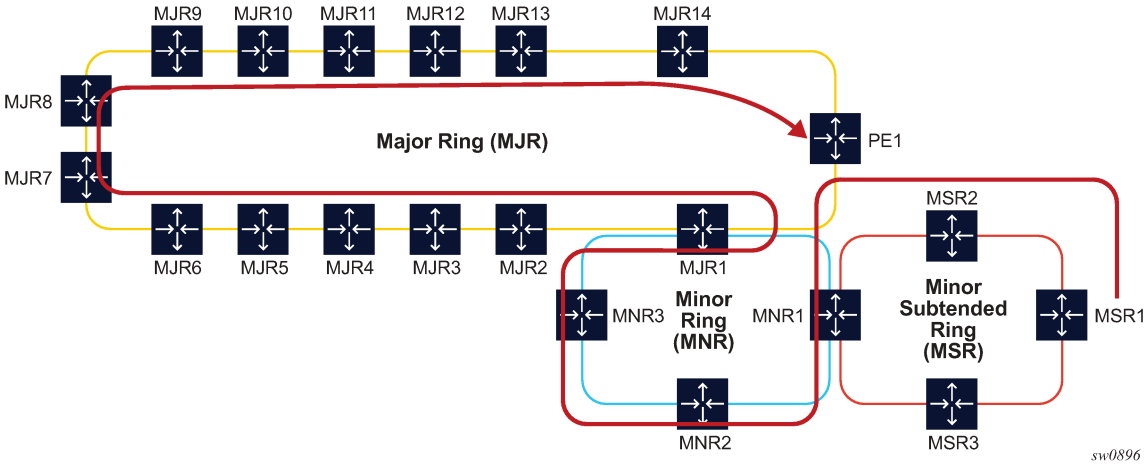
Label stack reduction is only attempted when the path protection **local-sr-protection** command is disabled or configured to the value of **preferred**.

If **local-sr-protection** is configured to a value of **none** or **mandatory**, the command is ignored, and the fully explicit path computed in Phase 1 is returned by the TE-DB CSPF routine to MPLS. This is because a node SID used to replace an adjacency SID or an adjacency set SID can be unprotected or protected by LFA and this is based on local configuration on each router which resolves this node SID but is not directly known in the information advertised into the TE-DB. Therefore, CSPF cannot enforce the protection constraint requested along the path to that node SID.

2.2.5.3.4 Examples of SR-TE LSP path label stack reduction

The following figure shows a metro aggregation network with three levels of rings for aggregating regional traffic from edge ring routers to a PE router.

Figure 23: Label stack reduction in a 3-tier ring topology



The path of the highlighted LSP uses admin groups to force the traffic eastwards or westwards over the 3-ring topologies such that it uses the longest path possible. Assume all links in a bottom-most ring1 have admin-group=east1 for the eastward direction and admin-group=west1 for the westward direction.

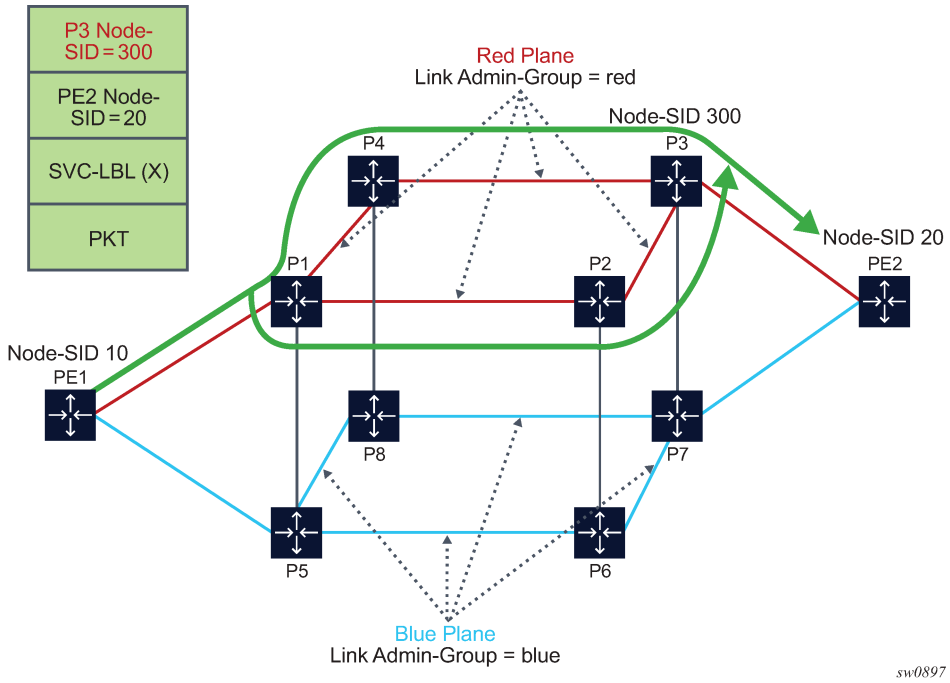
Similarly, links in middle ring2 have admin-group=east2 and admin-group=west2, and links in top-most ring3 have admin-group=east3 and admin-group=west3. To achieve the longest path shown, the LSP or path should have an include statement: include east1 west2 east3. The fully explicit path computed in Phase 1 of CSPF results in label stack of size 18.

The label stack reduction algorithm searches for the farthest node SID in that path, which can replace a segment of the strict path while maintaining the stated admin-group constraints. The reduced label stack contains the SID adjacency MSR1-MSR2, the found node SIDs plus the node SID of the destination for a total of four labels to be pushed on the packet (the label for the adjacency MSR1-MSR2 is not pushed):

{N-SID MNR2, N-SID of MNR3, N-SID of MJR8, N-SID of PE1}

The following figure shows an example topology that creates two TE planes by applying a common admin group to all links of a specified plane. There are a total of four ECMP paths to reach PE2 from PE1, two within the red plane and two within the blue plane.

Figure 24: Label stack reduction in the presence of ECMP paths



For an SR-TE LSP from PE1 to PE2, which includes the red admin-group as a constraint, Phase 1 of CSPF results in two fully explicit paths using adjacency SID of the red TE links:

path 1 = {PE1-P1, P1-P2, P2-P3, P3-PE2}

path 2 = {PE1-P1, P1-P4, P4-P3, P3-PE2}

Phase 2 of CSPF finds node SID of P3 as the farthest hop it can reach directly from PE1 while still satisfying the constraint to include the red admin-group constraint. If the node SID of PE2 is used as the only SID, then traffic would also be sent over the blue links.

Then, the reduced label stack is: {P3 Node-SID=300, PE2 Node-SID=20}.

The resulting SR-TE LSP path combines the two explicit paths out of Phase 1 into a single path with ECMP support.

2.2.6 SR-TE LSP paths using explicit SIDs

SR OS supports the ability for SR-TE primary and secondary paths to use a configured path containing explicit SID values. The SID value for an SR-TE LSP hop is configured using the **sid-label** *sid-value* parameter of the **configure**>**router mpls path hop** command, where *sid-value* specifies an MPLS label value for that hop in the path.

When SIDs are explicitly configured for a path that consists of either all SIDs or all IP address hops, the user must provide all of the necessary SIDs to reach the destination. The router does not validate whether the provided label stack is correct.

A path containing SID label hops is used even if **path-computation-method** {**local-cspf** | **pce**} is configured for the LSP. That is, the path computation method configured at the LSP level is ignored when explicit SIDs are used in the path. This means that the router can bring up the path if the configured path contains SID hops even if the LSP has path computation enabled.

Note:
When an LSP consists of some SID labeled paths and some paths under local-CSPF computation, the router cannot guarantee SRLG diversity between the CSPF paths and the SID labeled paths. CSPF is not aware of the existence of the SID labeled paths because they are not listed in the TE database.

Paths containing explicit SID values can only be used by SR-TE LSPs.

2.2.7 SR-TE LSP protection

The router supports local protection of a specific segment of an SR-TE LSP and end-to-end protection of the complete SR-TE LSP.

Whenever possible, an LFA next hop protects each path locally along the network. The protection of a node SID reuses the base LFA, TI-LFA, and remote LFA features used with SR shortest path tunnels. To augment the protection level, the SR OS adds the protection of an adjacency SID in the specific context of an SR-TE LSP. The user must enable the loopfree [**remote-lfa**] [**ti-lfa**] command in IS-IS or OSPF.

- **MD-CLI**

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

- **classic CLI**

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

An SR-TE LSP has state at the ingress LER only. The LSR has state for the node SIDs and adjacency SIDs that have labels programmed in the label stack of the received packet and that represent the part of the ERO of the SR-TE LSP on this router and downstream of this router. To provide protection for an SR-

TE LSP, each LSR node must attempt to program a link-protect or node-protect LFA next hop in the ILM record of a node SID or of an adjacency SID, and the LER node must do the same in the LTN record of the SR-TE LSP. Details about this behavior are as follows:

- For an ILM record of a node SID of a downstream router that is not directly connected, the ILM of the node SID points to the backup NHLFE computed by the LFA SPF and programmed by the SR module for this node SID. Depending on the topology and LFA policy used, this can be a link-protect or node-protect LFA next hop.

  This behavior is already supported in the SR shortest path tunnel feature at both the LER and the LSR. Consequently, an SR-TE LSP that transits at an LSR and matches the ILM of a downstream node SID automatically takes advantage of this protection, when enabled. If required, the user can disable node SID protection under the IGP instance by excluding the prefix of the node SID from the LFA.

- For an ILM record of a node SID of a directly connected router, the LFA SPF provides only link protection. The ILM or LTN record of this node SID points to the backup NHLFE of this LFA next hop. An SR-TE LSP that transits at an LSR and matches the ILM of a neighboring node SID automatically takes advantage of this protection, when enabled.

  Note:
  Only link protection is possible in this case because packets matching this ILM record can either terminate on the neighboring router owning the node SID or can be forwarded to different next hops of the neighboring router (that is, to different next-next-hops of the LSR providing the protection). The LSR providing the connection does not have context to distinguish among all possible SR-TE LSPs and, therefore, can protect only the link to the neighboring router.

- For an ILM or LTN record of an adjacency SID, the handling is the same as for an ILM record of a node SID of a directly connected router.

  When protecting an adjacency SID, the PLR first tries to select a parallel link to the node SID of the directly connected neighbor. The selection is based on the lowest interface ID. If no parallel links exist, regular LFA or remote LFA algorithms are applied to find a loopfree path to reach the node SID of the neighbor via other neighbors.

  The ILM or LTN for the adjacency SID must point to this backup NHLFE and benefits from FRR link-protection. As a result, an SR-TE LSP that transits at an LSR and matches the ILM of a local adjacency SID automatically takes advantage of this protection, when enabled.

- For an ingress LER, the LTN record points to the SR-TE LSP NHLFE at the ingress LER, which itself points to the NHLFE of the SR shortest path tunnel to the node SID or adjacency SID of the first hop in the ERO of the SR-TE LSP. For this reason, the FRR link or node protection at an ingress LER is inherited directly from the SR shortest path tunnel.

When an adjacency to a neighbor fails, the following procedures are followed for both the LFA protected SID and the LFA unprotected SID of this adjacency in SR-MPLS. An adjacency can have both types of SIDs assigned by configuration. An LFA protected adjacency SID is eligible for LFA protection, but the following procedures apply even if a LFA backup was not programmed at the time of the failure. An LFA unprotected adjacency SID is not eligible for LFA protection.

- IGP withdraws the advertisement of the link TLV as well as its adjacency SID sub-TLV.
- The adjacency SID hold timer starts.
- The LTN and ILM records of the adjacency are kept in the datapath for as long as the adjacency SID hold timer is running. This allows packets to flow over the LFA backup path, when the adjacency is

protected, and allows the ingress LER or PCE time to compute a new path of the SR-TE LSP after IGP converges.

- If the adjacency is restored while the adjacency SID hold timer is running, it remains programmed in the datapath with the retained SID values. However, the backup NHLFE may change if a new LFA SPF runs while the adjacency SID hold timer is running. An update to the backup NHLFE is performed immediately following the LFA SPF. In all cases, the adjacency keeps its assigned SID label value.
- If the adjacency SID hold timer expires before the adjacency is restored, the SID is deprogrammed from the datapath and the label returned into the common pool where it was drawn from. Users of the adjacency (SR-TE LSP and SR Policy) are also informed. When the adjacency is subsequently restored, it gets assigned its allocated static-label value or a new dynamic-label value.
- A new PG-ID is assigned each time an adjacency comes back up. This PG-ID is used by the ILM and LTN of the adjacency SID and of all downstream node SIDs that resolve to a next hop over this adjacency.

The adjacency SID hold timer is configured using the **adj-sid-hold** command; it is activated when the adjacency to neighbor fails because of any of the following conditions.

- The network IP interface goes down because of a link or port failure or because the user performed a shutdown of the port.
- The user shuts down the network IP interface in the **configure router**, or **configure router ospf**, or **configure router isis** context.
- The adjacency SID hold timer is not activated if the user deletes an interface in the following contexts.

```
configure router ospf
configure router isis
```

Note: The adjacency SID hold timer does not apply to the ILM or LTN of a node SID, because NHLFE information is updated in the datapath as soon as IGP is converged locally and new primary and LFA backup next hops have been computed.

Although protection is enabled globally for all node SIDs and local adjacency SIDs when the user enables the **loopfree-alternate** option in IS-IS or OSPF at the LER and the LSR, applications may exist for which the user wants traffic to never divert from the strict hop computed by CSPF for an SR-TE LSP. In such cases, use the **sid-protection** command to disable protection for all adjacency SIDs formed over a specific network IP interface. Alternatively, configure a second unprotected SID for each adjacency using the **allocate-dual-sids** command.

The protection state of an adjacency SID is advertised in the B-flag of the IS-IS or OSPF adjacency SID sub-TLV.

2.2.7.1 Local protection

Whenever possible, an LFA next hop protects each path locally along the network. The protection of a SID node reuses the base LFA, TI-LFA, and remote LFA features introduced with segment routing shortest path tunnels. To augment the protection level, the SR OS adds the protection of an adjacency SID in the specific context of an SR-TE LSP. You must enable the loopfree [**remote-lfa**] [**ti-lfa**] command in IS-IS or OSPF.

- **MD-CLI**

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

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

- **classic CLI**

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

This behavior is already supported in the SR shortest path tunnel feature at both LER and LSR. An SR-TE LSP that transits at an LSR and that matches the ILM of a downstream SID node automatically takes advantage of this protection, when enabled. If required, SID node protection can be disabled under the IGP instance by excluding the prefix of the SID node from LFA.

2.2.7.2 End-to-end protection

This section provides a brief introduction to end to end protection for SR-TE LSPs. See Seamless BFD for SR-TE LSPs for more detailed description of protection switching using Seamless BFD and a configured failure-action.

End-to-end protection for SR-TE LSPs is provided using secondary or standby paths. Standby paths are permanently programmed in the datapath, whereas secondary paths are only programmed when they are activated. S-BFD is used to provide end-to-end connectivity checking. The **failure-action failover-or-down** command under the **bfd** context of the LSP is used to configure a switchover from the currently active path to an available standby or secondary path if the S-BFD session fails on the currently active path. If S-BFD is not configured, the router that is local to a segment can only detect failures of the top SID for that segment. End-to-end protection with S-BFD can be combined with local protection, but it is recommended that the S-BFD control packet timers be set to 1 second or more to allow sufficient time for any local protection action for a specific segment to complete without triggering S-BFD to go down on the end-to-end LSP path.

To prevent failure between the paths of an SR-TE LSP, that is to avoid, for example, a failure of a primary path that affects its standby backup path, then disjoint paths should be configured or the **srlg** command configured on the secondary paths.

As with RSVP-TE LSPs, SR-TE standby paths support the configuration of a path preference. This value is used to select the standby path to be used when more than one available path exists.

See Seamless BFD for SR-TE LSPs for more information about end-to-end protection of SR-TE LSPs with S-BFD.

2.2.8 Seamless BFD for SR-TE LSPs

Seamless BFD (S-BFD) is a form of BFD that requires significantly less state and reduces the need for session bootstrapping as compared to LSP BFD. For more information, see "Seamless Bidirectional Forwarding Detection" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide*. S-BFD also requires centralized configuration for the reflector function and a mapping at the head-end node between the remote session discriminator and the IP address for the reflector by each session. This user guide describes the application of S-BFD to SR-TE LSPs and the LSP configuration required for this feature. The configuration and mapping are described in the *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide*.

By default, S-BFD operates in asynchronous mode. The reflector encapsulates and routes IP/UDP encapsulated S-BFD packets back to the initiator using the IGP shortest path. SR OS also supports a controlled return TE path for BFD reply packets when S-BFD operates in echo mode with the reflector router forwarding packets back toward the initiator on a specified labelled path using, for example, an SR policy. This allows the operator to configure a specific TE return path for each S-BFD session on an SR-TE LSP at the initiating node. In this case, the reflector function at the tail end of the LSP is bypassed.

S-BFD is supported in the following SR objects or contexts:

- PCC-initiated
  - SR-TE LSP level
  - SR-TE primary path
  - SR-TE secondary and standby path
- PCE-initiated SR-TE LSPs
- SR-TE auto-LSPs

2.2.8.1 Configuration of S-BFD on SR-TE LSPs

For PCC-initiated or PCC-controlled LSPs, an operator can configure an S-BFD session under the **bfd** context of SR-TE LSP, the primary path, the SR-TE secondary path, and the lsp-template by using:

- Classic CLI commands

  config>router>mpls>lsp bfd

  config>router>mpls >lsp>primary bfd

  config>router>mpls>lsp>secondary bfd

  config>router>mpls>lsp-template bfd
- MD-CLI commands

  configure router mpls lsp bfd

  configure router mpls lsp primary bfd

  configure router mpls lsp secondary bfd

  configure router mpls lsp-template bfd

The operator can configure S-BFD to operate in one of the following modes:

- **routed return path**

  In this mode, the session initiator sends BFD packets on the LSP toward the reflector node. The reflector node sends the BFD reply packet back to the initiator through a routed return path. The remote discriminator value is determined by passing the "to" address of the LSP to BFD, which then matches it to a mapping table of peer IP addresses to reflector remote discriminators that are created by the centralized configuration under the IGP. If no match for the "to" address of the LSP exists, a BFD session is not established on the LSP or path. For more information, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide*.

  Note: A remote peer IP address to discriminator mapping must exist before bringing an LSP administratively up.

- **controlled return path**

In this mode, operators configure a return path label for the BFD session to the initiator. The router pushes an additional MPLS label on S-BFD packets at the bottom of the stack and the BFD session operates in echo mode. The return path label refers to an MPLS binding SID of an SR policy programmed at the far end of the SR-TE LSP. The operator can use this SR policy to forward S-BFD reply packets along an explicit TE path back to the initiator, avoiding the IGP shortest path. The operator can configure different LSPs or LSP paths using different return path labels referring to different SR policies at the LSP far end. The SR policies can have segment lists with different paths, ensuring the BFD reply packets from different LSP paths do not share the same outcome. S-BFD packets on these sessions bypass the reflector at the far end of the LSP. Therefore, the operator does not need to configure a reflector discriminator for these sessions.

The referenced BFD template must specify parameters consistent with an S-BFD session. For example, the endpoint type is **cpm-np** for platforms supporting a CPM P-chip; otherwise, a CLI error is generated. Operators can use the same BFD template for both S-BFD and any other type of BFD session requested by MPLS.

If S-BFD is configured at the LSP level, sessions are created on all paths of the LSP.

- **Classic CLI commands**

```
config>router>mpls
   lsp-template name on-demand-p2p-srte template-id
      bfd
        bfd-enable
        bfd-template name
        return-path-label label value
        wait-for-up-timer seconds
```

- **MD-CLI commands**

```
*[gl:/configure router "Base" mpls lsp-template name]
                  type p2p-sr-te-on-demand
                  bfd
                      bfd-liveness {true|false}
                      bfd-template reference
                      return-path-label number
                      wait-for-up-timer number
```

Operators can also configure S-BFD on the primary or a specific secondary path of the LSP, as follows:

- **Classic CLI commands**

```
config>router>mpls>lsp name sr-te
   primary name
      bfd
        bfd-enable
        bfd-template name
        return-path-label label
        wait-for-up-timer seconds
        exit
```

```
config>router>mpls>lsp name sr-te
   secondary name
      bfd
        bfd-enable
        bfd-template name
        return-path-label label
        wait-for-up-timer seconds
        exit
```

- **MD-CLI commands**

```
*[gl:/configure router "Base" mpls lsp name primary name]
                bfd
                  bfd-liveness {true|false}
                  bfd-template reference
                  return-path-label number
                  wait-for-up-timer number
                  exit
```

```
*[gl:/configure router "Base" mpls lsp name secondary name]
                bfd
                  bfd-liveness {true|false}
                  bfd-template reference
                  return-path-label number
                  wait-for-up-timer number
                  exit
```

The wait-for-up-timer is only applicable if a failure action is configured using the **failover-or-down** command. For more information, see Support for BFD failure action with SR-TE LSPs.

For PCE-initiated LSPs and SR-TE auto-LSPs, the operator specifies the S-BFD session parameters in the LSP template. The "to" address used for determining the remote discriminator is derived from the far-end address of the auto-LSP or PCE-initiated LSP.

- **Classic CLI commands**

```
config>router>mpls
   lsp-template name mesh-p2p-srte
      bfd
        bfd-enable
        bfd-template name
        wait-for-up-timer seconds
```

```
config>router>mpls
   lsp-template name one-hop-p2p-srte
      bfd
        bfd-enable
        bfd-template
        wait-for-up-timer seconds
```

```
config>router>mpls
   lsp-template name on-demand-p2p-srte
      bfd
        bfd-enable
        bfd-template name
        wait-for-up-timer seconds
```

- **MD-CLI commands**

```
*[gl:/configure router "Base" mpls lsp-template name]
                  type p2p-sr-te-mesh
                  bfd
                    bfd-liveness {true|false}
                    bfd-template reference
```

```
                    wait-for-up-timer number
```

```
*[gl:/configure router "Base" mpls lsp-template name]
                  type p2p-sr-te-one-hop
                  bfd
                    bfd-liveness {true|false}
                    bfd-template reference
                    wait-for-up-timer number
```

```
*[gl:/configure router "Base" mpls lsp-template name]
                  type p2p-sr-te-on-demand
                  bfd
                    bfd-liveness {true|false}
                    bfd-template reference
                    wait-for-up-timer number
```

2.2.8.2 Support for BFD failure action with SR-TE LSPs

SR OS supports the configuration of a **failure-action** of type **failover-or-down** for SR-TE LSPs. The **failure-action** command is configured at the LSP level or in the LSP template. It can be configured whether S-BFD is applied at the LSP level or the individual path level.

For LSPs with a primary path and a standby or secondary path and **failure-action** of type **failover-or-down**:

- A path is held in an operationally down state when its S-BFD session is down.
- If all paths are operationally down, then the SR-TE LSP is taken operationally down and a trap is generated.
- If S-BFD is enabled at the LSP or active path level, a switchover from the active path to an available path is triggered on failure of the S-BFD session on the active path (primary or standby).
- If S-BFD is not enabled on the active path, and this path is shut down, then a switchover is triggered.
- If S-BFD is enabled on the candidate standby or secondary path, then this path is only selected if S-BFD is up.
- An inactive standby path with S-BFD configured is only considered as available to become active if it is not operationally down; for example, if the S-BFD session is up and all other criteria for it to become operational are true. It is held in an inactive state if the S-BFD session is down.
- The system does not revert to the primary path nor start a reversion timer when the primary path is either administratively down or operationally down because the S-BFD session is not up or down for any other reason.

For LSPs with only one path and **failure-action** of type **failover-or-down**:

- A path is held in an operationally down state when its S-BFD session is down.
- If the path is operationally down, then the LSP is taken operationally down and a trap is generated.

Note: S-BFD and other OAM packets can still be sent on an operationally down SR-TE LSP.

2.2.8.2.1 SR-TE LSP state changes and failure actions based on S-BFD

A path is first configured with S-BFD. This path is held operationally down and not added to the TTM until BFD comes up (subject to the BFD wait time).

The BFD **wait-for-up-timer** command provides a mechanism that cleans up the LSP path state at the head end in both cases where S-BFD does not come up in the first place and where S-BFD goes from up to down. This timer is started when BFD is first enabled on a path or an existing S-BFD session transitions from up to down. When this timer expires and if S-BFD is not up, the path is torn down by removing it from the TTM and the IOM and the LSP retry timer is started.

In the case where S-BFD goes from up to down, if there is only one path, the LSP is removed immediately from the TTM when S-BFD fails, then is deprogrammed when the **wait-for-up-timer** expires.

If all the paths of an LSP are operationally down because of S-BFD, then the LSP is taken operationally down and removed from the TTM and the BFD **wait-for-up-timer** is started for each path. If one or more paths do not have S-BFD configured on them, or are otherwise not down, then the LSP is not taken operationally down.

When an existing S-BFD session fails on a path and the failure action is **failover-or-down**, the path is put into the operationally down state. This state and reason code are displayed in a **show>router>bfd>seamless-bfd** command and a trap is generated. The configured failure action is then enacted.

2.2.8.3 S-BFD operational considerations

A minimum control packet timer transmit interval of 10 ms can be configured. To maximize the reliability of S-BFD connectivity checking in scaled scenarios with short timers, cases where BFD can go down because of normal changes to the next hop of an LSP path at the head end must be avoided. Nokia recommends that LFA should not be configured at the head-end LER when S-BFD is used with sub-second timers. When the LFA is not configured, protection of the SR-TE LSP is still provided end-to-end by the combination of S-BFD connectivity checking and primary or secondary path protection.

Similar to the case of LDP and RSVP, S-BFD uses a single path for a loose hop; multiple S-BFD sessions for each of the ECMP paths or spraying of S-BFD packets across the paths is not supported. S-BFD is not down until all the ECMP paths of the loose hop go down.

Note:
With very short control packet timer values in scaled scenarios, S-BFD may bounce if the next hop that the path is currently using goes down because it takes a finite time for BFD to update to use another next hop in the ECMP set.

2.2.9 Static route resolution using SR-TE LSP

The user can forward packets of a static route to an indirect next hop over an SR-TE LSP programmed in the TTM by configuring the following static route tunnel binding command:

```
config>router>static-route-entry {ip-prefix/prefix-length} [mcast] indirect {ip-address}
 tunnel-next-hop
       – resolution {any | disabled | filter}
       – resolution-filter
           – [no] sr-te
               – [no] [lsp name1]
```

```
                - [no] [lsp name2]
                - .
                - .
                - [no] [lsp name-N]
            - exit
        - [no] disallow-igp
        - exit
    - exit
```

The user can select the **sr-te** tunnel type and either specify a list of SR-TE LSP names to use to reach the indirect next hop of this static route or allow the SR-TE LSPs to automatically select the indirect next hop in the TTM.

2.2.10 SR-MPLS shortcuts using SR-TE LSP

SR OS supports SR-TE IGP shortcuts with OSPFv2 and IS-IS under SR-MPLS. By default, the SR-TE LSPs are not eligible as SR-MPLS IGP shortcuts. This configuration reduces the risk of accidental SR-MPLS forwarding loops.

To enable SR-TE SR-MPLS shortcuts, configure SR-TE LSPs to be eligible as shortcuts and instruct IGP that SR-TE LSPs can be used as IGP shortcuts in SR-MPLS.

Use the following commands to make an SR-TE LSP eligible as a SR-MPLS IGP shortcut:

- **MD-CLI**

```
configure router mpls lsp igp-shortcut allow-sr-over-srte
configure router mpls lsp igp-shortcut relative-metric
```

- **classic CLI**

```
configure router mpls lsp sr-te igp-shortcut [lfa-protect | lfa-only] allow-sr-over-srte
configure router mpls lsp sr-te igp-shortcut relative-metric [offset] allow-sr-over-srte
```

Use the following commands to make IGP consider the eligible SR-TE LSPs as IGP shortcuts in SR-MPLS:

```
configure router ospf igp-shortcut allow-sr-over-srte
configure router isis igp-shortcut allow-sr-over-srte
```

The following considerations apply when using SR-TE tunnels consisting of only adjacency SIDs as SR-MPLS shortcuts:

- The SR-TE LSP must be locally configured. An SR-TE LSP configured by PCE is not supported as an IGP shortcut for SR-MPLS.
- A configured SR-TE LSP path or SR policy candidate path with a top SID that resolves into another SR-TE LSP shortcut results in packet drops.
- A configured SR-TE LSP is used by IGP as shortcut for a SR-MPLS tunnel only if the top SID in the SR-TE path list is an adjacency SID or adjacency SET. If the top SID is a node SID, the SR-TE LSP is not used for SR-MPLS, but can still be used as a shortcut for IP.
- A configured SR-TE LSP used as shortcut for an SR-MPLS tunnel must consist of a SR-TE path list containing only adjacency SIDs.
- Nokia recommends that you run sBFD over a SR-TE LSP used as shortcut for a SR-MPLS tunnel.

- When SR-TE shortcuts are enabled and ECMP paths exist, the SR-TE shortcut LSP with the least number of labels is preferred.
- An LSP with **allow-sr-over-srte** explicitly configured is selected ahead of all other next hops, and will be selected as the first available next hop in the list of IP next hops.
- The IP next hops may be a mixture of enabled and disabled **allow-sr-over-srte** configurations, but only those SR-TE LSPs that have configured **allow-sr-over-srte** are copied as SR next hops. As such, the SR next hops may be a subset of the IP next hops.
- As with IP next hops, the configured TTM preference determines the preference between RSVP shortcuts and SR-TE shortcuts.
- As with IP next hops, a set of ECMP next hops cannot be a mixture of RSVP shortcuts and SR-TE shortcuts.

To verify the eligibility of an SR-TE LSP, you can review the TTM SR-TE tunnel flags.

```
show router tunnel-table protocol sr-te detail
```

Output example: Displaying SR-TE tunnel flags to verify SR-TE LSP eligibility as SR-MPLS IGP shortcut

```
===============================================================================
 Tunnel Table (Router: Base)
===============================================================================
 Destination : 10.20.1.6/32
 NextHop : 1.0.13.1 (524291, ospf (0))
 Tunnel Flags : is-over-tunnel entropy-label-capable allow-sr-over-sr-te
 Age : 00h01m12s
 CBF Classes : (Not Specified)
 Owner : sr-te                       Encap : MPLS Tunnel
 ID : 655366                         Preference : 8
 Tunnel Label : 524287               Tunnel Metric : 200
 Tunnel MTU : 1548                   Max Label Stack : 4
 LSP Weight : 0
 -------------------------------------------------------------------------------
 Number of tunnel-table entries : 1
 Number of tunnel-table entries with LFA : 0
 ===============================================================================
```

2.2.11 BGP shortcuts using SR-TE LSP

The user can forward packets of BGP prefixes over an SR-TE LSP programmed in TTM by configuring the following BGP shortcut tunnel binding command:

```
config>router>bgp>next-hop-resolution
        – shortcut-tunnel
            – [no] family {ipv4}
                – resolution {any | disabled | filter}
                – resolution-filter
                    – [no] sr-te
                – exit
            – exit
        – exit
```

2.2.12 BGP labeled route resolution using SR-TE LSP

The user can enable SR-TE LSP, as programmed in TTM, for resolving the next hop of a BGP IPv4 or IPv6 (6PE) labeled route by enabling the following BGP transport tunnel command:

```
config>router>bgp>next-hop-res>
        – labeled-routes
            – transport-tunnel
                – [no] family {label-ipv4 | label-ipv6 | vpn}
                    – resolution {any | disabled | filter}
                    – resolution-filter
                        – [no] sr-te
                    – exit
                – exit
            – exit
```

2.2.13 Service packet forwarding using SR-TE LSP

An SDP sub-type of the MPLS encapsulation type allows service binding to an SR-TE LSP programmed in the TTM by MPLS.

```
*A:Dut-A# configure service sdp 100 mpls create
    –config>service>sdp$ sr-te-lsp lsp-name
```

The user can specify up to 16 SR-TE LSP names. The destination address of all LSPs must match that of the SDP far-end option. Service data packets are sprayed over the set of LSPs in the SDP using the same procedures as for tunnel selection in ECMP. However, each SR-TE LSP can have up to 32 next hops at the ingress LER when the first segment is a node SID-based SR tunnel. Consequently, service data packets are forwarded over one of a maximum of 16×32 next hops. The **tunnel-far-end** option is not supported. In addition, the **mixed-lsp-mode** option does not support the **sr-te** tunnel type.

The signaling protocol for the service labels of an SDP that is using an SR-TE LSP can be configured to static (**off**), T-LDP (**tldp**), or BGP (**bgp**).

Use the following command syntax to configure an SR-TE LSP used in VPRN auto-bind.

```
config>service>vprn>
        – auto-bind-tunnel
            – resolution {any | disabled | filter}
            – resolution-filter
                – [no] sr-te
            – exit
        – exit
```

Both VPN-IPv4 and VPN-IPv6 (6VPE) are supported in a VPRN service using segment routing transport tunnels with the **auto-bind-tunnel** command.

Use the following command syntax with the BGP EVPN service.

```
config>service>vpls>bgp-evpn>mpls>
        – auto-bind-tunnel
            – resolution {any | disabled | filter}
            – resolution-filter
                – [no] sr-te
            – exit
```

```
        – exit
```

The following service contexts are supported with SR-TE LSP:

- VLL, LDP VPLS, IES/VPRN spoke-interface, R-VPLS, BGP EVPN
- BGP-AD VPLS, BGP-VPLS, BGP VPWS when the **use-provisioned-sdp** option is enabled in the binding to the PW template
- intra-AS BGP VPRN for VPN-IPv4 and VPN-IPv6 prefixes with both auto-bind and explicit SDP
- inter-AS options B and C for VPN-IPv4 and VPN-IPv6 VPRN prefix resolution
- IPv4 BGP shortcut and IPv4 BGP labeled route resolution
- IPv4 static route resolution
- multicast over IES/VPRN spoke interface with **spoke SDP** riding a SR-TE LSP

2.2.14 Datapath support

To support SR-TE in the datapath, the ingress LER must push a label stack where each label represents a hop, a TE link, or a node, in the ERO for the LSP path computed by the router or the PCE. However, only the label and the outgoing interface to the first strict or loose hop in the ERO factor into the forwarding decision of the ingress LER, because the SR-TE LSP only needs to track the reachability of the first strict or loose hop. This represents the NHLFE of the SR shortest path tunnel to the first strict or loose hop.

To ensure that its NHLFE is readily available, the SR OS stores the SR shortest path tunnel to a downstream node SID or adjacency SID in the tunnel table. The rest of the label stack is not meaningful to the forwarding decision. In this guide, "super NHLFE" refers specifically to this part of the label stack because it can have a much larger size.

An SR-TE LSP is modeled in the ingress LER datapath as a hierarchical LSP, with the super NHLFE tunneled over the NHLFE of the SR shortest path tunnel to the first strict or loose hop in the SR-TE LSP path ERO.

The following are characteristics of this model.

- The model saves on NHLFE usage. When many SR-TE LSPs travel to the same first hop, they ride the same SR shortest path tunnel, and each consumes one super NHLFE. However, the SR-TE LSPs point to a single NHLFE or set of NHLFEs if ECMP exists for the first strict or loose hop of the first-hop SR tunnel.

  In addition, the ingress LER does not need to program a separate backup super NHLFE. Instead, the single super NHLFE automatically begins forwarding packets over the LFA backup path of the SR tunnel to the first hop as soon as it is activated.
- When the path of a SR-TE LSP contains a maximum of two SIDs, that is the destination SID and one additional loose or strict-hop SID, the SR-TE LSP uses a hierarchy consisting of a regular NHLFE pointing to the NHLFE of top SID corresponding to the first loose or strict hop.
- If the first segment is a node SID tunnel and multiple next hops exist, ECMP spraying is supported at the ingress LER.
- If the first-hop SR tunnel, node, or adjacency SID goes down, the SR module informs MPLS that the outer tunnel is down, and MPLS brings the SR-TE LSP down and requests SR to delete the SR-TE LSP in the IOM.

The datapath behavior at the LSR and the egress LER for an SR-TE LSP is similar to that of a shortest path tunnel, because there is no tunnel state in these nodes. Packet forwarding is based on processing the

incoming label stack, consisting of a node SID or adjacency SID label, or both. If the ILM is for a node SID and multiple next hops exist, ECMP spraying is supported at the LSR.

The link-protect LFA backup next hop for an adjacency SID can be programmed at the ingress LER and LSR nodes. See SR-TE LSP protection for more information.

A maximum of 12 labels, including all transport, service, hash, and OAM labels, can be pushed. The label stack size for the SR-TE LSP can be 1 to 11 labels, with a default value of 6.

The maximum value of 11 is obtained for an SR-TE LSP whose path is not protected via FRR backup and with no entropy or hash label feature enabled when such an LSP is used as a shortcut for an IGP IPv4/IPv6 prefix or as a shortcut for BGP IPv4/IPv6. In this case, the IPv6 prefix requires pushing the IPv6 explicit-null label at the bottom of the stack. This leaves 11 labels for the SR-TE LSP.

The default value of 6 is obtained in the worst cases, such as forwarding a vprn-ping packet for an inter-AS VPN-IP prefix in Option C:

6 SR-TE labels + 1 remote LFA SR label + BGP 8277 label + ELI (RFC 6790) + EL (entropy label) + service label + OAM Router Alert label = 12 labels.

The label stack size manipulation includes the following LER and LSR roles:

LER role

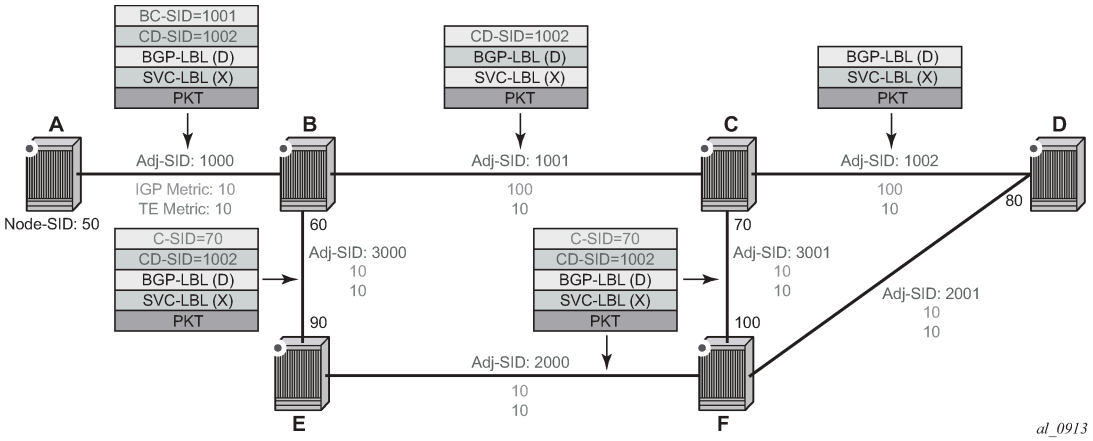
- push up to 12 labels
- pop-up to 8 labels of which 4 labels can be transport labels

LSR role

- pop-up to 5 labels and swap one label for a total of 6 labels
- LSR hash of a packet with up to 16 labels

The following figure shows an example of the label stack pushed by the ingress LER and by an LSR acting as a PLR.

Figure 25: SR-TE LSP label stack programming



On node A, the user configures an SR-TE LSP to node D with a list of explicit strict hops mapping to the adjacency SID of links A-B, B-C, and C-D.

Ingress LER A programs a super NHLFE consisting of the label for the adjacency over link C-D and points the NHLFE to the already programmed NHLFE of the SR tunnel of its local adjacency over link A-B. The latter NHLFE has the top label and also the outgoing interface to send the packet to.

Note:
SR-TE LSP does not consume a separate backup super NHLFE; it only points the single super NHLFE to the NHLFE of the SR shortest path tunnel it is riding. When the latter activates its backup NHLFE, the SR-TE LSP automatically forwards over it.

LSR node B has already programmed the primary NHLFE for the adjacency SID over link C-D and has the ILM with label 1001 point to it. In addition, node B preprograms the link-protect LFA backup next hop for link B-C and points the same ILM to it.

Note:
There is no super NHLFE at node B, because it only deals with programming the ILM and primary or backup NHLFE of its adjacency SIDs and its local and remote node SIDs.

VPRN service in node A forwards a packet to the VPN-IPv4 prefix X advertised by BGP peer D. Figure 25: SR-TE LSP label stack programming shows the resulting datapath at each node for the primary path and for the FRR backup path at LSR B.

2.2.14.1 SR-TE LSP metric and MTU settings

The MPLS module assigns the maximum LSP metric value (16777215) to a TE LSP when the local router provides the hop-to-label translation for its path. If a TE LSP uses the local CSPF (**path-computation-method local-cspf** option enabled) or PCE for path computation (**path-computation-method pce** option enabled) or delegates control to the PCE (**pce-control** enabled), the PCE returns the computed LSP IGP or TE metric in the PCReq and PCUpd messages.

In all cases, using the **config router mpls lsp metric** command to configure an admin metric overrides the returned value.

The MTU computations are as follows:

- The MTU setting of an SR-TE LSP is derived from the MTU of the outgoing SR shortest path tunnel it is riding, adjusted with the size of the super NHLFE label stack size.

  The following calculation is used:

  `SR_Tunnel_MTU = MIN {Cfg_SR_MTU, IGP_Tunnel_MTU- (1 + frr-overhead) × 4}`

  where:

  – `Cfg_SR_MTU` is the MTU configured for all SR tunnels within a specific IGP instance using the **config router ospf segment-routing tunnel-mtu** or **config router isis segment-routing tunnel-mtu** command. If no value is configured, the SR tunnel MTU is fully determined by the IGP interface calculation, described in the next bullet point.

  – `IGP_Tunnel_MTU` is the minimum of the IS-IS or OSPF interface MTU among all the ECMP paths or among the primary and LFA backup paths of the SR tunnel.

  – *frr-overhead* is set to:

    - value of **ti-lfa** [**max-sr-frr-labels** *labels*] if **loopfree-alternates** and **ti-lfa** are enabled in this IGP instance
    - 1 if **loopfree-alternates** and **remote-lfa** are enabled but **ti-lfa** is disabled in this IGP instance
    - 0 for all other cases

  This calculation is performed by IGP and passed to the SR module each time there is a change because of an updated resolution of the node SID.

The SR OS also provides the MTU for the adjacency SID tunnel because it is needed in an SR-TE LSP if the first hop in the ERO is an adjacency SID. In this case, the calculation for SR_Tunnel_MTU, initially introduced for a node SID tunnel, is applied to derive the MTU of the adjacency SID tunnel.

- The MTU of the SR-TE LSP is derived as follows:

  SRTE_LSP_MTU= SR_Tunnel_MTU - numLabels × 4

  where:

  – SR_Tunnel_MTU is the MTU SR tunnel shortest path that the SR-TE LSP is riding. The formula for this is provided in the previous bullet point.

  – numLabels is the number of labels found in the super NHLFE of the SR-TE LSP. At the LER, the super NHLFE points to the SR tunnel NHLFE, which itself has a primary and a backup NHLFE.

  This calculation is performed by the SR module and is updated each time the SR-TE LSP path changes or the SR tunnel it is riding is updated.

Note:
The above calculated SR-TE LSP MTU is used for the determination of an SDP MTU and for checking the Layer 2 service MTU. In the case of fragmentation of IP packets forwarded in GRT or in a VPRN over an SR-TE LSP, the IOM always deducts the worst-case MTU (12 labels) from the outgoing interface MTU when deciding whether to fragment the packet. In this case, the preceding formula is not used.

2.2.14.2 LSR hashing on SR-TE LSPs

The LSR supports hashing up to a maximum of 16 labels in a stack. The LSR is able to hash on the IP headers when the payload below the label stack is IPv4 or IPv6, including when a MAC header precedes it (**ethencap-ip** option). Alternatively, it is able to hash based only on the labels in the stack, which may include the entropy label (EL) or the hash label. See the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide* for more information about the hash label and entropy label features.

When the hash-label option is enabled in a service context, a hash label is always inserted at the bottom of the stack as per RFC 6391.

The EL feature, as specified in RFC 6790, indicates the presence of a flow on an LSP that should not be reordered during load balancing. It can be used by an LSR as input to the hash algorithm. The Entropy Label Indicator (ELI) is used to indicate the presence of the EL in the label stack. The ELI, followed by the actual EL, is inserted immediately below the transport label for which the EL feature is enabled. If multiple transport tunnels have the EL feature enabled, the ELI and EL are inserted below the lowest transport label in the stack.

The EL feature is supported with an SR-TE LSP. See the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide* for more information.

The LSR hashing operates as follows:

- If the **lbl-only** hashing option is enabled, or if one of the other LSR hashing options is enabled but an IPv4 or IPv6 header is not detected below the bottom of the label stack, the LSR parses the label stack and hashes only on the EL or hash label.
- If the **lbl-ip** option is enabled, the LSR parses the label stack and hashes on the EL or hash label and the IP headers.
- If the **ip-only** or **eth-encap-ip** is enabled, the LSR hashes on the IP headers only.

2.2.15 SR-TE Auto-LSP

The SR-TE auto-LSP feature supports auto-creation of the following types of LSPs:

- SR-TE mesh
- SR-TE one-hop
- SR-TE on-demand

The SR-TE mesh LSP feature binds an SR-TE mesh P2P LSP template with one or more prefix lists. When the TE database discovers a router that has an ID matching an entry in the prefix list, the database triggers MPLS to instantiate an SR-TE LSP to the router using the LSP parameters in the LSP template.

The SR-TE one-hop LSP feature activates an SR-TE one-hop P2P LSP template. In this case, the TE database tracks each TE link that is made to a directly connected IGP neighbor. The database then instructs MPLS to instantiate an SR-TE LSP with the following parameters:

- the source address of the local router
- an outgoing interface matching the interface index of the TE link
- a destination address matching the router ID of the neighbor on the TE link

In both these types of SR-TE auto-LSP, the router hop-to-label translation or local CSPF computes the label stack required to instantiate the LSP path.

The SR-TE on-demand LSP feature creates an LSP using an SR-TE on-demand P2P LSP template. When an imported BGP route matches an entry in a policy statement with an MPLS create tunnel action, an LSP is created to the next hop for the route. If a route admin tag policy is applied when the route is imported, only an auto-LSP with a template containing a matching **admin-tag** is created. The SR-TE on-demand LSP supports path computation using hop-to-label translation, local-CSPF, or a PCE.

Note:
An SR-TE mesh or one-hop auto-LSP can be reported to a PCE but cannot be delegated or have its paths computed by PCE. An SR-TE on-demand LSP can also be controlled and have its path computed by a PCE, as well as being reported to a PCE.

2.2.15.1 Feature configuration

About this task

This feature uses three SR-TE LSP template types: one-hop P2P, on-demand P2P, and mesh P2P. For the one-hop P2P and mesh P2P types, the configuration of the commands is the same as that of the RSVP-TE auto-LSP.

Procedure

Step 1. Create an LSP template using one of the following commands, depending on the type of auto-LSP required.

MD-CLI

```
configure router mpls lsp-template type p2p-sr-te-mesh
configure router mpls lsp-template type p2p-sr-te-one-hop
configure router mpls lsp-template type p2p-sr-te-on-demand
```

classic CLI

```
configure router mpls lsp-template mesh-p2p-srte
configure router mpls lsp-template one-hop-p2p-srte
configure router mpls lsp-template on-demand-p2p-srte
```

Step 2. In the template, configure the common LSP- and path-level parameters or options shared by all LSPs using this template.

Note:
These LSP template types contain the SR-TE LSP-specific commands and other LSP or path commands that are common to RSVP-TE and SR-TE LSPs and are supported by the existing RSVP-TE LSP template.

Step 3. Bind the LSP templates as follows:

- For the SR-TE mesh P2P LSP template, use the **configure router mpls lsp-template policy** command.
- For the SR-TE one-hop P2P LSP template, use the **configure router mpls lsp-template one-hop** command.
- For the on-demand SR-TE LSP template, bind the template to the creation of SR-TE auto-LSPs using the **configure router mpls auto-lsp** command and configure the **create-mpls-tunnel** command as an action in a route import policy statement.

See Configuring and operating SR-TE for an example of SR-TE auto-LSP creation using an LSP template type **mesh-p2p-srte**.

2.2.15.2 Automatic creation of an SR-TE mesh LSP

The **auto-lsp** command binds an LSP template of type **mesh-p2p-srte** with one or more prefix lists. When the TE database discovers a router that has a router ID matching an entry in the prefix list, it triggers MPLS to instantiate an SR-TE LSP to that router using the LSP parameters in the LSP template.

The prefix match can be exact or longest. Prefixes in the prefix list that do not correspond to a router ID of a destination node cannot match.

The path of the LSP is that of the default path name specified in the LSP template. The hop-to-label translation tool or the local CSPF determines the node SID and adjacency SID corresponding to each loose and strict hop in the default path definition, respectively.

The LSP has an auto-generated name using the following structure:

TemplateName-DestIpv4Address-TunnelId

where:

- *TemplateName* is the name of the template
- *DestIpv4Address* is the address of the destination of the auto-created LSP
- *TunnelId* is the TTM tunnel ID

In SR OS, an SR-TE LSP uses three different identifiers:

- LSP Index is used for indexing the LSP in the MIB table shared with RSVP-TE LSP. The LSP Index range is as follows:

  - provisioned SR-TE LSP: 65536 to 81920
  - SR-TE auto-LSP: 81921 to 131070
- LSP Tunnel ID is used in the interaction with the PCC or PCE. Range: 1 to 65536
- TTM Tunnel ID is the tunnel ID service, shortcut, and steering applications use to bind to the LSP. Range: 655362 to 720897

The path name is the default path specified in the LSP template.

Note: This feature is limited to SR-TE LSPs that are controlled by the router (PCC-controlled), where the path is provided using the hop-to-label translation or the local CSPF path computation method.

2.2.15.3 Automatic creation of an SR-TE one-hop LSP

Although the provisioning model and CLI syntax differ from that of a mesh LSP by the absence of a prefix list, the actual behavior is quite different. When the **one-hop-p2p** command is executed, the TE database keeps track of each TE link that comes up to a directly connected IGP neighbor. It then instructs the MPLS to instantiate an SR-TE LSP with the following parameters:

- the source address of the local router
- an outgoing interface matching the interface index of the TE link
- a destination address matching the router ID of the neighbor on the TE link

The hop-to-label translation or the local CSPF returns the SID for the adjacency to the remote address of the neighbor on this link. Therefore, the **auto-lsp** command binding an LSP template of type **one-hop-p2p-srte** with the **one-hop** option results in one SR-TE LSP instantiated to the IGP neighbor for each adjacency over any interface.

Because the local router installs the adjacency SID to a link regardless of whether the neighbor is SR-capable, the TE-DB finds the adjacency SID and a one-hop SR-TE LSP can still come up to such a neighbor. However, remote LFA using the neighbor's node SID does not protect the adjacency SID or the one-hop SR-TE LSP because the node SID is not advertised by the neighbor.

The LSP has an auto-generated name using the following structure:

TemplateName-DestIpv4Address-TunnelId

where:

- *TemplateName* = the name of the template
- *DestIpv4Address* = the address of the destination of the auto-created LSP
- *TunnelId* = the TTM tunnel ID

The path name is the default path specified in the LSP template.

Note:
This feature is limited to an SR-TE LSP that is controlled by the router (PCC-controlled) and the path labels are provided by the hop-to-label translation or the local CSPF path computation method.

2.2.15.4 Automatic creation of an on-demand SR-TE LSP

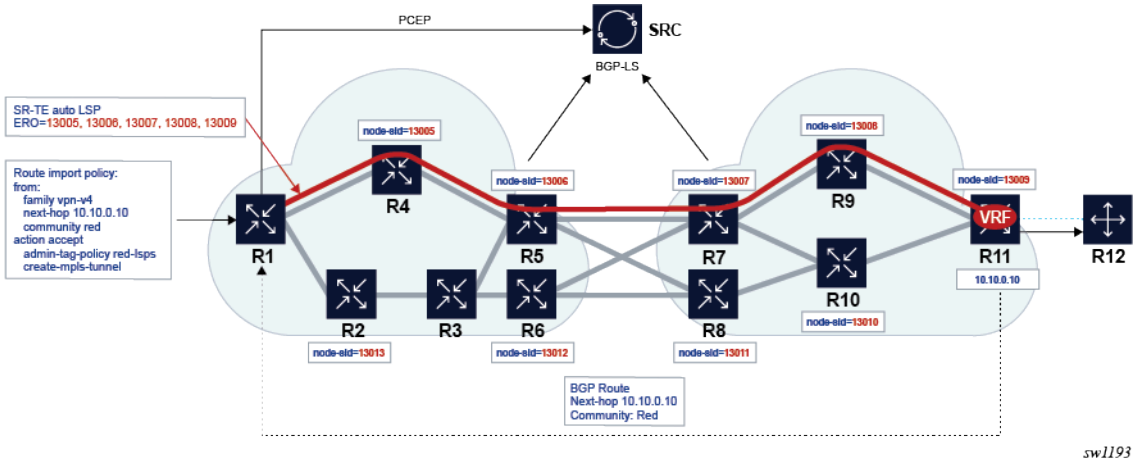
The SR-TE on-demand LSP simplifies provisioning for networks that may or may not be managed by a network service manager, such as the Nokia NSP. Instead of using a full mesh, LSPs can be automatically created on-demand when a suitable tunnel does not exist for a specific BGP prefix next hop. The prefix could be for VPRN, EVPN, BGP-LU, or BGP shortcut routes. Both intradomain and inter-domain use cases are supported.

This mechanism is an extension of the LSP **admin-tag** and auto-LSP mechanisms and applies to the following objects:

- VPRN auto-bind-tunnel
- EVPN VPWS auto-bind-tunnel
- EVPN VPLS auto-bind-tunnel
- BGP-LU, both as an LER and LSR at an ABR or ASBR
- BGP shortcuts

The following figure shows an application of SR-TE on-demand LSPs.

Figure 26: VPRN example of an on-demand SR-TE LSP



This example combines route transport coloring and auto LSPs to simplify provisioning for intent-based networking for specific services. In this use case, intent means the ability to meet traffic engineering requirements for a service. This could be, for example, a delay or loss, or the ability to steer the service traffic to avoid LSPs that transit specific geographies or prefer those that take another route.

In this example, a BGP route is advertised for a VRF in a PE for a VPRN service. An extended color community is assigned to the route. This color implies an intent associated with the transport requirements.

When this route is imported at the head-end PE, the router performs the following steps:

1. The route is matched in a route-import policy.
2. An **admin-tag** policy called red-lsps is applied.
3. A trigger action occurs to create an MPLS tunnel to the BGP next-hop for the route.

This causes the head-end router to create an SR-TE auto-LSP that matches the red-lsps **admin-tag** policy and steers the traffic associated with "red" routes to the far-end PE, into the red LSP. This SR-TE auto-LSP is created based on the configuration in the matching LSP template.

SR OS also offers the ability to use the local CSPF, hop-to-label-translation, or a PCE to provide a path for the red LSP. This is determined by a configuration in the matching LSP template.

2.2.15.4.1 Deletion of on-demand SR-TE LSPs

SR-TE on-demand P2P auto LSPs are removed by the router in all the following cases.

- The classic CLI **no auto-lsp** (or MD-CLI **delete auto-lsp**) command is executed. This triggers MPLS to remove auto-LSPs created by this command.
- The **no create-mpls-tunnel** is configured in a policy statement that previously had **create-mpls-tunnel** configured. This triggers a reevaluation of the policy statement and potentially triggers BGP to inform MPLS that it no longer needs a tunnel.
- BGP tracks the binding of a route to an **admin-tag-policy**. If an **admin-tag-policy** name in a policy statement action changes, the policy is reevaluated, which could change the binding. This may result in a request to create a new tunnel or delete an existing tunnel. However, if the contents of an **admin-tag-policy** that is referenced in a policy statement action change, BGP does not react (for example, request the creation or deletion of a tunnel), although a subsequent route resolution may change.
- MPLS reacts to **admin-tag** changes in the LSP template. When this occurs, it reevaluates the **admin-tag-policy** associated with a request from BGP and deletes or creates tunnels accordingly.
- If a new LSP is created that is not an on-demand LSP and is preferred to an existing on-demand LSP, BGP can resolve the next hop over the new LSP and traffic moves to it. In this case, the system does not remove the older less-preferred auto-LSP, which was created through an on-demand LSP trigger, until the next hops are removed.
- If the LSP template is shut down, all associated LSPs are administratively disabled. To delete the LSP template you must first shut it down, using a **no auto-lsp** command in classic CLI or **delete auto-lsp** command in MD-CLI. This removes all the auto-LSPs that are using the template.

2.2.15.4.2 Configuring SR-TE on-demand LSPs

About this task

Configure SR-TE on-demand LSPs using the steps in this section.

Procedure

Step 1. Define a policy statement to import the route, as shown in the following example:

Example

```
configure>router>policy-options>policy-statement

   entry
      from
        family <family>
        next-hop <ip-address>
        community <comm-name>
      action accept
         admin-tag-policy <admin-tag-policy-name>
```

```
        create-mpls-tunnel
```

Step 2. Configure the auto-LSP under MPLS with the template type **on-demand-p2p-srte**.

The **create-mpls-tunnel** action is supported for the following address families:

- vpn-ipv4
- vpn-ipv6
- evpn
- label-ipv4
- label-ipv6
- ipv4
- ipv6

The router-policy action assigns an **admin-tag-policy** to the routes that are imported with a specific next hop and match a specified extended community. In most applications, the extended community is the transport color extended community. The **create-mpls-tunnel** command action causes BGP to send the next hop and the include and exclude constraints in the **admin-tag-policy** (if one was assigned to a route by the policy statement) to the MPLS application.

When such a policy statement is applied in the context of a specific VRF, the **create-mpls-tunnel** command trigger is only actioned by BGP on a per-next-hop basis.

This type of LSP template supports PCE computation, control, and the fallback path computation method if the PCE is unreachable. The auto-LSP is configured using the following command:

```
configure>router>mpls
     auto-lsp <on-demand-p2p-srte-template-name>
```

The LSP template may contain an LSP **admin-tag-policy**. MPLS takes the next hop, and the **admin-tag** command includes or excludes constraints from BGP and matches them against the auto-LSP statement with a template with an **admin-tag** command that conforms to the **admin-tag-policy** constraints.

If BGP does not pass any **admin-tag-policy** constraints, MPLS only matches against LSP templates that do not have the **admin-tag** command configured.

If the next-hop and **admin-tag-policy** match more than one auto-LSP statement, an LSP is created for each matching entry. This results in an ECMP set to the next hop.

Note:
Each LSP may have a different **admin-tag** value, but it is an ECMP next-hop tunnel from the perspective of the colored route that triggers the tunnel creation.

A new SR-TE LSP is consequently created to the next hop passed by BGP according to the parameters contained in the LSP template.

The router tracks the binding between BGP triggers and on-demand LSPs that are successfully created and deleted toward a specified BGP next-hop matching an **admin-tag-policy**.

2.2.15.5 Interaction with PCEP

A template-based SR-TE auto-LSP, with the exception of on-demand SR-TE LSPs, can only be operated as a PCC-controlled LSP. It can, however, be reported to the PCE using the **pce-report** command. It

cannot be operated as a PCE-computed or PCE-controlled LSP. This is the same interaction with PCEP as that of a template-based RSVP-TE LSP.

On-demand SR-TE LSPs can be reported to the PCE and operated as PCE-computed or PCE-controlled LSPs.

The auto LSP can be delegated to a PCE in the configuration of the SR-TE on-demand P2P LSP template, using the **pce-control** or **path-computation-method pce** commands.

Fallback to local CSPF or hop-to-label translation is also supported for SR-TE on-demand P2P LSPs in case the PCE becomes unreachable.

In general, an on-demand SR-TE auto-LSP that is PCE controlled or has path computation method PCE is treated as any other PCC-initiated LSP by PCC. Path profile and path group are also supported. Path profile and path group IDs are passed to the PCC in the same way as for a PCC-initiated SR-TE LSP.

2.2.15.6 Forwarding contexts supported with SR-TE auto-LSP

The following are the forwarding contexts that can be used by an auto-LSP:

- resolution of IPv4 BGP labeled routes and IPv6 BGP labeled routes (6PE) in the TTM
- resolution of IPv4 BGP route in the TTM (BGP shortcut)
- resolution of IPv4 static route to indirect next hop in the TTM
- VPRN and BGP-EVPN auto-bind for both IPv4 and IPv6 prefixes

The auto-LSP cannot be used in a provisioned SDP for explicit binding by services. Therefore, an auto-LSP can also not be used directly for auto-binding of a PW template with the **use-provisioned-sdp** option in BGP-AD VPLS or FEC129 VLL service. However, an auto-binding of a PW template to an LDP LSP, which is then tunneled over an SR-TE auto-LSP is supported.

2.2.16 Allocation and binding of labels to SR-TE LSPs

SR OS supports the allocation and binding of labels to SR-TE LSPs. The LSPs have to be named LSPs or LSPs based on a template, whether the LSPs are PCC initiated (**on-demand-p2p-srte** template type) or PCE initiated (**pce-init-p2p-srte** template type).

The result of a binding SID label is the programming of an ILM with a swap operation pointing to the LSP NHLFE.

A single binding SID label can be allocated to a specific LSP.

Named LSPs

Use the commands in the following context to configure named LSPs.

```
configure router mpls lsp
```

Use the following command to configure the binding SID label value for named LSPs.

```
configure router mpls lsp binding-sid
```

The value of the binding SID label must be within the label block that is reserved for binding SID labels. The reserved label block is configured like any other reserved block. Use the following command to reference the reserved label block for statically configured binding SIDs.

```
configure router mpls lsp-bsid-block
```

A binding SID label can be assigned or removed at any time. In the case where the LSP is delegated to a PCE, the appropriate messages are triggered in both situations (assignment and removal).

The node that allocates the label is considered to be the owner; therefore, the PCE cannot change the binding SID label.

PCC-initiated LSPs

To enable the allocation of a binding SID to LSPs created via the **on-demand-p2p-srte** template, the user must configure the template. Use the following command to configure the template:

```
configure router mpls lsp-template binding-sid
```

When enabled, the system dynamically selects a label from the dynamic label range.

The binding SID label is only allocated to LSPs if the template also has **pce-report** enabled.

The node allocated the label is considered to be the owner; therefore, the PCE cannot change the binding SID label.

PCE-initiated LSPs

There is no command to configure binding SIDs for PCE-initiated LSPs. PCE-initiated LSPs support binding SID labels by default. The PCE initiates the allocation of labels. The only supported mode of operation is for the PCE to request the node to select a label and program it. It is not supported for the PCE to suggest a label value. The node tries to select a label in the dynamic label range and programs it.

The PCE is the initiator of the label allocation and can also request the deallocation of the label.

2.2.17 SR-TE LSP traffic statistics

The collection of traffic statistics on SR-TE LSPs using either a named LSP or SR-TE templates is available on egress of ingress LER. Also, traffic statistics cannot be recorded into an accounting file.

SR-TE LSP statistics are provided without any forwarding class or QoS profile distinction. However, traffic statistics are recorded and made available for each path of the LSP (primary and backup). Statistic indexes are only allocated at the time the path is effectively programmed, are maintained across switchover for primary and standby LSPs only, and are released if egress statistics are disabled or the LSP is deleted.

Note: SR-TE LSP egress statistics are not supported on VSR.

2.2.17.1 Rate statistics

About this task

SR OS also provides traffic rate statistics.

The frequency at which rate statistics are determined is configured in the accounting policy using the **collection-interval** command. The minimum interval is 5 minutes.

Rate statistics for SR-TE LSPs cannot be written to an accounting file. The **to no-file** command must be configured in the accounting policy.

Rate statistics are provided in pkt/s and Mb/s. Rate statistics are provided as an aggregate across all paths of the LSP for which a statistical index has been assigned, and for all forwarding classes in- or out-of-profile.

Rate statistics are only available on egress of the ingress LER. At least two samples are needed to determine a rate.

For SR-TE LSPs, including template-based LSPs, the user enables this capability by performing the following tasks:

Procedure

Step 1. Configure an accounting policy that uses the record **combined-mpls-srte-egress**.

Step 2. Assign the configured accounting policy to a specific LSP (or template).

Step 3. Enable stats collection.

2.2.18 SR-TE label stack checks

This section describes the SR-TE label stack checks.

2.2.18.1 SR-TE label stack check for services and shortcuts

If a packet forwarded in a service results in the net label stack size being pushed on the packet to exceed the maximum label stack supported by the router, the packet is dropped on the egress. Each service and shortcut application on the router performs a check of the resulting net label stack after pushing all the labels required for forwarding the packet in that context.

To that effect, the MPLS module populates each SR-TE LSP in the TTM with the maximum transport label stack size, which consists of the sum of the values in **max-sr-labels** *label-stack-size* and **additional-frr-labels** *labels*.

Each service or shortcut application then adds the additional, context-specific labels such as service label, entropy or hash label, and control-word, required to forward the packet in that context and check that the resulting net label stack size does not exceed the maximum label stack supported by the router.

If the check succeeds, the service binds or the prefix resolves to the SR-TE LSP.

If the check fails, the service does not bind to this SR-TE LSP. Instead, the service either finds another SR-TE LSP or another tunnel type to bind to, if the user has configured other tunnel types. Otherwise, the service goes down. If the service uses an SDP with one or more SR-TE LSP names, the spoke SDP bound to this SDP remains operationally down as long as at least one SR-TE LSP fails the check. In this case, a new spoke SDP flag is displayed in the show output of the service: `labelStackLimitExceeded`. Similarly, the prefix is not resolved to the SR-TE LSP and is either resolved to another SR-TE LSP or another tunnel type, or becomes unresolved.

The value of **additional-frr-labels** *labels* is checked against the maximum value across all IGP instances of the *frr-overhead parameter*. This parameter is computed within a specific IGP instance. The following table lists the parameter values.

Table 8: Values of the frr-overhead parameter

Condition	frr-overhead parameter value
segment-routing is disabled in the IGP instance	0
segment-routing is enabled but **remote-lfa** is disabled	0
segment-routing is enabled and **remote-lfa** is enabled	1

If the user configures or changes the configuration of the **additional-frr-labels** command, MPLS ensures that the new value accommodates the *frr-overhead* value across all IGP instances. Consider the sequence in the following example:

1. The user configures the **config router isis loopfree-alternate remote-lfa** command.
2. The user creates an SR-TE LSP or changes the configuration of an existing one, as follows:

   mpls>**lsp**>**max-sr-labels** *10* **additional-frr-labels** *0*.

Note:
Performing a **no shutdown** of the new LSP or changing the existing LSP configuration is blocked because the IS-IS instance enabled remote LFA, which requires one additional label on top of the 10 SR labels of the primary path of the SR-TE LSP.

If the check is successful, MPLS adds **max-sr-labels** and **additional-frr-labels** and checks that the value is less than or equal to the maximum label stack supported by the router. MPLS then populates the value of {**max-sr-labels** + **additional-frr-labels**}, along with tunnel information in TTM, and also passes **max-sr-labels** to the PCEP module.

Conversely, if the user attempts a configuration change that may result in a change to the computed *frr-overhead* value, IGP checks that all SR-TE LSPs can properly account for the overhead; if the check fails, the change is rejected. On the IGP, enabling the **remote-lfa** command may cause the *frr-overhead* value to change. Consider the sequence in the following example:

1. An MPLS LSP is administratively enabled and has **mpls**>**lsp**>**max-sr-labels** *10* **additional-frr-overhead** *0* configured.
2. The current configuration in IS-IS has the **loopfree-alternate** command disabled.
3. The user attempts to configure **isis loopfree-alternate remote-lfa**. This changes the *frr-overhead* value to 1.

   This configuration change is blocked.

2.2.18.2 Control plane handling of egress label stack limitations

As described in Datapath support, the egress IOM can push a maximum of 12 labels; however, this number may be reduced if other fields are pushed into the packets. For example, for a VPRN service, the ingress LER can send an IP VPN packet with 12 labels in the stack, including one service label, one label for OAM, and 10 transport labels. However, if entropy is configured, the number of transport labels is reduced by two (Entropy Label (EL) and Entropy Label Indicator (ELI)). Similarly, for EVPN services, the egress IOM may push specific fields that reduce the total number of supported transport labels.

To avoid silent packet drops in cases where the egress IOM cannot push the required number of labels, SR OS implements a set of procedures that prevent the system from sending packets if it is determined

that the SR-TE label stack to be pushed exceeds the number of bytes that the egress IOM can put on the wire.

The following table describes the label stack egress IOM restrictions on FP-based hardware for IPVPN and EVPN services.

Table 9: Label stack egress IOM restrictions on FP-based hardware for IPVPN and EVPN services

Features that reduce the label stack		Source service type				
		IP-VPN (VPRN)	EVPN-IFL (VPRN)	EVPN VPLS or EVPN Epipe	EVPN B-VPLS (PBB-EVPN)	EVPN-IFF (R-VPLS)
Always Computed[3]	Service Label	1	1	1	1	1
	OAM Label	1	1	0	0	0
	Control Word	0	0	1	1	1
	ESI Label	0	0	1	0	0
Computed if configured[4]	Hash Label (mutex with EL)	1	1	0	0	0
	Entropy EL+ELI	2	2	2	2	2
Required Labels[5]		2	2	3	2	2
Required Labels + Options[5]		4	4	5	4	4
Maximum available labels[6]		12	12	10	6	9
Maximum available transport labels without options[7]		10	10	7	4	7
Maximum available transport labels with options[7]		8	8	5	2	5

3 These rows indicate the number of labels that the system assumes are always used on a specific service. For example, the system always computes two labels to be reduced from the total number of labels for VPRN services with EVPN-IFL (EVPN Interface-less model enabled).

4 These rows indicate the number of labels that the system subtracts from the total only if they are configured on the service. For example, on VPRN services with EVPN-IFL, if the user configures hash-label, the system computes one additional label. If the user configures entropy-label, the system deducts two labels instead.

5 These rows indicate the number of labels that the system deducts from the total number.

6 This row indicates a different number depending on the service type and the inner encapsulation used by each service, which reduces the maximum number of labels to push on egress. For example, while the number of labels for VPRN services is 12, the maximum number for VPLS and Epipe services is 10 (to account for space for an inner Ethernet header).

7 This row indicates the maximum SR-TE labels that the system can push when sending service packets on the wire.

The total number of labels configured in the **max-sr-labels** *label-stack-size* [**additional-frr-labels** *labels*] command must not exceed the labels indicated in the *Maximum available transport labels without options* and *Maximum available transport labels with options* rows in the preceding table. If the configured LSP labels exceed the available labels listed in the preceding table, the BGP route next hop for the LSP is not resolved and the system does not even try to send packets to that LSP.

For example, for a VPRN service with EVPN-IFL where the user configures entropy-label, the maximum available transport labels is eight. If an IP Prefix route for next-hop X is received for the service and the SR-TE LSP to-X is the best tunnel to reach X, the system checks that (**max-sr-labels** + **additional-frr-labels**) is less than or equal to eight. Otherwise, the IP Prefix route is not resolved.

The same control plane check is performed for other service types, including IP shortcuts, spoke SDPs on IP interfaces, spoke SDPs on Epipes, VPLS, B-VPLS, R-VPLS, and R-VPLS in I-VPLS or PW-SAP. In all cases, the spoke SDP is brought down if the configured (**max-sr-labels** + **additional-frr-labels**) is greater than the maximum available transport labels. The following table indicates the maximum available transport labels for IP shortcuts and spoke SDP services.

Note: For PW-SAPs, the maximum available labels differ depending on the type of service PW-SAP used (Epipe or VPRN interface).

Table 10: Maximum available transport labels for IP shortcuts and spoke SDP services

Features that reduce the label stack		**Source service type**							
		IP shortcuts	**Spoke-sdp interface**	**Spoke-sdp Epipe**	**Spoke-sdp VPLS**	**Spoke-sdp B-VPLS**	**Spoke-sdp R-VPLS**	**Spoke-sdp R-VPLS I-VPLS**	**PW-SAP Epipe/ interface**
Always Computed[8]	Service Label	0	1	1	1	1	1	1	1/1
	OAM Label	0	1	1	1	1	0	0	0/0
	IPv6 label	1	0	0	0	0	0	0	0/0
Computed if configured[9]	Hash Label (mutex with EL)	0	1	1	1	1	1	0	0/0
	Entropy EL +ELI	2	2	2	2	2	2	2	0/0
	Control Word	0	1	1	1	1	1	1	0/0
Required Labels[10]		1	2	2	2	2	1	1	1/1

8 Indicates the number of labels that the system assumes are always used on a specific service

9 Indicates the number of labels that the system subtracts from the total only if they are configured on the service

10 Number of labels that the system deducts from the total number

Features that reduce the label stack	Source service type							
	IP shortcuts	Spoke-sdp interface	Spoke-sdp Epipe	Spoke-sdp VPLS	Spoke-sdp B-VPLS	Spoke-sdp R-VPLS	Spoke-sdp R-VPLS I-VPLS	PW-SAP Epipe/ interface
Required Labels + Options[10]	3	5	5	5	5	4	4	1/1
Maximum available labels[11]	12	9	10	10	6	8	4	10/7
Maximum available transport labels without options [12]	11	7	8	8	4	7	3	9/7
Maximum available transport labels with options[12]	9	4	5	5	1	4	0	8/6

In general, the labels shown in Table 9: Label stack egress IOM restrictions on FP-based hardware for IPVPN and EVPN services and Table 10: Maximum available transport labels for IP shortcuts and spoke SDP services are valid for network ports that are null or dot1q encapsulated. For QinQ network ports, the available labels are deducted by one.

2.2.18.3 Flexible SR-TE label stack allocation for services

SR OS supports a dynamic egress label limit configuration mode that extends the number of allowed MPLS labels in the egress label stack by not counting specific labels in the BGP next-hop resolution check when those labels are not used. The configuration mode exists in EVPN services configured on Epipe, VPLS, and VPRN (EVPN-IFL), and in IP-VPN services.

- **Classic CLI**

```
+-- dynamic-egress-label-limit
    no dynamic-egress-label-limit
```

- **MD CLI**

```
+-- dynamic-egress-label-limit <boolean>
```

When the **dynamic-egress-label-limit** command is configured, the always-computed labels are no longer considered when resolving the next hop of the route. As a result, the following rules apply to the specified services:

11 Indicates a different number depending on the service type and the inner encapsulation used by each service, which reduces the maximum number of labels to push on egress

12 Maximum SR-TE labels that the system can push when sending service packets on the wire

- For VPRN services, the OAM label is never computed. This is true whether the BGP next hop is being resolved over an auto-bind tunnel or an SDP in the **vprn**>**spoke-sdp** context. The dynamic mode is supported for EVPN-IFL and IP-VPN families.
- For EVPN (Epipe or VPLS) services with **dynamic-egress-label-limit** configured, the control word (CW) and ESI label are only computed if they are used.
  - In the case of the CW, the system reduces the egress label limit by one label when the CW is configured in the service. The CW is always accounted when the **dynamic-egress-label-limit** command is not configured.
  - When the **dynamic-egress-label-limit** command is configured, the ESI label is not accounted for in Epipes or VPLS services without an ES; however, the ESI label is always accounted if **dynamic-egress-label-limit** is not configured.

When **no dynamic-egress-label-limit** is configured, the behavior follows the procedures described in Control plane handling of egress label stack limitations.

In summary, when the **dynamic-egress-label-limit** is configured, the total amount of labels (X) configured in X= (max-sr-labels Y + additional-frr-labels Z) can go higher for EVPN and IP-VPN services.

The following table summarizes the required behavior.

Table 11: Egress label stack limits for BGP services based on dynamic-egress-label-limit

Features that reduce the Label Stack		**no dynamic-egress-label-limit**			**dynamic-egress-label-limit**		
		IP-VPN (VPRN)	**EVPN-IFL (VPRN)**	**EVPN VPLS EVPN Epipe**	**IP-VPN (VPRN)**	**EVPN-IFL (VPRN)**	**EVPN VPLS EVPN Epipe**
Always computed	Service label	1	1	1	1	1	1
	OAM label [13]	1	1	0	0	0	0
	CW	0	0	1	0	0	0
	ESI label	0	0	1	0	0	0
Computed if configured	Hash label (mutex with EL)	1	1	0	1	0	0
	Entropy EL +ELI	2	2	2	2	2	2
	CW	0	0	0	0	0	1

[13] **vprn-ping** and **vprn-trace** commands are not supported when the **dynamic-egress-label-limit** command is configured.

Features that reduce the Label Stack		no dynamic-egress-label-limit			dynamic-egress-label-limit		
		IP-VPN (VPRN)	EVPN-IFL (VPRN)	EVPN VPLS EVPN Epipe	IP-VPN (VPRN)	EVPN-IFL (VPRN)	EVPN VPLS EVPN Epipe
	ESI label [14]	0	0	0	0	0	1
Required labels		2	2	3	1	1	1
Required labels + All Options		4	4	5	3	3	5
Maximum available labels		12	12	10	12	12	10
Maximum available transport labels without options		10	10	7	11	11	9
Maximum available transport labels with options		8	8	5	9	9	5

R-VPLS and B-VPLS services, with EVPN-MPLS enabled, also support the **dynamic-egress-label-limit** command when **dynamic-egress-label-limit** is configured, the CW is accounted for only if the **control-word** command is added.

- In R-VPLS services, the ESI label is not accounted because the routed encapsulation is always larger (and either ESI label for bridged traffic, or routed traffic without ESI label is transmitted by the R-VPLS).
- In B-VPLS services, only the CW applies; there is no ESI label.

2.2.19 IPv6 traffic engineering

This feature extends the traffic engineering capability with the support of IPv6 TE links and nodes.

This feature enhances IS-IS, BGP-LS, and the TE database with the additional IPv6 link TLVs and TE link TLVs. The following modes of operation are provided for IPv4 and IPv6 traffic engineering in a network.

- **Legacy mode**

  This mode of operation enables the existing traffic engineering behavior for IPv4 RSVP-TE and IPv4 SR-TE. Only the RSVP-TE attributes are advertised in the legacy TE TLVs that are used by both RSVP-TE and SR-TE LSP path computation in the TE domain routers. In addition, IPv6 SR-TE LSP path computation can now use these common attributes.

- **Legacy mode with application indication**

[14] When the **dynamic-egress-label-limit** command is configured, the ESI label is only accounted in EVPN VPLS services that have a SAP or SDP-bind associated to an ES.

This mode of operation is intended for cases where link TE attributes are common to RSVP-TE and SR-TE applications and have the same value, but the user wants to indicate on a per-link basis which application is enabled.

Routers in the TE domain use these attributes to compute paths for IPv4 RSVP-TE LSP and IPv4/IPv6 SR-TE LSP.

- **Application-specific mode**

  This mode of operation is intended for future use cases where TE attributes may have different values in RSVP-TE and SR-TE applications or are specific to one application (for example, RSVP-TE uses the Unreserved Bandwidth and Max Reservable Bandwidth attributes).

  SR OS does not support configuring TE attributes that are specific to the SR-TE application. As a result, enabling this mode advertises the common TE attributes a single time using the Application Specific Link Attributes TLV. Routers in the TE domain use these attributes to compute paths for IPv4 RSVP-TE LSP and IPv4/IPv6 SR-TE LSP.

See IS-IS IPv4/IPv6 SR-TE and IPv4 RSVP-TE feature behavior for more information about the IPv4 and IPv6 Traffic Engineering modes of operation.

This feature also adds support of IPv6 destinations to the SR-TE LSP configuration. In addition, this feature extends the MPLS path configuration with hop indexes that include IPv6 addresses.

IPv6 SR-TE LSP is supported with the hop-to-label and the local CSPF path computation methods. This requires the enabling of the IPv6 traffic engineering feature in IS-IS.

2.2.19.1 Global configuration

To enable IPv6 TE on the router, the IPv6 TE router ID must have a valid IPv6 address. Use the following CLI command to configure the IPv6 TE router ID:

configure>router>ipv6-te-router-id interface *interface-name*

The IPv6 TE router ID is a mandatory parameter that uniquely identifies the router as being IPv6 TE capable to other routers in an IGP TE domain. IS-IS advertises this information using the IPv6 TE Router ID TLV as described in TE attributes supported in IGP and BGP-LS.

When the command is not configured or the **no** form of the command is configured, the value of the IPv6 TE router ID reverts to the preferred primary global unicast address of the system interface. The user can also explicitly enter the name of the system interface to achieve the same outcome.

In addition, the user can specify a different interface and the preferred primary global unicast address of that interface is used instead. Only the system or a loopback interface is allowed because the TE router ID must use the address of a stable interface.

This address must be reachable from other routers in a TE domain and the associated interface must be added to IGP for reachability. Otherwise, IS-IS withdraws the advertisement of the IPv6 TE router ID TLV.

When configuring a new interface name for the IPv6 TE router ID, or when the same interface begins using a new preferred primary global unicast address, IS-IS immediately floods the new value.

If the referenced system is shut down or the referenced loopback interface is deleted or is shut down, or the last IPv6 address on the interface is removed, IS-IS withdraws the advertisement of the IPv6 TE router ID TLV.

2.2.19.2 IS-IS configuration

To enable the advertisement of additional link IPv6 and TE parameters, use the following **traffic-engineering-options** CLI syntax.

```
configure
     router
          ipv6-te-router-id interface interface-name
          no ipv6-te-router-id
          [no] isis [instance]
               traffic-engineering
               no traffic-engineering
               traffic-engineering-options
               no traffic-engineering-options
                    ipv6
                    no ipv6
                    application-link-attributes
                    no application-link-attributes
                         legacy
                         no legacy
```

The **traffic-engineering** command is the main CLI command used to enable TE in an IS-IS instance. This command enables the advertisement of the IPv4 and TE link parameters using the legacy TE encoding, in accordance with RFC 5305. These parameters are used in IPv4 RSVP-TE and IPv4 SR-TE.

When the **ipv6** command under the **traffic-engineering-options** context is enabled, the traffic engineering behavior with IPv6 TE links is enabled. This IS-IS instance automatically advertises the new RFC 6119 IPv6 and TE TLVs and sub-TLVs, as described in TE attributes supported in IGP and BGP-LS.

The **application-link-attributes** context allows the advertisement of the TE attributes of each link on a per-application basis. Two applications are supported in SR OS: RSVP-TE and SR-TE. The legacy mode of advertising TE attributes that is used in RSVP-TE is supported. Use the **no legacy** command to disable the legacy mode and also enable the per-application TE attribute advertisement for RSVP-TE.

See IS-IS IPv4/IPv6 SR-TE and IPv4 RSVP-TE feature behavior for more information about feature behavior and the interaction of the preceding CLI commands.

2.2.19.3 MPLS configuration

The SR-TE LSP configuration can accept an IPv6 address in the **to** and **from** parameters.

In addition, the MPLS path configuration can accept a hop index with an IPv6 address. The IPv6 address used in the **from** and **to** commands in the IPv6 SR-TE LSP, as well as the address used in the **hop** command of the path used with the IPv6 SR-TE LSP must correspond to the preferred primary global unicast IPv6 address of a network interface or a loopback interface of the corresponding LER or LSR router. The IPv6 address can also be set to the system interface IPv6 address. Failure to follow the preceding IPv6 address guidelines for the **from**, **to**, and **hop** commands causes path computation to fail with failure code "noCspfRouteToDestination".

A link-local IPv6 address of a network interface is also not allowed in the **hop** command of the path used with the IPv6 SR-TE LSP.

All other MPLS-level, LSP-level, and primary or secondary path-level configuration parameters available for an IPv4 SR-TE LSP are supported.

2.2.19.4 IS-IS, BGP-LS, and TE database extensions

IS-IS control plane extensions add support for the following RFC 6119 TLVs in IS-IS advertisements and TE-DB.

- IPv6 interface Address TLV (ISIS_TLV_IPv6_IFACE_ADDR 0xe8)
- IPv6 Neighbor Address sub-TLV (ISIS_SUB_TLV_NBR_IPADDR6 0x0d)
- IPv6 Global Interface Address TLV (only used by ISIS in IIH PDU)
- IPv6 TE Router ID TLV
- IPv6 SRLG TLV

IS-IS also supports the use of the Application Specific Link Attributes (ASLA) sub-TLV to advertise the protocol enabled on a specific TE-link (SR-TE, RSVP-TE, or both), as defined in *draft-ietf-isis-te-app*. The advertising router sends potentially different link TE attributes for RSVP-TE and SR-TE applications, and the router receiving the link TE attributes can identify the application that is enabled on the advertising router. For backward compatibility, the router continues to support the legacy mode of advertising link TE attributes, as recommended in RFC 5305, but the user can disable this mode.

Note:
SR OS does not support configuring and advertising different link TE attribute values for RSVP-TE and SR-TE applications. The router advertises the same link TE attributes for both RSVP-TE and SR-TE applications.

See IS-IS IPv4/IPv6 SR-TE and IPv4 RSVP-TE feature behavior for more information about the behavior of the per-application TE capability.

These TLVs and sub-TLVs are advertised in IS-IS and added into the local TE-DB when received from IS-IS neighbors. In addition, if the **database-export** command is enabled in this ISIS instance, this information is also added in the Enhanced TE-DB.

The following enhancements are added to support advertisement of TE parameters in BGP-LS routes over an IPv4 or IPv6 transport.

- Importing IPv6 TE link TLVs from a local Enhanced TE-DB into the local BGP process for exporting to other BGP peers using the BGP-LS route family that is enabled on an IPv4 or an IPv6 transport BGP session
  - RFC 6119 IPv6 and TE TLVs and sub-TLVs are carried in the BGP-LS link NLRI, as per RFC 7752.
  - When the link TE attributes are advertised by IS-IS on a per-application basis using the ASLA TLV (ISIS TLV Type 16), they are carried in the new BGP-LS ASLA TLV (TLV Type TBD), in accordance with *draft-ietf-idr-bgp-ls-app-specific-attr*.
  - When a TE attribute of a link is advertised for both RSVP-TE and SRTE applications, there are three methods IS-IS can use. Each method results in a specific way the BGP-LS originator carries this information. These methods are summarized here, but more information is provided in IS-IS IPv4/IPv6 SR-TE and IPv4 RSVP-TE feature behavior.
    - In the legacy mode of operation, all TE attributes are carried in the legacy IS-IS TE TLVs and the corresponding BGP-LS link attributes TLVs, as listed in Table 12: Legacy link TE TLV support in TE-DB and BGP-LS.
    - In the legacy with application indication mode of operation, IGP and BGP-LS advertises the legacy TE attribute TLVs, and also advertises the ASLA TLV with the Legacy (L) flag set and the

RSVP-TE and SR-TE application flags set. No TE sub-sub-TLVs are advertised within the ASLA TLV.

The legacy with application indication mode is intended for cases where link TE attributes are common to RSVP-TE and SR-TE applications and have the same value, but the user wants to indicate on a per-link basis which application is enabled.

- In the application-specific mode of operation, the TE attribute TLVs are sent as sub-sub-TLVs within the ASLA TLV. Common attributes to RSVP-TE and SR-TE applications have the main TLV L-flag cleared and the RSVP-TE and SR-TE application flags set. Any attribute that is specific to an application (RSVP-TE or SR-TE) is advertised in a separate ASLA TLV with the main TLV L-flag cleared and the specific application (RSVP-TE or SR-TE) flags set.

  The application-specific mode of operation is intended for future cases where TE attributes may have different values in RSVP-TE and SR-TE applications or are specific to one application (for example, RSVP-TE uses the Unreserved Bandwidth and Max Reservable Bandwidth attributes).

- Exporting of any IPv6 and TE link TLVs that have been received from a BGP peer from the local BGP process to the local Enhanced TE-DB via a BGP-LS route family that is enabled on an IPv4 or IPv6 transport BGP session
- Exporting of IPv6 and TE link TLVs from local Enhanced TE-DB to NSP via the CPROTO channel on the VSR-NRC

2.2.19.4.1 BGP-LS originator node handling of TE attributes

The specification of the BGP-LS originator node in support of the ASLA TLV addresses the following main objectives.

1. Accommodate IGP node advertising the TE attribute in both legacy or application specific modes of operation.
2. Allow BGP-LS consumers (for example, PCE) that support the ASLA TLV to receive per-application attributes, even if the attribute values are duplicate, and easily store them per-application in the TE-DB. Also, if BGP-LS consumers receive legacy attributes, they can make a determination without ambiguity that these attributes are only for the RSVP-TE LSP application.
3. Provide continued support for older BGP-LS consumers that rely only on the legacy attributes.

The preceding objectives are supported by enhancements implemented in SR OS on the BGP-LS originator node. The following excerpts adapted from *draft-ietf-idr-bgp-ls-app-specific-attr* describe the enhancements:

1. Application-specific link attributes received from an IGP node without the use of ASLA encodings continue to be encoded using the respective BGP-LS top-level TLVs.
2. Application-specific link attributes received from an OSPF node using ASLA sub-TLV or from an IS-IS node using either ASLA sub-TLV or Application-Specific SRLG TLV must be encoded in the BGP-LS ASLA TLV as sub-TLVs. Exceptions to this rule are specified in 3.f and 3.g.
3. In the case of IS-IS, the following specific procedures are to be followed:
   a. When application-specific link attributes are received from a node with the L-flag set in the IS-IS ASLA sub-TLV and application bits other than RSVP-TE are set in the application bit masks, the application-specific link attributes advertised in the corresponding legacy IS-IS TLVs/sub-TLVs must be encoded within the BGP-LS ASLA TLV as sub-TLVs with the application bits, other than the RSVP-TE bit, copied from the IS-IS ASLA sub-TLV. The link attributes advertised in the legacy IS-IS TLVs/sub-TLVs are also advertised in BGP-LS top-

level TLVs as per [RFC7752] [RFC8571] [RFC9104]. The same procedure also applies for the advertisement of the SRLG values from the IS-IS Application-Specific SRLG TLV.

b. When the IS-IS ASLA sub-TLV has the RSVP-TE application bit set, the link attributes for the corresponding IS-IS ASLA sub-TLVs must be encoded using the respective BGP-LS top-level TLVs as per [RFC7752] [RFC8571] [RFC9104]. Similarly, when the IS-IS Application-Specific SRLG TLV has the RSVP-TE application bit set, the SRLG values within it must be encoded using the top-level BGP-LS SRLG TLV (1096) as per [RFC7752].

c. The SRLGs advertised in IS-IS Application-Specific SRLG TLVs and the other link attributes advertised in IS-IS ASLA sub-TLVs are required to be collated, on a per-application basis, only for those applications that meet all of the following criteria:

- Their bit is set in the SABM/UDABM in one of the two types of IS-IS encodings (for example, IS-IS ASLA sub-TLV).
- The other encoding type (for example, IS-IS Application Specific SRLG TLV) has an advertisement with zero-length application bit masks.
- There is no corresponding advertisement of that other encoding type (following the example, IS-IS Application Specific SRLG TLV) with that specific application bit set.

For each such application, its collated information must be carried in a BGP-LS ASLA TLV with that application's bit set in the SABM/UDABM.

d. If the resulting set of collated link attributes and SRLG values is common across multiple applications, they may be advertised in a common BGP-LS ASLA TLV instance, where the bits for all such applications would be set in the application bit mask.

e. Both the SRLG values from IS-IS Application-Specific SRLG TLVs and the link attributes from IS-IS ASLA sub-TLVs, with the zero-length application bit mask, must be advertised into a BGP-LS ASLA TLV with a zero-length application bit mask, independent of the collation described in 3.c and 3.d.

f. [RFC8919] allows the advertisement of the Maximum Link Bandwidth within an IS-IS ASLA sub-TLV, even though it is not an application-specific attribute. However, when originating the Maximum Link Bandwidth into BGP-LS, the attribute must be encoded only in the top-level BGP-LS Maximum Link Bandwidth TLV (1089) and must not be advertised within the BGP-LS ASLA TLV.

g. [RFC8919] also allows the advertisement of the Maximum Reservable Link Bandwidth and the Unreserved Bandwidth within an IS-IS ASLA sub-TLV, even though these attributes are specific to RSVP-TE application. However, when originating the Maximum Reservable Link Bandwidth and Unreserved Bandwidth into BGP-LS, these attributes must be encoded only in the BGP-LS top-level Maximum Reservable Link Bandwidth TLV (1090) and Unreserved Bandwidth TLV (1091) respectively and not within the BGP-LS ASLA TLV.

2.2.19.4.2 TE attributes supported in IGP and BGP-LS

Table 12: Legacy link TE TLV support in TE-DB and BGP-LS lists the TE attributes that are advertised using the legacy link TE TLVs defined in RFC 5305 for IS-IS and in RFC 3630 for OSPF. These TE attributes are carried in BGP-LS in accordance with RFC 7752. These legacy TLVs are already supported in SR OS and in IS-IS, OSPF, and BGP-LS.

To support IPv6 TE, the IS-IS IPv6 TE attributes (IPv6 TE router ID and IPv6 SRLG TLV) are advertised in BGP-LS in accordance with RFC 7752. These attributes can now be advertised within the ASLA TLV in IS-IS as recommended in RFC 8919 and in BGP-LS as recommended in *draft-ietf-idr-bgp-ls-app-specific-attr*.

In the latter case, BGP-LS uses the same TLV type defined in RFC 7752 but is included as a sub-TLV of the new BGP-LS ASLA TLV. The following table also lists the code points for IS-IS and BGP-LS TLVs.

Table 12: Legacy link TE TLV support in TE-DB and BGP-LS

Link TE TLV description	IS-IS TLV type (RFC 5305)	OSPF TLV type (RFC 3630)	BGP-LS link NLRI link-attribute TLV type (RFC 7752)
Administrative group (color)	3	9	1088
Maximum link bandwidth	9	6	1089
Maximum reservable link bandwidth	10	7	1090
Unreserved bandwidth	11	8	1091
TE Default Metric	18	5	1092
SRLG	138 (RFC 4205)	16 (RFC 4203)	1096
IPv6 SRLG TLV	139 (RFC 6119)	—	1096
IPv6 TE Router ID	140 (RFC 6119)	—	1029
Application Specific Link Attributes	16 (RFC 8919)	—	1122 (provisional, as per *draft-ietf-idr-bgp-ls-app-specific-attr*)
Application Specific SRLG TLV	238 (RFC 8919)	—	1122 (provisional, as per *draft-ietf-idr-bgp-ls-app-specific-attr)*

The following table lists the TE attributes that are received from a third-party router implementation in legacy TE TLVs, or in the ASLA TLV for the RSVP-TE or SR-TE applications that are added into the local SR OSTE-DB; these are also distributed by the BGP-LS originator. However, these TLVs are not originated by an SR OS router IGP implementation.

Table 13: Additional link TE TLV support in TE-DB and BGP-LS

Link TE TLV description	IS-IS TLV type (RFC 7810)	OSPF TLV type (RFC 7471)	BGP-LS link NLRI link-attribute TLV type (RFC 8571)
Unidirectional Link Delay	33	27	1114
Min/Max Unidirectional Link Delay	34	28	1115
Unidirectional Delay Variation	35	29	1116
Unidirectional Link Loss	36	30	1117
Unidirectional Residual Bandwidth	37	31	1118

Link TE TLV description	IS-IS TLV type (RFC 7810)	OSPF TLV type (RFC 7471)	BGP-LS link NLRI link-attribute TLV type (RFC 8571)
Unidirectional Available Bandwidth	38	32	1119
Unidirectional Utilized Bandwidth	39	33	1120

Any other TE attribute received in a legacy TE TLV or in an Application Specific Link Attributes TLV is not added to the local router TE-DB and, therefore, is not distributed by the BGP-LS originator.

2.2.19.5 IS-IS IPv4/IPv6 SR-TE and IPv4 RSVP-TE feature behavior

The TE feature in IS-IS allows the advertising router to indicate to other routers in the TE domain which applications the advertising router has enabled: RSVP-TE, SR-TE, or both. As a result, a receiving router can safely prune links that are not enabled in one of the applications from the topology when computing a CSPF path in that application.

TE behavior consists of the following steps.

1. A valid IPv6 address value must exist for the system or loopback interface assigned to the **ipv6-te-router-id** command. The IPv6 address value can be either the preferred primary global unicast address of the system interface (default value) or that of a loopback interface (user-configured value).

   The IPv6 TE router ID is mandatory for enabling IPv6 TE and enabling the router to be uniquely identified by other routers in an IGP TE domain as being IPv6 TE capable. If a valid value does not exist, then the IPv6 and TE TLVs described in IS-IS, BGP-LS, and TE database extensions are not advertised.

2. The **traffic-engineering** command enables the existing traffic engineering behavior for IPv4 RSVP-TE and IPv4 SR-TE. Enable the **rsvp** context on the router and enable **rsvp** on the interfaces to have IS-IS begin advertising TE attributes in the legacy TLVs. By default, the **rsvp** context is enabled as soon as the **mpls** context is enabled on the interface. If the **ipv6** command is also enabled, then the RFC 6119 IPv6 and TE link TLVs described above are advertised such that a router receiving these advertisements can compute paths for IPv6 SR-TE LSP in addition to paths for IPv4 RSVP-TE LSP and IPv4 SR-TE LSP. The receiving node cannot determine if IPv4 RSVP-TE, IPv4 SR-TE, or IPv6 SR-TE applications are enabled on the other routers. Legacy TE routers must assume that RSVP-TE is enabled on those remote TE links it received advertisements for.

3. When the **ipv6** command is enabled, IS-IS automatically begins advertising the RFC 6119 TLVs and sub-TLVs: the IPv6 TE router ID TLV, the IPv6 interface Address sub-TLV and Neighbor Address sub-TLV, or the Link-Local Interface Identifiers sub-TLV if the interface has no global unicast IPv6 address. The TLVs and sub-TLVs are advertised regardless of whether TE attributes are added to the interface in the **mpls** context. The advertisement of these TLVs is only performed when the **ipv6** command is enabled and **ipv6-routing** is enabled in this IS-IS instance and **ipv6-te-router-id** has a valid IPv6 address.

   A network IP interface is advertised with the Link-Local Interface Identifiers sub-TLV if the network IP interface meets the following conditions:

   - network IP interface has a link-local IPv6 address and no global unicast IPv6 address on the interface **ipv6** context
   - network IP interface has no IPv4 address (and may or may not have the **unnumbered** option enabled on the interface **ipv4** context)

4. The **application-link-attributes** command enables the ability to send the link TE attributes on a per-application basis and explicitly conveys that RSVP-TE or SR-TE is enabled on that link on the advertising router.

   Three modes of operation are allowed by the **application-link-attributes** command.

   - Legacy mode
   - Legacy mode with application indication
   - Application-specific mode

   The following table summarizes the IS-IS link TE parameter advertisement details for the three modes of operation of the IS-IS advertisement.

Table 14: Details of link TE advertisement methods

IGP traffic engineering options		Link TE advertisement details		
		RSVP-TE (rsvp enabled on interface)	**SR-TE (segment-routing enabled in IGP instance)**	**RSVP-TE and SR-TE (rsvp enabled on interface and segment-routing enabled in IGP instance)**
Legacy mode: Use the **no application-link-attributes** command.		Legacy TE TLVs	—	Legacy TE TLVs
Legacy mode with application indication: Enable **configure router isis traffic-engineering-options application-link-attributes legacy**	**rsvp** disabled on router (**rsvp** operationally down on all interfaces)	—	Legacy TE TLVs ASLA TLV -Flags: {Legacy=0, SR-TE=1}; TE sub-sub-TLVs	Legacy TE TLVs ASLA TLV -Flags: {Legacy=1, RSVP-TE=0, SR-TE=1}
	rsvp enabled on router	Legacy TE TLVs ASLA TLV -Flags: {Legacy=1, RSVP-TE=1}	Legacy TE TLVs ASLA TLV -Flags: {Legacy=1, SR-TE=1}	Legacy TE TLVs ASLA TLV -Flags: {Legacy=1, RSVP-TE=1, SR-TE=1}
Application specific mode: Disable **configure router isis traffic-engineering-options application-link-attributes legacy**		ASLA TLV -Flags: {Legacy=0, RSVP-TE=1}; TE sub-sub-TLVs	ASLA TLV -Flags: {Legacy=0, SR-TE=1}; TE sub-sub-TLVs	ASLA TLV -Flags: {Legacy=0, RSVP-TE=1; SR-TE=1}; TE sub-sub-TLVs (common attributes) ASLA TLV -Flags: {Legacy=0, RSVP-TE=1}; TE sub-sub-TLVs (RSVP-TE specific attributes; for example, Unreserved BW and Resvble BW)

IGP traffic engineering options	Link TE advertisement details		
	RSVP-TE (rsvp enabled on interface)	**SR-TE (segment-routing enabled in IGP instance)**	**RSVP-TE and SR-TE (rsvp enabled on interface and segment-routing enabled in IGP instance)**
			ASLA TLV -Flags: {Legacy=0, SR-TE=1}; TE sub-sub-TLVs (SR-TE specific attributes; not supported in SR OS 19.10.R1)

Legacy mode

For legacy mode, use the **no application-link-attributes** command.

The **application-link-attributes** command is disabled by default and the **no** form matches the TE behavior described in list item 2. It enables the existing TE behavior for IPv4 RSVP-TE and IPv4 SR-TE. Only the RSVP-TE attributes are advertised in the legacy TE TLVs that are used by both RSVP-TE and SR-TE LSP CSPF in the TE domain routers. No separate SR-TE attributes are advertised.

If the **ipv6** command is also enabled, then the *RFC 6119* IPv6 and TE link TLVs are advertised in the legacy TLVs. A router in the TE domain receiving these advertisements can compute paths for IPv6 SR-TE LSP.

If the user shuts down the **rsvp** context on the router or on a specific interface, the legacy TE attributes of all the MPLS interfaces or of that specific MPLS interface are not advertised. Routers can still compute SR-TE LSPs using those links, but LSP path TE constraints are not enforced because the links appear in the TE Database as if they did not have TE parameters.

Table 12: Legacy link TE TLV support in TE-DB and BGP-LS shows the encoding of the legacy TE TLVs in both IS-IS and BGP-LS.

Legacy mode with application indication

To use legacy mode with application indication, enable the **legacy** command in the **configure router isis traffic-engineering-options application-link-attributes** context.

The legacy with application indication mode is intended for cases where link TE attributes are common to RSVP-TE and SR-TE applications and have the same value, but the user wants to indicate on a per-link basis which application is enabled.

IS-IS continues to advertise the legacy TE attributes for both RSVP-TE and SR-TE applications and includes the Application-Specific Link Attributes TLV with the application flag set to RSVP-TE, SR-TE, or both, but without the sub-sub-TLVs. IS-IS also advertises the Application-Specific SRLG TLV with the application flag set to RSVP-TE, SR-TE, or both, but without the actual values of the SRLGs.

Routers in the TE domain use these attributes to compute CSPF for IPv4 RSVP-TE LSP and IPv4 SR-TE LSP.

If the **ipv6** command is also enabled, the *RFC 6119* IPv6 and TE TLVs are advertised. A router in the TE domain that receives these advertisements can compute paths for IPv6 SR-TE LSP.

Note:
The **segment-routing** command must be enabled in the IS-IS instance; otherwise, the flag for the SR-TE application cannot be set in the Application-Specific Link Attributes TLV or Application-Specific SRLG TLV.

To disable advertising of RSVP-TE attributes, shut down the **rsvp** context on the router. However, doing so reverts to advertising the link SR-TE attributes using the Application-Specific Link Attributes TLV and the TE sub-sub-TLVs as shown in Table 14: Details of link TE advertisement methods. If legacy attributes were used, legacy routers incorrectly interpret that this router enabled RSVP and may signal RSVP-TE LSP paths using its links.

Table 12: Legacy link TE TLV support in TE-DB and BGP-LS lists the code points for IS-IS and BGP-LS legacy TLVs.

Example

The following excerpt from the Link State Database (LSDB) shows the advertisement of TE parameters for a link with both RSVP-TE and SR-TE applications enabled.

```
TE IS Nbrs   :
   Nbr   : Dut-A.00
   Default Metric  : 10
   Sub TLV Len     : 124
   IF Addr   : 10.10.2.3
   IPv6 Addr : 3ffe::10:10:2:3
   Nbr IP    : 10.10.2.1
   Nbr IPv6  : 3ffe::10:10:2:1
   MaxLink BW: 100000 kbps
   Resvble BW: 500000 kbps
   Unresvd BW:
       BW[0] : 500000 kbps
       BW[1] : 500000 kbps
       BW[2] : 500000 kbps
       BW[3] : 500000 kbps
       BW[4] : 500000 kbps
       BW[5] : 500000 kbps
       BW[6] : 500000 kbps
       BW[7] : 500000 kbps
   Admin Grp : 0x1
   TE Metric : 123
   TE APP LINK ATTR    :
        SABML-flags:Legacy SABM-flags:RSVP-TE SR-TE
   Adj-SID: Flags:v4VL Weight:0 Label:524287
   Adj-SID: Flags:v6BVL Weight:0 Label:524284
   TE SRLGs     :
      SRLGs : Dut-A.00
      Lcl Addr  : 10.10.2.3
      Rem Addr  : 10.10.2.1
      Num SRLGs       : 1
             1003
   TE APP SRLGs     :
       Nbr : Dut-A.00
       SABML-flags:Legacy SABM-flags: SR-TE
       IF Addr   : 10.10.2.3
       Nbr IP    : 10.10.2.1
```

Application-specific mode

To use legacy mode with application indication, disable the **legacy** command in the **configure router isis traffic-engineering-options application-link-attributes** context.

The application-specific mode of operation is intended for future use cases where TE attributes may have different values in RSVP-TE and SR-TE applications (this capability is not supported in SR OS) or are specific to one application (for example, RSVP-TE uses the Unreserved Bandwidth and Max Reservable Bandwidth attributes).

IS-IS advertises the TE attributes that are common to RSVP-TE and SR-TE applications in the sub-sub-TLVs of the new ASLA sub-TLV. IS-IS also advertises the link SRLG values in the Application-Specific SRLG TLV. In both cases, the application flags for RSVP-TE and SR-TE are also set in the sub-TLV.

IS-IS advertises the TE attributes that are specific to the RSVP-TE application separately in the sub-sub-TLVs of the new application attribute sub-TLV. The application flag for RSVP-TE is also set in the sub-TLV.

SR OS does not support configuring and advertising TE attributes that are specific to the SR-TE application.

Common value RSVP-TE and SR-TE TE attributes are combined in the same application attribute sub-TLV with both application flags set, while the non-common value TE attributes are sent in their own application attribute sub-TLV with the corresponding application flag set.

Figure 27: Attribute mapping per application shows an excerpt from the Link State Database (LSDB). Attributes in green font are common to both RSVP-TE and SR-TE applications and are combined, while the attribute in red font is specific to the RSVP-TE application and is sent separately.

Figure 27: Attribute mapping per application

```
TE IS Nbrs   :
     Nbr   : Dut-A.00
     Default Metric  : 100
     Sub TLV Len     : 111
     IF Addr   : 1.0.13.3
     IPv6 Addr : 3ffe::102:606
     Nbr IP    : 1.0.13.1
     Adj-SID: Flags:v4BVL Weight:0 Label:524285
     Adj-SID: Flags:v6BVL Weight:0 Label:524284
     SABML-flags:Non-Legacy SABM-flags:RSVP-TE SR-TE
         MaxLink BW: 99999997 kbps
         Admin Grp : 0x0
         TE Metric : 100
     SABML-flags:Non-Legacy SABM-flags:RSVP-TE
         Resvble BW: 99999997 kbps
         Unresvd BW:
             BW[0] : 99999997 kbps
             BW[1] : 99999997 kbps
             BW[2] : 99999997 kbps
             BW[3] : 99999997 kbps
             BW[4] : 99999997 kbps
             BW[5] : 99999997 kbps
             BW[6] : 99999997 kbps
             BW[7] : 99999997 kbps
TE APP SRLGs    :
    Nbr : Dut-A.00
    SABML-flags:Non-Legacy SABM-flags:RSVP-TE SR-TE
    IF Addr   : 1.0.13.3
    Nbr IP    : 1.0.13.1
    Num SRLGs : 1
    SRLGs     : 1
```

sw0973

Routers in the TE domain use these attributes to compute CSPF for IPv4 RSVP-TE LSPs and IPv4 SR-TE LSPs. If the **ipv6** command is also enabled, the *RFC 6119* IPv6 TLVs are advertised. A router in the TE domain receiving these advertisements can compute paths for IPv6 SR-TE LSP.

Note:
The **segment-routing** command must be enabled in the IS-IS instance or the common TE attribute is not advertised for the SR-TE application.

To disable advertising of RSVP-TE attributes, use the **rsvp shutdown** command on the router.

2.2.19.6 IPv6 SR-TE LSP support in MPLS

This feature is supported with the hop-to-label, the local CSPF, and the PCE (PCC-initiated and PCE-initiated) path computation methods.

All capabilities of an IPv4 provisioned SR-TE LSP are supported with an IPv6 SR-TE LSP unless indicated otherwise. This section describes some important differences between an IPv4 and IPv6 SR-TE LSP support in MPLS.

The IPv6 address used in the **from** and **to** commands in the IPv6 SR-TE LSP, as well as the address used in the **hop** command of the path used with the IPv6 SR-TE LSP, must correspond to the preferred primary global unicast IPv6 address of a network interface or a loopback interface of the corresponding LER or LSR router. The IPv6 address can also be set to the system interface IPv6 address. Failure to follow the preceding IPv6 address guidelines for the **from**, **to**, and **hop** commands causes path computation to fail with failure code "noCspfRouteToDestination". A link-local IPv6 address of a network interface is also not allowed in the **hop** command of the path used with the IPv6 SR-TE LSP. The configuration fails.

A TE link with no global unicast IPv6 address and only a link local IPv6 address can be used in the path computation by the local CSPF. The address shown in the Computed Hops and in the Actual Hops fields of the output of the path **show** command uses the neighbor's IPv6 TE router ID and the Link-Local Interface Identifiers sub-TLV. The exceptions are if the interface is of type broadcast or type point-to-point but also has a local IPv4 address. Only the neighbor's IPv6 TE router ID is shown, as the Link-Local Interface Identifiers sub-TLV is not advertised in these situations.

The UP value of the global MPLS IPv4 state requires that the system interface be in the admin UP state and to have a valid IPv4 address.

The UP value of the global MPLS IPv6 state requires that the interface used for the IPv6 TE router ID be in admin UP state and to have a valid preferred primary IPv6 global unicast address.

The UP value of the TE interface MPLS IPv4 state requires the interface be in the admin UP state in the **router** context and the global MPLS IPv4 state be in UP state.

The UP value of the TE interface MPLS IPv6 state requires the interface be in the admin UP state in the **router** context and the global MPLS IPv6 state be in UP state.

2.2.19.6.1 IPv6 SR-TE auto-LSP

This feature supports the auto-creation of an IPv6 SR-TE mesh LSP and for an IPv6 SR-TE one-hop LSP.

The SR-TE mesh LSP feature specifically binds an LSP template of type **mesh-p2p-srte** with one or more IPv6 prefix lists. When the TE-DB discovers a router that has an IPv6 TE router ID matching an entry in the prefix list, it triggers MPLS to instantiate an SR-TE LSP to that router using the LSP parameters in the LSP template.

The SR-TE one-hop LSP feature specifically activates an LSP template of type **one-hop-p2p-srte**. In this case, the TE database keeps track of each TE link that comes up to a directly connected IGP TE neighbor. It then instructs MPLS to instantiate an SR-TE LSP with the following parameters:

- the source IPv6 address of the local router
- an outgoing interface matching the interface index of the TE-link
- a destination address matching the IPv6 TE router ID of the neighbor on the TE link

A **family** CLI leaf is added to the LSP template configuration and must be set to the **ipv6** value. By default, this command is set to the **ipv4** value for backward compatibility. When establishing both IPv4 and IPv6 SR-TE mesh auto-LSPs with the same parameters and constraints, a separate LSP template of type **mesh-p2p-srte** must be configured for each address family with the **family** command set to the IPv4 or IPv6 value. SR-TE one-hop auto-LSPs can only be established for either IPv4 or IPv6 family, not both. The **family** command in the LSP template of type **one-hop-p2p-srte** should be set to the needed IP family value.

Note:
An IPv6 SR-TE auto-LSP can be reported to a PCE but cannot be delegated or have its paths computed by the PCE.

All capabilities of an IPv4 SR-TE auto-LSP are supported with an IPv6 SR-TE auto-LSP unless indicated otherwise.

2.2.20 OSPF link TE attribute reuse

This section describes the support of OSPF application-specific TE link attributes.

2.2.20.1 OSPF application-specific TE link attributes

Existing definitions for the advertisement of OSPFv2 TE-related link attributes (for example, bandwidth) are used in RSVP-TE deployments (see *draft-ietf-spring-segment-routing-policy-07.txt* for more information). Initially, these TE-related link attributes were only used by RSVP-TE. However, additional applications emerged that also required link attributes (for example, SR-TE). The link attributes used by these new applications are not always identical to those advertised in RSVP-TE.

The usage of link attributes has introduced ambiguity in deployments that include a mix of RSVP-TE and SR-TE support. For example, it is not possible to unambiguously indicate the specific advertisements used by RSVP-TE and SR-TE. Although this may not be an issue for fully congruent topologies, any incongruence causes ambiguity. An additional issue arises in cases where both applications are supported on a link but the link attribute values associated with each application differ. Advertisements without OSPFv2 application-specific TE link attributes do not support the advertisement of application specific values for the same attribute on a specific link.

CLI syntax:

```
Config
   router
      ospf
         traffic-engineering-options
               sr-te {legacy | application-specific-link-attributes}
               no sr-te
```

The **traffic-engineering-options** command enables the context to configure advertisement of the TE attributes of each link on a per-application basis. Two applications are supported in SR OS: RSVP-TE and SR-TE.

The **legacy** mode of advertising TE attributes that is used in RSVP-TE is still supported. In addition, the following configuration options are allowed:

- **no sr-te**

  This option advertises the TE information for RSVP links using TE opaque LSAs. The **no** form is the default value.

- **sr-te legacy**

  This option advertises the TE information for MPLS-enabled SR links using TE opaque LSAs.

  Note: The operator should not use the **sr-te legacy** option if the network has both RSVP-TE and SR-TE applied and the links are not congruent.

- **sr-te application-specific-link-attributes**

  This option advertises the TE information for MPLS-enabled SR links using the new Application Specific Link Attributes (ASLA) TLVs.

See RFC 8920 for definitions of a subset of all possible TE extensions and TE Metric Extensions that can be encoded within Application Specific Link sub-TLVs. The following table describes the relevant values for SR OS.

Table 15: Nokia support for ASLA extended link TLV encoding

OSPFv2 extended link TLV sub-TLVs (RFC 7684)				
IANA	Attribute type	TE-DB[15]	SR OS sub-TLV of Extended Link TLV[16]	SR OS Nested sub-TLV of ASLA Extended Link TLV encoding[17]
10	ASLA	✓	✓	—
11	Shared Risk Link Group	✓	—	✓
12	Unidirectional Link Delay	✓	—	—
13	Min/Max Unidirectional Link Delay	✓	—	—
14	Unidirectional Delay Variation	✓	—	—
15	Unidirectional Link Loss	✓	—	—

15 Support to include the attributes from received LSAs into the Nokia TE-DB and export into BGP-LS. See *draft-ietf-idr-bgp-ls-app-specific-attr* for more information.

16 Node support to encode the link attribute as a sub-TLV in an OSPFv2 Extended Link TLV.

17 Node support to encode the link attribute as a sub-TLV in an OSPFv2 Application Specific Extended Link sub-TLV.

OSPFv2 extended link TLV sub-TLVs (RFC 7684)				
16	Unidirectional Residual Bandwidth	✓	—	—
17	Unidirectional Available Bandwidth	✓	—	—
18	Unidirectional Utilized Bandwidth	✓	—	—
19	Administrative Group	✓	—	✓
20	Extended Administrative Group	✓	—	—
22	TE Metric	✓	—	✓
23	Maximum Link Bandwidth	✓	✓	—

The solution proposed in *draft-ietf-ospf-te-link-attr-reuse-14* assumes that OSPF does not need to move all RSVP-TE attributes from the TE Opaque LSA into the extended link LSA. For, RSVP-TE, consequently, there is no significant modification and it can continue to be advertised using existing OSPF TLVs. For SR-TE and future applications, the ASLA TLVs may be used. Alternatively, existing TE Opaque LSAs could be used through configuration. The following table describes the possible configurations for TE Opaque LSAs.

Table 16: Configuration considerations for TE Opaque LSAs

IGP configuration	ospf>traffic-engineering <20.7	ospf>traffic-engineering ospf>te-opts>no sr-te	ospf>traffic-engineering ospf>te-opts>sr-te legacy	ospf>traffic-engineering ospf>te-opts>sr-te application-link-attribute
Interface configuration	—	—	—	—
MPLS + RSVP	TE-Opaque	TE-Opaque	TE-Opaque	TE-Opaque
MPLS + SR	—	—	TE-Opaque[18]	ASLA (SR-TE)
MPLS + RSVP + SR	TE-Opaque	TE-Opaque	TE-Opaque	TE-Opaque (RSVP) + ASLA (SR-TE)

2.2.21 Configuring and operating SR-TE

This section provides information about the configuration and operation of the SR-TE LSP.

[18] If the local router interface is configured with MPLS and SR, and RSVP-TE is deployed on remote servers, the remote routers incorrectly conclude that the link is RSVP-enabled.

2.2.21.1 SR-TE configuration prerequisites

About this task

To configure SR-TE, the user must first configure prerequisite parameters.

Procedure

Step 1. Configure the label space partition for the Segment Routing Global Block (SRGB) for all participating routers in the segment routing domain by using the **mpls-labels**>**sr-labels** command.

Example

```
mpls-labels
        – sr-labels start 200000 end 200400
    – exit
```

Step 2. Enable segment routing, traffic engineering, and advertisement of router capability in all participating IGP instances in all participating routers by using the **traffic-engineering**, **advertise-router-capability**, and **segment-routing** commands.

Example

```
ospf 0
        – traffic-engineering
        – advertise-router-capability area
        – loopfree-alternates remote-lfa
        – area 0.0.0.202
            – stub
                – no summaries
            – exit
            – interface "system"
                – node-sid index 194
                – no shutdown
            – exit
            – interface "toSim199"
                – interface-type point-to-point
                – no shutdown
            – exit
            – interface "toSim213"
                – interface-type point-to-point
                – no shutdown
            – exit
            – interface "toSim219"
                – interface-type point-to-point
                – metric 2000
                – no shutdown
            – exit
        – exit
        – segment-routing
            – prefix-sid-range global
            – no shutdown
        – exit
        – no shutdown
    – exit
```

Step 3. Configure an segment routing tunnel MTU for the IGP instance, if required, by using the **tunnel-mtu** command.

Example

```
prefix-sid-range global
    – tunnel-mtu 1500
    – no shutdown
```

Step 4. Assign a node SID to each loopback interface that a router would use as the destination of a segment routing tunnel by using the **node-sid** command.

Example

```
ospf 0
        – area 0.0.0.202
            – interface "system"
                – node-sid index 194
                – no shutdown
            – exit
```

2.2.21.2 SR-TE LSP configuration overview

The use can configure an SR-TE LSP as a label switched path (LSP) under the MPLS context by specifying the **sr-te** LSP type.

```
config>router>mpls>lsp lsp-name [mpls-tp src-tunnel-num | sr-te]
```

The user can configure a primary path for an RSVP LSP.

Use the following CLI syntax to associate an empty path or a path with strict or loose explicit hops with the primary paths of the SR-TE LSP:

```
config>router>mpls>path>hop hop-index ip-address {strict | loose}
    – config>router>mpls>lsp>primary path-name
```

2.2.21.3 Configuring path computation and control for SR-TE LSPs

Use the following syntax to configure the path computation requests only (PCE-computed) or both path computation requests and path updates (PCE-controlled) to the PCE for a specific LSP:

```
config>router>mpls>lsp>path-computation-method pce
    – config>router>mpls>lsp>pce-control
```

Use the following command syntax to ensure the PCC LSP database is synchronized with the PCE LSP database using the PCEP PCRpt (PCE Report) message for LSPs that have the following commands enabled:

```
config>router>mpls>pce-report sr-te {enable | disable}
    – config>router>mpls>lsp>pce-report {enable | disable | inherit}
```

2.2.21.3.1 Configuring the path profile and group for PCC-initiated and PCE-computed and PCE-controlled LSP

The PCE supports the computation of disjoint paths for two different LSPs originating or terminating on the same or different PE routers. To indicate this constraint to the PCE, the user must configure the PCE path profile ID and path group ID that the LSP belongs to. These parameters are passed transparently by PCC to PCE and are opaque data to the router. Use the following CLI command syntax to configure the path profile and path group.

```
config>router>mpls>lsp>path-profile profile-id [path-group group-id]
```

The association of the optional path group ID allows PCE to determine the profile ID to use with this path group ID. One path group ID is allowed per profile ID. However, the user can enter the same path group ID with multiple profile IDs by executing this command multiple times. A maximum of five entries of **path-profile** [*path-group*] can be associated with the same LSP. See Path Computation Element Protocol (PCEP) for information about the operation of the PCE path profile.

2.2.21.4 Configuring SR-TE LSP label stack size

Use the following CLI command syntax to configure the maximum number of labels that the ingress LER can push for a specific SR-TE LSP.

```
config>router>mpls>lsp>max-sr-labels label-stack-size
```

This command allows the user to reduce the SR-TE LSP label stack size by accounting for additional transport, service, and other labels when packets are forwarded in a particular context. See Datapath support for more information about label stack size requirements in various forwarding contexts. If the CSPF on the PCE or the hop-to-label translation of the router cannot find a path that meets the maximum SR label stack, the SR-TE LSP remains on its current path or remains down if it has no path. The range is 1-10 labels with a default value of 6.

2.2.21.5 Configuring adjacency SID parameters

Configure the adjacency hold timer for the LFA or remote LFA backup next hop of an adjacency SID.

Use the following CLI command syntax to configure the length of the interval during which LTN or ILM records of an adjacency SID are kept:

```
config>router>ospf>segment-routing>adj-sid-hold seconds[1..300, default 15]
    – config>router>isis>segment-routing>adj-sid-hold seconds[1..300, default 15]
```

```
adj-sid-hold 15
    – no entropy-label-capability
    – prefix-sid-range global
    – no tunnel-table-pref
    – no tunnel-mtu
    – no backup-node-sid
    – no shutdown
```

When protection is enabled globally for all node SIDs and local adjacency SIDs with the **loopfree-alternates** command in IS-IS or OSPF at the LER and LSR, applications may exist for which the user wants traffic to never divert from the strict hop computed by CSPF for an SR-TE LSP. In such cases, use the following CLI command syntax to disable protection for all adjacency SIDs formed over a network IP interface:

```
config>router>ospf>area>if>no sid-protection
    – config>router>isis>if>no sid-protection
```

Example: Configuration output

```
node-sid index 194
    – no sid-protection
    – no shutdown
```

2.2.21.6 Configuring PCC-controlled, PCE-computed, and PCE-controlled SR-TE LSPs

The following examples are configuration outputs.

Output example: PCEP PCC parameters on LER routers that require peering with the PCE server

```
keepalive 30
    – dead-timer 120
    – no local-address
    – unknown-message-rate 10
    – report-path-constraints
    – peer 192.168.48.226
        – no shutdown
    – exit
    – no shutdown
```

Output example: PCC-controlled SR-TE LSP not reported to the PCE

```
lsp "to-SanFrancisco" sr-te
        – to 192.168.48.211
        – path-computation-method local-cspf
        – pce-report disable
        – metric 10
        – primary "loose-anycast"
        – exit
        – no shutdown
    – exit
```

Output example: PCC-controlled SR-TE LSP reported to the PCE

```
lsp "to-SanFrancisco" sr-te
        – to 192.168.48.211
        – path-computation-method local-cspf
        – pce-report enable
        – metric 10
        – primary "loose-anycast"
        – exit
        – no shutdown
    – exit
```

Output example: PCE-computed SR-TE LSP reported to the PCE

```
lsp "to-SanFrancisco" sr-te
        – to 192.168.48.211
        – path-computation-method local-cspf
        – pce-report enable
        – metric 10
        – primary "loose-anycast"
        – exit
        – no shutdown
    – exit
```

Output example: PCE-controlled SR-TE LSP with no PCE path profile

```
lsp "from Reno to Atlanta no Profile" sr-te
        – to 192.168.48.224
        – path-computation-method local-cspf
        – pce-report enable
        – pce-control
        – primary "empty"
        – exit
        – no shutdown
    – exit
```

Output example: PCE-controlled SR-TE LSP with a PCE path profile and a maximum label stack set to a non-default value

```
lsp "from Reno to Atlanta no Profile" sr-te
        – to 192.168.48.224
        – max-sr-labels 8 additional-frr-labels 1
        – path-computation-method pce
        – pce-report enable
        – pce-control
        – path-profile 10 path-group 2
        – primary "empty"
            – bandwidth 15
        – exit
        – no shutdown
    – exit
```

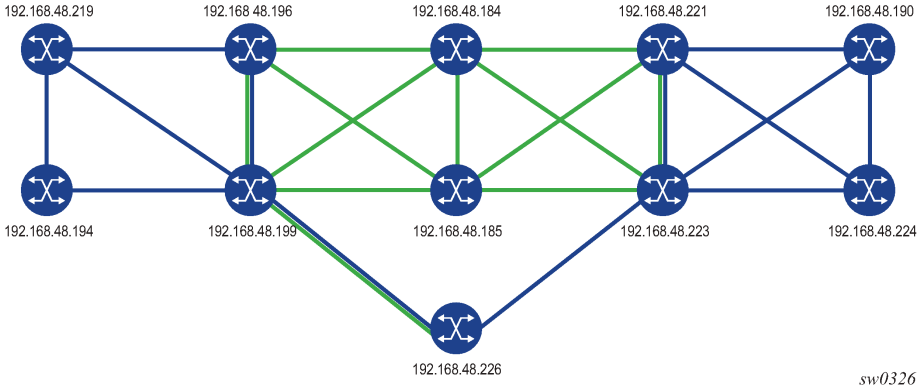
2.2.21.7 Configuring a mesh of SR-TE auto-LSPs

The following shows the detailed configuration for the creation of a mesh of SR-TE auto-LSPs. The network uses IS-IS with the backbone area being in Level 2 and the leaf areas being in Level 1.

The NSP is used for network discovery only and the NRC-P learns the network topology using BGP-LS.

The following figure shows the view of the multilevel IS-IS topology in the NSP GUI. The backbone Level 2 area is highlighted in green.

Figure 28: Multilevel IS-IS topology in the NSP GUI



The mesh of SR-TE auto-LSPs is created in the backbone area and originates on an ABR node with address 192.168.48.199 (Phoenix 199). The LSP template uses a default path that includes an anycast SID prefix corresponding to transit routers 192.168.48.184 (Dallas 184) and 192.168.48.185 (Houston 185).

Output example

The following is the configuration of transit router Dallas 184, which shows the creation of a loopback interface with the anycast prefix and the assignment of a SID to it. The same configuration must be performed on the transit router Houston 185. See lines marked with an asterisk (*).

```
*A:Dallas 184>config>router# info
----------------------------------------------
echo "IP Configuration"
#----------------------------------------------
        if-attribute
            admin-group "olive" value 20
            admin-group "top" value 10
            srlg-group "top" value 10
        exit
        interface "anycast-sid"                                                 *
            address 192.168.48.99/32                                            *
            loopback                                                            *
            no shutdown                                                         *
        exit
        interface "system"
            address 192.168.48.184/32
            no shutdown
        exit
        interface "toJun164"
            address 10.19.2.184/24
            port 1/1/4:10
            no shutdown
        exit
        interface "toSim185"
            address 10.0.3.184/24
            port 1/1/2
            no shutdown
        exit
        interface "toSim198"
            address 10.0.2.184/24
            port 1/1/3
            if-attribute
                admin-group "olive"
```

```
            exit
            no shutdown
        exit
        interface "toSim199"
            address 10.0.13.184/24
            port 1/1/5
            no shutdown
        exit
        interface "toSim221"
            address 10.0.4.184/24
            port 1/1/1
            no shutdown
        exit
        interface "toSim223"
            address 10.0.14.184/24
            port 1/1/6
            no shutdown
        exit
#--------------------------------------------------

*A:Dallas 184>config>router>isis# info
----------------------------------------------
            level-capability level-2
            area-id 49.0000
            database-export identifier 10 bgp-ls-identifier 10
            traffic-engineering
            advertise-router-capability area
            level 2
                wide-metrics-only
            exit
            interface "system"
                ipv4-node-sid index 384
                no shutdown
            exit
            interface "toSim198"
                interface-type point-to-point
                no shutdown
            exit
            interface "toSim185"
                interface-type point-to-point
                no shutdown
            exit
            interface "toSim221"
                interface-type point-to-point
                no shutdown
            exit
            interface "toSim199"
                interface-type point-to-point
                level 2
                    metric 100
                exit
                no shutdown
            exit
            interface "toSim223"
                interface-type point-to-point
                level 2
                    metric 100
                exit
                no shutdown
            exit
            interface "anycast-sid"                                          *
                ipv4-node-sid index 99                                       *
                no shutdown                                                  *
            exit
```

```
            segment-routing
                prefix-sid-range global
                no shutdown
            exit
            no shutdown
-----------------------------------------------
```

In the ingress LER Phoenix 199 router, the anycast SID is learned from both transit routers, but is currently resolved in IS-IS to transit router Houston 185. See lines marked with an asterisk (*).

```
*A:Phoenix 199# show router isis prefix-sids
===============================================================================
Rtr Base ISIS Instance 0 Prefix/SID Table
===============================================================================
Prefix                          SID        Lvl/Typ    SRMS   AdvRtr
                                                       MT     Flags
-------------------------------------------------------------------------------
192.168.48.194/32                399        1/Int.      N     Reno 194
                                                         0       NnP
192.168.48.194/32                399        2/Int.      N     Salt Lake 198
                                                         0       RNnP
192.168.48.194/32                399        2/Int.      N     Phoenix 199
                                                         0       RNnP
192.168.48.99/32                 99         2/Int.      N     Dallas 184         *
                                                         0       NnP             *
192.168.48.99/32                 99         2/Int.      N     Houston 185        *
                                                         0       NnP             *
192.168.48.184/32                384        2/Int.      N     Dallas 184
                                                         0       NnP
192.168.48.185/32                385        2/Int.      N     Houston 185
                                                         0       NnP
192.168.48.190/32                390        2/Int.      N     Chicago 221
                                                         0       RNnP
192.168.48.190/32                390        2/Int.      N     St Louis 223
                                                         0       RNnP
192.168.48.194/32                394        1/Int.      N     Reno 194
                                                         0       NnP
192.168.48.194/32                394        2/Int.      N     Salt Lake 198
                                                         0       RNnP
192.168.48.194/32                394        2/Int.      N     Phoenix 199
                                                         0       RNnP
192.168.48.198/32                398        1/Int.      N     Salt Lake 198
                                                         0       NnP
192.168.48.198/32                398        2/Int.      N     Salt Lake 198
                                                         0       NnP
192.168.48.198/32                398        2/Int.      N     Phoenix 199
                                                         0       RNnP
192.168.48.199/32                399        2/Int.      N     Salt Lake 198
                                                         0       RNnP
192.168.48.199/32                399        1/Int.      N     Phoenix 199
                                                         0       NnP
192.168.48.199/32                399        2/Int.      N     Phoenix 199
                                                         0       NnP
192.168.48.219/32                319        2/Int.      N     Salt Lake 198
                                                         0       RNnP
192.168.48.219/32                319        2/Int.      N     Phoenix 199
                                                         0       RNnP
192.168.48.219/32                319        1/Int.      N     Las Vegas 219
                                                         0       NnP
192.168.48.221/32                321        2/Int.      N     Chicago 221
                                                         0       NnP
192.168.48.221/32                321        2/Int.      N     St Louis 223
                                                         0       RNnP
```

```
192.168.48.223/32                   323        2/Int.      N     Chicago 221
                                                             0       RNnP
192.168.48.223/32                   323        2/Int.      N     St Louis 223
                                                             0       NnP
192.168.48.224/32                   324        2/Int.      N     Chicago 221
                                                             0       RNnP
192.168.48.224/32                   324        2/Int.      N     St Louis 223
                                                             0       RNnP
192.168.48.226/32                   326        2/Int.      N     PCE Server 226
                                                             0       NnP
3ffe::a14:194/128                   294        1/Int.      N     Reno 194
                                                             0       NnP
3ffe::a14:194/128                   294        2/Int.      N     Phoenix 199
                                                             0       RNnP
3ffe::a14:199/128                   299        1/Int.      N     Phoenix 199
                                                             0       NnP
3ffe::a14:199/128                   299        2/Int.      N     Phoenix 199
                                                             0       NnP
-------------------------------------------------------------------------------
No. of Prefix/SIDs: 32 (15 unique)
-------------------------------------------------------------------------------
SRMS : Y/N  = prefix SID advertised by SR Mapping Server (Y) or not (N)
       S    = SRMS prefix SID is selected to be programmed
Flags: R    = Re-advertisement
       N    = Node-SID
       nP   = no penultimate hop POP
       E    = Explicit-Null
       V    = Prefix-SID carries a value
       L    = value/index has local significance
===============================================================================

*A:Phoenix 199# tools dump router segment-routing tunnel
===========================================================================================
Legend: (B) - Backup Next-hop for Fast Re-Route
        (D) - Duplicate
===========================================================================================
-------------------------------------------------------------------------------------------+
 Prefix                                                                                    |
 Sid-Type        Fwd-Type       In-Label  Prot-Inst                                        |
                 Next Hop(s)                       Out-Label(s) Interface/Tunnel-ID        |
-------------------------------------------------------------------------------------------+
 192.168.48.99                                                                             *
 Node            Orig/Transit   200099    ISIS-0                                         *
                 10.0.5.185                        200099      toSim185                   *
 3ffe::a14:194
 Node            Orig/Transit   200294    ISIS-0
                 fe80::62c2:ffff:fe00:0            200294      toSim194
 3ffe::a14:199
 Node            Terminating    200299    ISIS-0
 192.168.48.219
 Node            Orig/Transit   200319    ISIS-0
                 10.202.5.194                      200319      toSim194
 192.168.48.221
 Node            Orig/Transit   200321    ISIS-0
                 10.0.5.185                        200321      toSim185
 192.168.48.223
 Node            Orig/Transit   200323    ISIS-0
                 10.0.5.185                        200323      toSim185
 192.168.48.224
 Node            Orig/Transit   200324    ISIS-0
                 10.0.5.185                        200324      toSim185
 192.168.48.226
 Node            Orig/Transit   200326    ISIS-0
                 10.0.1.2                          100326      toSim226PCEServer
```

```
 192.168.48.184
 Node            Orig/Transit   200384    ISIS-0
                 10.0.5.185                       200384      toSim185
 192.168.48.185
 Node            Orig/Transit   200385    ISIS-0
                 10.0.5.185                       200385      toSim185
 192.168.48.190
 Node            Orig/Transit   200390    ISIS-0
                 10.0.5.185                       200390      toSim185
 192.168.48.194
 Node            Orig/Transit   200394    ISIS-0
                 10.202.5.194                     200394      toSim194
 192.168.48.198
 Node            Orig/Transit   200398    ISIS-0
                 10.0.9.198                       100398      toSim198
 192.168.48.199
 Node            Terminating    200399    ISIS-0
 10.0.9.198
 Adjacency       Transit        262122    ISIS-0
                 10.0.9.198                       3           toSim198
 10.202.1.219
 Adjacency       Transit        262124    ISIS-0
                 10.202.1.219                     3           toSim219
 10.0.5.185
 Adjacency       Transit        262133    ISIS-0
                 10.0.5.185                       3           toSim185
 fe80::62c2:ffff:fe00:0
 Adjacency       Transit        262134    ISIS-0
                 fe80::62c2:ffff:fe00:0           3           toSim194
 10.0.1.2
 Adjacency       Transit        262137    ISIS-0
                 10.0.1.2                         3           toSim226PCEServer
 10.0.13.184
 Adjacency       Transit        262138    ISIS-0
                 10.0.13.184                      3           toSim184
 10.0.2.2
 Adjacency       Transit        262139    ISIS-0
                 10.0.2.2                         3           toSim226PCEserver202
 10.202.5.194
 Adjacency       Transit        262141    ISIS-0
                 10.202.5.194                     3           toSim194
-------------------------------------------------------------------------------
No. of Entries: 22
-------------------------------------------------------------------------------
```

A policy is configured to add the list of prefixes to which the ingress LER Phoenix 199 must auto-create SR-TE LSPs.

```
*A:Phoenix 199>config>router>policy-options# info
----------------------------------------------
            prefix-list "sr-te-level2"
                prefix 192.168.48.198/32 exact
                prefix 192.168.48.221/32 exact
                prefix 192.168.48.223/32 exact
            exit
            policy-statement "sr-te-auto-lsp"
                entry 10
                    from
                        prefix-list "sr-te-level2"
                    exit
                    action accept
                    exit
                exit
```

```
                default-action drop
                exit
            exit
-----------------------------------------------
```

An LSP template of type **mesh-p2p-srte** is configured, which uses a path with a loose hop corresponding to anycast SID prefix of the transit routers. The LSP template is then bound to the policy containing the prefix list. See lines marked with an asterisk (*).

```
*A:Phoenix 199>config>router>mpls# info
-----------------------------------------------
            cspf-on-loose-hop
            interface "system"
                no shutdown
            exit
            interface "toESS195"
                no shutdown
            exit
            interface "toSim184"
                no shutdown
            exit
            interface "toSim185"
                admin-group "bottom"
                srlg-group "bottom"
                no shutdown
            exit
            interface "toSim194"
                admin-group "bottom"
                srlg-group "bottom"
                no shutdown
            exit
            interface "toSim198"
                no shutdown
            exit
            interface "toSim219"
                no shutdown
            exit
            path "loose-anycast-sid"                                          *
                hop 1 192.168.48.99 loose                                     *
                no shutdown                                                   *
            exit                                                              *
            lsp-template "sr-te-level2-mesh" mesh-p2p-srte                    *
                default-path "loose-anycast-sid"                              *
                max-sr-labels 8 additional-frr-labels 2                       *
                pce-report enable                                             *
                no shutdown                                                   *
            exit                                                              *
            auto-lsp lsp-template "sr-te-level2-mesh" policy "sr-te-auto-lsp" *
            no shutdown                                                       *
-----------------------------------------------
```

One SR-TE LSP is automatically created to each destination matching the prefix in the policy as soon as the router with the router ID matching the address of the prefix appears in the TE database.

The following shows the three SR-TE auto-LSPs created. See lines marked with an asterisk (*).

```
*A:Phoenix 199# show router mpls sr-te-lsp
===============================================================================
MPLS SR-TE LSPs (Originating)
===============================================================================
LSP Name                          To              Tun     Protect   Adm  Opr
                                                  Id      Path
```

```
-------------------------------------------------------------------------------
Phoenix-SL-1                        192.168.48.223    1       N/A       Up   Up
Phoenix-SL-2-Profile                192.168.48.223    2       N/A       Up   Up
Phoenix-SL-3-Profile                192.168.48.223    3       N/A       Up   Up
Phoenix-SL-4-Profile                192.168.48.223    4       N/A       Up   Up
Phoenix-SL-1-Profile                192.168.48.223    5       N/A       Up   Up
Phoenix-SL-2                        192.168.48.223    6       N/A       Up   Up
Phoenix-SL-3                        192.168.48.223    7       N/A       Up   Up
Phoenix-SL-4                        192.168.48.223    8       N/A       Up   Up
sr-te-level2-mesh-192.168.48.198-   192.168.48.198    61442   N/A       Up   Up     *
716803                                                                             *
sr-te-level2-mesh-192.168.48.221-   192.168.48.221    61443   N/A       Up   Up     *
716804                                                                             *
sr-te-level2-mesh-192.168.48.223-   192.168.48.223    61444   N/A       Up   Up     *
716805                                                                             *
-------------------------------------------------------------------------------
LSPs : 17
===============================================================================
```

The auto-generated name uses the syntax convention *TemplateName-DestIpv4Address-TunnelId*, as described in Automatic creation of an SR-TE mesh LSP. The tunnel ID used in the name is the TTM tunnel ID, not the MPLS LSP tunnel ID. See lines marked with an asterisk (*).

```
*A:Phoenix 199# show router mpls sr-te-lsp "sr-te-level2-mesh-192.168.48.223-
716805" detail
===============================================================================
MPLS SR-TE LSPs (Originating) (Detail)
===============================================================================
-------------------------------------------------------------------------------
Type : Originating
-------------------------------------------------------------------------------
LSP Name        : sr-te-level2-mesh-192.168.48.223-716805
LSP Type        : MeshP2PSrTe               LSP Tunnel ID        : 61444          *
LSP Index       : 126979                    TTM Tunnel Id        : 716805         *
From            : 192.168.48.199            To                   : 192.168.48.223
Adm State       : Up                        Oper State           : Up
LSP Up Time     : 0d 00:02:12               LSP Down Time        : 0d 00:00:00
Transitions     : 3                         Path Changes         : 3
Retry Limit     : 0                         Retry Timer          : 30 sec
CSPF            : Enabled
Metric          : N/A                       Use TE metric        : Disabled
Include Grps    :                           Exclude Grps         :
None                                           None
VprnAutoBind    : Enabled
IGP Shortcut    : Enabled                   BGP Shortcut         : Enabled
IGP LFA         : Disabled                  IGP Rel Metric       : Disabled
BGPTransTun     : Enabled
Oper Metric     : 16777215
PCE Report      : Enabled
PCE Compute     : Disabled                  PCE Control          : Disabled
Max SR Labels   : 8                         Additional FRR Labels: 2
Path Profile    :
None
Primary(a)      : loose-anycast-sid         Up Time              : 0d 00:02:12
Bandwidth       : 0 Mbps
===============================================================================
```

The automatically created SR-TE auto-LSPs are also added into the tunnel table to be used by services and shortcut applications. See lines marked with an asterisk (*).

```
*A:Phoenix 199# show router tunnel-table
===============================================================================
```

```
IPv4 Tunnel Table (Router: Base)
===============================================================================
Destination       Owner     Encap TunnelId  Pref     Nexthop        Metric
-------------------------------------------------------------------------------
10.0.5.185/32     isis (0)  MPLS  524370    11       10.0.5.185     0
10.0.9.198/32     isis (0)  MPLS  524368    11       10.0.9.198     0
10.0.13.184/32    isis (0)  MPLS  524340    11       10.0.13.184    0
10.202.1.219/32   isis (0)  MPLS  524333    11       10.202.1.219   0
10.202.5.194/32   isis (0)  MPLS  524355    11       10.202.5.194   0
10.0.1.2/32       isis (0)  MPLS  524364    11       11.0.1.2       0
10.0.2.2/32       isis (0)  MPLS  524363    11       11.0.2.2       0
192.168.48.99/32  isis (0)  MPLS  524294    11       10.0.5.185     10
192.168.48.184/32 ldp       MPLS  65605     9        10.0.5.185     20
192.168.48.184/32 isis (0)  MPLS  524341    11       10.0.5.185     20
192.168.48.185/32 ldp       MPLS  65602     9        10.0.5.185     10
192.168.48.185/32 isis (0)  MPLS  524371    11       10.0.5.185     10
192.168.48.190/32 ldp       MPLS  65606     9        10.0.5.185     40
192.168.48.190/32 isis (0)  MPLS  524362    11       10.0.5.185     40
192.168.48.194/32 ldp       MPLS  65577     9        10.202.5.194   10
192.168.48.194/32 isis (0)  MPLS  524331    11       10.202.5.194   10
192.168.48.198/32 sr-te     MPLS  716803    8        192.168.48.99  16777215      *
192.168.48.198/32 ldp       MPLS  65601     9        10.0.9.198     10
192.168.48.198/32 isis (0)  MPLS  524369    11       10.0.9.198     10
192.168.48.219/32 ldp       MPLS  65579     9        10.202.5.194   20
192.168.48.219/32 isis (0)  MPLS  524334    11       10.202.5.194   20
192.168.48.221/32 sr-te     MPLS  716804    8        192.168.48.99  16777215      *
192.168.48.221/32 ldp       MPLS  65607     9        10.0.5.185     30
192.168.48.221/32 isis (0)  MPLS  524358    11       10.0.5.185     30
192.168.48.223/32 sr-te     MPLS  655362    8        10.0.13.184    200
192.168.48.223/32 sr-te     MPLS  655363    8        10.0.13.184    200
192.168.48.223/32 sr-te     MPLS  655364    8        10.0.5.185     40
192.168.48.223/32 sr-te     MPLS  655365    8        10.0.13.184    120
192.168.48.223/32 sr-te     MPLS  655366    8        10.0.5.185     120
192.168.48.223/32 sr-te     MPLS  655367    8        10.0.13.184    120
192.168.48.223/32 sr-te     MPLS  655368    8        10.0.13.184    200
192.168.48.223/32 sr-te     MPLS  655369    8        10.0.5.185     40
192.168.48.223/32 sr-te     MPLS  716805    8        192.168.48.99  16777215      *
192.168.48.223/32 ldp       MPLS  65603     9        10.0.5.185     20
192.168.48.223/32 isis (0)  MPLS  524306    11       10.0.5.185     20
192.168.48.224/32 ldp       MPLS  65604     9        10.0.5.185     30
192.168.48.224/32 isis (0)  MPLS  524361    11       10.0.5.185     30
192.168.48.226/32 isis (0)  MPLS  524365    11       11.0.1.2       65534
-------------------------------------------------------------------------------
Flags: B = BGP backup route available
       E = inactive best-external BGP route
===============================================================================
```

The details of the path of one of the SR-TE auto-LSPs now show the ERO transiting through the anycast SID of router Houston 185. See lines marked with an asterisk (*).

```
*A:Phoenix 199# show router mpls sr-te-lsp "sr-te-level2-mesh-192.168.48.223-
716805" path detail
===============================================================================
MPLS SR-TE LSP sr-te-level2-mesh-192.168.48.223-716805 Path  (Detail)
===============================================================================
Legend :
    S      - Strict                      L      - Loose
    A-SID  - Adjacency SID               N-SID  - Node SID
    +      - Inherited
===============================================================================
-------------------------------------------------------------------------------
SR-TE LSP sr-te-level2-mesh-192.168.48.223-716805 Path loose-anycast-sid
-------------------------------------------------------------------------------
```

```
LSP Name         : sr-te-level2-mesh-192.168.48.223-716805
Path LSP ID      : 20480
From             : 192.168.48.199       To                   : 192.168.48.223
Admin State      : Up                   Oper State           : Up
Path Name        : loose-anycast-sid    Path Type            : Primary
Path Admin       : Up                   Path Oper            : Up
Path Up Time     : 0d 02:30:28          Path Down Time       : 0d 00:00:00
Retry Limit      : 0                    Retry Timer          : 30 sec
Retry Attempt    : 1                    Next Retry In        : 0 sec
CSPF             : Enabled              Oper CSPF            : Enabled
Bandwidth        : No Reservation       Oper Bandwidth       : 0 Mbps
Hop Limit        : 255                  Oper HopLimit        : 255
Setup Priority   : 7                    Oper Setup Priority  : 7
Hold Priority    : 0                    Oper Hold Priority   : 0
Inter-area       : N/A
PCE Updt ID      : 0                    PCE Updt State       : None
PCE Upd Fail Code: noError
PCE Report       : Enabled              Oper PCE Report      : Disabled
PCE Control      : Disabled             Oper PCE Control     : Disabled
PCE Compute      : Disabled
Include Groups   :                      Oper Include Groups  :
None                                           None
Exclude Groups   :                      Oper Exclude Groups  :
None                                           None
IGP/TE Metric    : 16777215             Oper Metric          : 16777215
Oper MTU         : 1492                 Path Trans           : 1
Failure Code     : noError
Failure Node     : n/a
Explicit Hops    :
    192.168.48.99(L)
Actual Hops      :
    192.168.48.99 (192.168.48.185)(N-SID)          Record Label        : 200099    *
 -> 192.168.48.223 (192.168.48.223)(N-SID)         Record Label        : 200323    *
===============================================================================
```

2.2.22 Entropy label on SR-TE LSPs

The router supports the MPLS entropy label on SR-TE LSPs as described in RFC 6790. LSR nodes in a network can load balance labeled packets more granularly than by hashing on the standard label stack. See the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide* for more information.

To allow the entropy label in the label stack for packets on a SR-TE LSP, the head end router must be able to determine the following:

- Is the far end of the LSP Entropy Label Capability (ELC)?
- Should two additional LSEs (Entropy Label and Entropy Label Indicator - EL/ELI) be added to the SR-TE LSP? This check is required to prevent cases where the additional LSEs may cause the maximum SID depth to be exceeded.

Entropy label can be inserted on packets from a service that is configured to use entropy label if both of these criteria are satisfied.

Announcing the ELC is supported by the OSPF or IS-IS routing protocol. However, processing the ELC signaling is not supported for OSPF or IS-IS segment-routed tunnels. That is, the router does not take into account the entropy label capability received in advertisements from nodes in the IS-IS or OSPF domain when determining if the far end of an SR-TE LSP is capable of receiving and processing packets containing the entropy label. ELC is therefore determined by configuring the **override-tunnel-elc** command, either under the IS-IS or OSPF IGP configuration, or under the SR-TE LSP itself, as described

in the following paragraphs. In addition, use the following command to instruct MPLS that EL/ELI can be inserted on SR-TE LSPs.

```
configure router mpls entropy-label sr-te
```

This command applies to all SR-TE LSPs originating on the router. Note that this global configuration can be overridden on an LSP by LSP basis using the following command.

```
configure router mpls lsp entropy-label
```

If the path of the SR-TE LSP is computed by the local CSPF or IP-to-label translation, the head-end router assumes ELC if either the following commands is configured.

```
configure router isis entropy-label override-tunnel-elc
configure router ospf entropy-label override-tunnel-elc
```

However, this case requires that the far-end node SID of the LSP is advertised within the same domain as the head end. This allows the head-end router to know the association between the far-end IP address and the SID of the node for which to insert the entropy label.

When some types of SR-TE LSP paths are specified as a list of SID labels, the head-end LER cannot derive the ELC of the SR-TE LSP from the IGP. It therefore needs to be explicitly configured for each LSP. This applies to the following cases:

- **for SR-TE LSPs where the primary or secondary path hops consist of static SID labels**

  The SID labels are configured under the following context.

  ```
  configure router mpls path hop
  ```

  In this case, use the following command to configure the ELC.

  ```
  configure router mpls lsp override-tunnel-elc
  ```

- **when the PCE provides the ERO**

  This applies to PCC-initiated SR-TE LSPs with **path-computation-method pce**, PCE-initiated SR-TE LSPs, or on-demand SR-TE auto-LSPs. In these cases, use the following command to configure the ELC for PCC-initiated LSPs.

  ```
  configure router mpls lsp override-tunnel-elc
  ```

  Use the following command for PCE-initiated SR-TE LSPs and on-demand SR-TE auto-LSPs that use the LSP template.

  ```
  configure router mpls lsp-template override-tunnel-elc
  ```

2.3 Segment routing policies

The concept of an SR policy is described in *draft-ietf-spring-segment-routing-policy*. An SR policy specifies a source-routed path from a head-end router to a network endpoint, and the traffic flows that are steered to that source-routed path. An SR policy intended for use by a particular head-end router can be statically configured on that router or advertised to it in the form of a BGP route.

The following terms describe the structure of an SR policy and the relationship between one policy and another.

- **SR policy**

  This is a policy identified by the tuple of (head-end router, endpoint and color). Each SR policy is associated with a set of one or more candidate paths, one of which is selected to implement the SR policy and is installed in the data plane. Certain properties of the SR policy come from the currently selected path, such as binding SID, segment lists, and so on.

- **endpoint**

  This is the far-end router that is the destination of the source-routed path. The endpoint may be null (all-zero IP address) if no specific far-end router is targeted by the policy.

- **color**

  This is a property of an SR policy that determines the sets of traffic flows that are steered by the policy.

- **path**

  This is a set of one or more segment lists that are explicitly or statically configured or dynamically signaled. If a path becomes active then traffic matching the SR policy is load-balanced across the segment lists of the path in an equal, unequal, or weighted distribution. Each path is associated with:

  – a protocol origin (BGP or static)

  – a preference value

  – a binding SID value

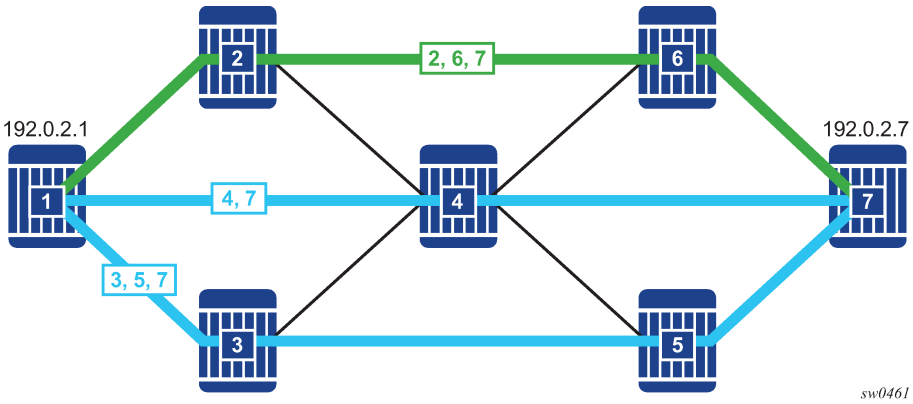
  – a validation state (valid or invalid)

- **binding SID (BSID)**

  This is a SID value that opaquely represents an SR policy (or more specifically, its selected path) to upstream routers. BSIDs provide isolation or decoupling between different source-routed domains and improve overall network scalability. Usually, all candidate paths of an SR policy are assigned the same BSID.

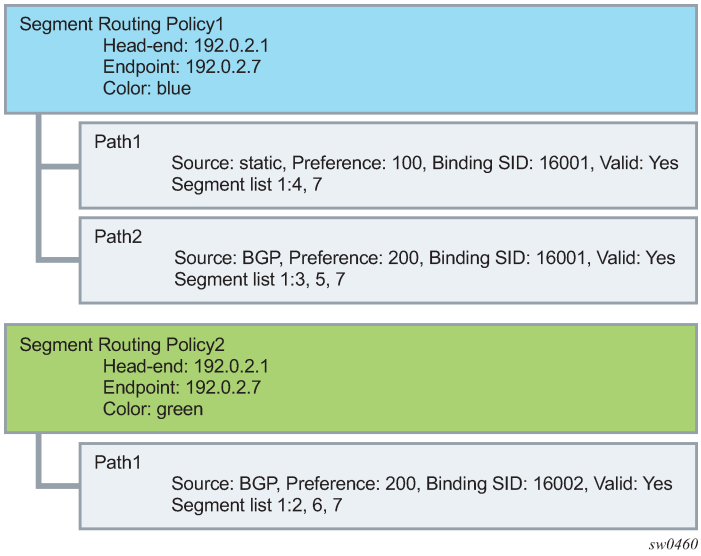
These concepts are illustrated by the following example. Suppose there is a network of seven nodes as shown in the following figure and there are two classes of traffic (blue and green) to be transported between node 1 and node 7. There is an SR policy for the blue traffic between node 1 and node 7 and another SR policy for the green traffic between these same two nodes.

Figure 29: Network example with two SR policies



The two SR policies that are involved in this example and the associated relationships are depicted in Figure 30: Relationship between SR policies and paths.

Figure 30: Relationship between SR policies and paths



2.3.1 Statically-configured segment routing policies

An SR policy is statically configured on the router using one of the supported management interfaces. In the Nokia data model, static policies are configured under **config>router>segment-routing>sr-policies**.

There are two types of static policies: local and non-local. A static policy is local when its **head-end** parameter is configured with the value *local*. This means that the policy is intended for use by the router where the static policy is configured. Local static policies are imported into the local segment routing database for further processing. If the local segment routing database chooses a local static policy as the best path for a specific (color, endpoint) combination, then the associated path and its segment lists are installed into the tunnel table (for next-hop resolution) and as a BSID-indexed MPLS label entry.

A static policy is non-local when its **head-end** parameter is set to any IPv4 address (even an IPv4 address that is associated with the local router, which is a configuration that should generally be avoided). A non-local policy is intended for use by a different router than the one where the policy is configured. Non-local policies are not installed in the local segment routing database and do not affect the forwarding state of the router where they are configured. To advertise non-local policies to the target router, either directly (over a single BGP session) or indirectly (using other intermediate routers, such as BGP route reflectors), the static non-local policies must be imported into the BGP RIB and then re-advertised as BGP routes. To import static non-local policies into BGP, the user must configure the **sr-policy-import** command under **config>router>bgp**. To advertise BGP routes containing SR policies, the user must add the **sr-policy-ipv4** or the **sr-policy-ipv6** family to the configuration of a BGP neighbor or group (or the entire base router BGP instance) so that the capability is negotiated with other routers.

Local and non-local static policies have the same configurable attributes. The function and rules associated with each attribute are:

- **shutdown**

  This command is used to administratively enable or disable the static policy.

- **binding-sid**

  This command is used to associate a binding SID with the static policy in the form of an MPLS label in the range of 32 to 1048575. This is a mandatory parameter. The binding SID must be an available label in the reserved label block associated with SR policies, otherwise the policy cannot be activated.

- **color**

  This command is used to associate a color with the static policy. This is a mandatory parameter.

- **distinguisher**

  This command is used to uniquely identify a non-local static policy when it is a re-advertised as a BGP route. The value is copied into the BGP NLRI field. A unique distinguisher ensures that BGP does not suppress BGP routes for the same (color, endpoint), but is targeted to different head-end routers. This is mandatory for non-local policies but optional in local policies.

- **endpoint**

  This command is used to identify the endpoint IPv4 or IPv6 address associated with the static policy. A value of 0.0.0.0 or 0::0 is permitted and is interpreted as a null endpoint. This is a mandatory parameter.

  Note: When a non-local SR policy with either an IPv4 or IPv6 endpoint is selected for advertisement, the **head-end** command supports an IPv4 address only. This is converted into an IPv4-address-specific RT extended community (0x4102) in the advertised route in the BGP Update message.

- **head-end**

  This command is used to identify the router that is the targeted node for installing the policy. This is a mandatory parameter. The **local** parameter must be used when the target is the local router itself. Otherwise, any valid IPv4 address is allowed, and the policy is considered non-local. When a non-local static policy is re-advertised as a BGP route, the configured head-end address is embedded in an IPv4-address-specific route-target extended community that is automatically added to the BGP route.

- **preference**

  This command is used to indicate the degree of preference of the policy if the local segment routing database has other policies (static or BGP) for the same (color, endpoint) combination. For a path to be selected as the active path for a (color, endpoint) combination, it must have the highest preference value amongst all the candidate paths.

The following are configuration rules related to the previously described attributes.

- Every static local policy must have a unique combination of color, endpoint, and preference.
- Every static non-local policy must have a unique distinguisher.

Each static policy (local and non-local) must include at least one segment list containing at least one segment in its configuration. Each static-policy can have up to 32 segment lists, each containing up to 11 segments. Each segment list can be assigned a weight to influence the share of traffic that it carries compared to other segment lists of the same policy. The default weight is 1.

The segment routing policy draft standard allows a segment list to be configured (and signaled) with a mix of different segment types. When the head-end router attempts to install such a segment routing policy, it must resolve all of the segments into a stack of MPLS labels. In SR OS, this complexity is avoided by requiring that all configured and signaled segments must already be provided in the form of MPLS label values. As described in the draft, this means that only type-1 segments are supported.

2.3.2 BGP-signaled SR policies

The base router BGP instance is configured to send and receive BGP routes containing SR policies. To exchange routes belonging to the (AFI=1, SAFI=73) or (AFI=2, SAFI=73) address family with a specific base router BGP neighbor, the family configuration that applies to that neighbor must include the **sr-policy-ipv4** or the **sr-policy-ipv6** keyword respectively.

When BGP receives an **sr-policy-ipv4** route (AFI=1, SAFI=73) or a **sr-policy-ipv6 route** (AFI=2, SAFI=73) from a peer, it runs its standard BGP best path selection algorithm to choose the best path for each NLRI combination of distinguisher, endpoint, and color. If the best path is targeted to this router as the head end, BGP extracts the SR policy details into the local SR database. A BGP SR policy route is deemed to be targeted to this router as the head end if either:

- it has no route-target extended community and a NO-ADVERTISE standard community
- it has an IPv4 address-specific route-target extended community with an IPv4 address matching the system IPv4 address of this router

An **sr-policy-ipv4** or a **sr-policy-ipv6** route can be received from either an IBGP or EBGP peer but it is never propagated to an EBGP peer. An **sr-policy-ipv4** or a **sr-policy-ipv6** route can be reflected to route reflector clients if this is allowed (a NO_ADVERTISE community is not attached) and the router does not consider itself the head end of the policy.

Note:
A BGP SR policy route is considered malformed if it does not have at least one segment list TLV with at least one segment TLV, which triggers error-handling procedures such as session reset or treat-as withdraw.

2.3.3 Segment routing policy path selection and tie-breaking

Segment Routing policies (static and BGP) for which the local router is the head end are processed by the local segment routing database. For each (color, endpoint) combination, the database validates each candidate path and chooses one to be the active path. The steps of this process are described in the Segment Routing policy validation and selection process.

1. Is the path missing a binding SID in the form of an MPLS label?
   - Yes: this path is invalid and cannot be used.
   - No: go to the next step.
2. Does the path have any segment list that contains a segment type not equal to 1 (that is, an MPLS label)?
   - Yes: this path is invalid and cannot be used.
   - No: go to the next step.

   node-SID
3. Are all segment lists of the path invalid?
   A segment list is invalid if it is empty, if the first SID cannot be resolved to a set of one or more next-hops, or if the weight is 0.
   - Yes: this path is invalid and cannot be used.
   - No: go to the next step.

At this step, the router attempts to resolve the first segment of each segment list to a set of one or more next-hops and outgoing labels. It does so by looking for a matching SID in the segment routing module, which must correspond to one of the following:

- SR IS-IS or SR-OSPF node SID
- SR IS-IS or SR-OSPF adjacency SID
- SR IS-IS or SR-OSPF adjacency set SID (parallel or non-parallel set)

Note:
The label value in the first segment of the segment list is matched against ILM label values that the local router has assigned to the node SIDs, adjacency SIDs, and adjacency set SIDs. The matched ILM entry may not program a swap to the same label value encoded in the segment routing policy; for example, in the case of an adjacency SID or of a node SID reachable through a next hop using a different SRGB base.

4. Is the binding SID an available label in the reserved-label-block range?
   - Yes: go to the next step.
   - No: this path is invalid and cannot be used.
5. Is there another path that has reached this step that has a higher preference value?
   - Yes: this path loses the tie-break and cannot be used.
   - No: go to the next step.
6. Is there a static path?
   - Yes: select the static path as the active path because the protocol-origin value associated with static paths (30) is higher than the protocol-origin value associated with BGP learned paths (20).
   - No: go to the next step.
7. Is there a BGP path with a lower originator value?
   The originator is a 160-bit numerical value formed by the concatenation of a 32-bit ASN and a 128-bit peer address (with IPv4 addresses encoded in the lowest 32 bits).
   - Yes: this path loses the tie-break and cannot be used.
8. Is there another BGP path with a higher distinguisher value?
   - Yes: select the BGP path with the highest distinguisher value.

2.3.4 Resolving BGP routes to segment routing policy tunnels

When a statically configured or BGP-signaled segment routing policy is selected to be the active path for a (color, endpoint) combination, the corresponding path and its segment lists are programmed into the tunnel table of the router. An IPv4 tunnel of type **sr-policy** (where the **endpoint** parameter is an IPv4 address) is programmed into the IPv4 tunnel table (TTMv4). Similarly, an IPv6 tunnel of type **sr-policy** (where the **endpoint** parameter is an IPv6 address) is programmed into the IPv6 tunnel table (TTMv6). The resulting tunnel entries can be used to resolve the following types of BGP routes:

- Unlabeled IPv4 routes
- Unlabeled IPv6 routes
- Label-unicast IPv4 routes

- Label-unicast IPv6 (6PE) routes
- VPN IPv4 and IPv6 routes
- EVPN routes

Specifically, an IPv4 tunnel of type **sr-policy** can be used to resolve:

- an IPv4 or the IPv4-mapped IPv6 next hop of the following route families:

  ipv4, **ipv6**, **vpn-ipv4**, **vpn-ipv6**, **label-ipv4**, **label-ipv6**, **evpn**
- the IPv6 next hop of the following route families:

  ipv6, **label-ipv4**, and **label-ipv6** (SR policy with **endpoint** = 0.0.0.0 only).

An IPv6 tunnel of type **sr-policy** can be used to resolve:

- the IPv6 next hop of the following route families:

  ipv4, **ipv6**, **vpn-ipv4**, **vpn-ipv6**, **label-ipv4**, **label-ipv6**, **evpn**
- the IPv4 next hop of the following route families:

  ipv4 and **label-ipv4** (SR policy with **endpoint** = 0::0 only).
- the IPv4-mapped IPv6 next hop of the following route families:

  label-ipv6 (SR policy with **endpoint** = 0::0 only).

2.3.4.1 Resolving unlabeled IPv4 BGP routes to segment routing policy tunnels

For an unlabeled IPv4 BGP route to be resolved by an SR policy:

- A color-extended community must be attached to the IPv4 route.
- The base instance BGP next-hop-resolution configuration of **shortcut-tunnel>family ipv4** must allow SR policy tunnels.

Note: Contrary to *draft-filsfils-segment-routing-05*, BGP only resolves a route with multiple color-extended communities to an SR policy using the color-extended community with the highest value.

For example, to resolve an IPv4 route with a color-extended community (value *C*) and BGP next-hop address *N* under these conditions, the router performs the following steps:

1. If there is an SR policy in the TTMv4 for the following are true, then use this tunnel to resolve the BGP next hop:
   - end-point = BGP next-hop address
   - color = C*n*
2. If no SR policy is found in the previous step and the C*n* color-extended community has its color-only (CO) bits set to 01 or 10, then search for an SR policy in the TTMv4 for which the following are true:
   - endpoint = null (0.0.0.0)
   - color = C*n*

   If there is such a policy, use it to resolve the BGP next hop.
3. If no SR policy is found in the previous steps and the C*n* color-extended community has its CO bits set to 01 or 10, then search for an SR policy in the TTMv6 for which the following are true.

- endpoint = null (0::0)
- color = C*n*

If there is such a policy, use it to resolve the BGP next hop.

4. If no SR policy is found in the previous steps but there is a non-SR policy tunnel in the TTMv4 that is allowed by the resolution options and for which endpoint = BGP next-hop address (and for which the admin tag meets the admin-tag-policy requirements applied to the BGP route, if applicable), then use this tunnel to resolve the BGP next hop if it has the highest TTM preference.
5. Otherwise, fall back to IGP, unless the **disallow-igp** option is configured.

2.3.4.2 Resolving unlabeled IPv6 BGP routes to segment routing policy tunnels

For an unlabeled IPv6 BGP route to be resolved by an SR policy:

- A color-extended community must be attached to the IPv6 route.
- The base instance BGP next-hop-resolution configuration of **shortcut-tunnel>family ipv6** must allow SR policy tunnels.

Note:

- Contrary to *draft-filsfils-segment-routing-05*, BGP only resolves a route with multiple color-extended communities to an SR policy using the color-extended community with the highest value.
- For AFI2/SAFI1 routes, an IPv6 explicit null label is pushed at the bottom of the stack if the policy endpoint is IPv4.

For example, to resolve an IPv6 route with a color-extended community (value *C*) and BGP next-hop address *N* under these conditions, the router performs the following steps:

1. If there is an SR policy in the TTMv6 for which the following are true, then use this tunnel to resolve the BGP next hop.
   - endpoint = the BGP next-hop address
   - color = C*n*
2. If no SR policy is found in the previous step and the C*n* color-extended community has its CO bits set to 01 or 10, then search for an SR policy in the TTMv6 for which the following are true:
   - endpoint = null (0::0)
   - color = C*n*

   If there is such a policy, use it to resolve the BGP next hop.
3. If no SR policy is found in the previous steps, and the C*n* color-extended community has its CO bits set to 01 or 10 and there is an SR policy in the TTMv4 for which the following are true, then use this tunnel to resolve the BGP next hop.
   - endpoint = null (0.0.0.0)
   - color = C*n*
4. If no SR policy is found in the previous steps but there is a non-SR policy tunnel in the TTMv6 that is allowed by the resolution options and for which endpoint = BGP next-hop address (and for which the

admin-tag meets the admin-tag-policy requirements applied to the BGP route, if applicable), then use this tunnel to resolve the BGP next hop if it has the highest TTM preference.

5. Otherwise, fall back to IGP, unless the **disallow-igp** option is configured.

2.3.4.3 Resolving label-IPv4 BGP routes to segment routing policy tunnels

For a label-unicast IPv4 BGP route to be resolved by an SR policy:

- A color-extended community must be attached to the label-IPv4 route.
- The base instance BGP next-hop-resolution configuration of **labeled-routes>transport-tunnel>family label-ipv4** must allow SR policy tunnels.

Note:
Contrary to *draft-filsfils-segment-routing-05*, BGP only resolves a route with multiple color-extended communities to an SR policy using the color-extended community with the highest value.

For example, to resolve a label-IPv4 route with a color-extended community (value *C*) and BGP next-hop address *N*, the router performs the following steps:

1. If there is an interface route that can resolve the BGP next hop, then use the direct route.
2. If the **allow-static** command is configured and there is a static route that can resolve the BGP next hop, then use the static route.
3. If there is no interface route or static route available or allowed to resolve the BGP next hop and the next hop uses IPv4, then:
   a. Look for an SR policy in the TTMv4 for which the following are true:
      - end-point = BGP next-hop address
      - color = C*n*

      If there is such an SR policy, use it to resolve the BGP next hop. If the selected SR policy has any segment list with more than {11 - **max-sr-frr-labels** under the IGPs} labels or segments, then the label-IPv4 route is unresolved.
   b. If no SR policy is found in the previous steps and the C*n* color-extended community has its CO bits set to 01 or 10, then search for an SR policy in the TTMv4 for which the following are true:
      - endpoint = null (0.0.0.0)
      - color = C*n*

      If there is such a policy, use it to resolve the BGP next hop. If the selected SR policy has any segment list with more than {11- **max-sr-frr-labels** under the IGPs} labels or segments, then the label-IPv4 route is unresolved.
   c. If no SR policy is found in the previous steps and the C*n* color-extended community has its CO bits set to 01 or 10, then search for an SR policy in the TTMv6 for the following are true:
      - endpoint = null (0::0)
      - color = C*n*

      If there is such a policy, use it to resolve the BGP next hop. If the selected SR policy has any segment list with more than {11- **max-sr-frr-labels** under the IGPs} labels or segments, then the label-IPv4 route is unresolved.

4. If there is no interface route or static route that is available or allowed to resolve the BGP next hop and the next hop uses IPv6, then:
   a. Search for an SR policy in the TTMv6 for which the following are true:
      - end-point = BGP next-hop address
      - color = C*n*

      If there is such an SR policy, use it to resolve the BGP next hop. If the selected SR policy has any segment list with more than {11- **max-sr-frr-labels** under the IGPs} labels or segments, then the label-IPv4 route is unresolved.
   b. If no SR policy is found in the previous steps and the C*n* color-extended community has its CO bits set to 01 or 10, then search for an SR policy in the TTMv6 for which the following are true:
      - endpoint = null (0::0)
      - color = C*n*

      If there is such a policy, use it to resolve the BGP next hop. If the selected SR policy has any segment list with more than {11- **max-sr-frr-labels** under the IGPs} labels or segments, then the label-IPv4 route is unresolved.
   c. If no SR policy is found in the previous steps and the C*n* color-extended community has its CO bits set to 01 or 10, then search for an SR policy in the TTMv4 for which the following are true:
      - endpoint = null (0.0.0.0)
      - color = C*n*

      If there is such a policy, use it to resolve the BGP next hop. If the selected SR policy has any segment list with more than {11- **max-sr-frr-labels** under the IGPs} labels or segments, then the label-IPv4 route is unresolved.
5. If no SR policy is found in the previous steps but there is a non-SR policy tunnel in the TTMv4 (the next hop uses IPv4) or TTMv6 (the next hop uses IPv6) that is allowed by the resolution options and for which endpoint = BGP next-hop address (and for which the admin tag meets the admin-tag-policy requirements applied to the BGP route, if applicable), then use this tunnel to resolve the BGP next hop if it has the highest TTM preference.

2.3.4.4 Resolving label-IPv6 BGP routes to segment routing policy tunnels

For a label-unicast IPv6 BGP route to be resolved by an SR policy:

- A color-extended community must be attached to the label-IPv6 route.
- The base instance BGP next-hop-resolution configuration of the **labeled-routes>transport-tunnel>family label-ipv6** command must allow SR policy tunnels.

Note:
Contrary to *draft-filsfils-segment-routing-05*, BGP only resolves a route with multiple color-extended communities to an SR policy using the color-extended community with the highest value.

For example, to resolve a label-IPv6 route with a color-extended community (value *C*) and BGP next-hop address *N*, the router performs the following steps:

1. If there is an interface route that can resolve the BGP next hop, then use the direct route.

2. If the **allow-static** command is configured and there is a static route that can resolve the BGP next hop, then use the static route.
3. If there is no interface route or static route available or allowed to resolve the BGP next hop and the next hop uses IPv6 then:
   a. Look for an SR policy in the TTMv6 for which the following are true:
      - end-point = BGP next-hop address
      - color = C*n*

      If there is such an SR policy, use it to resolve the BGP next hop.
   b. If no SR policy is found in the previous steps and the C*n* color-extended community has its CO bits set to 01 or 10, then search for an SR policy in the TTMv6 for which the following are true.
      - endpoint = null (0::0)
      - color = C*n*

      If there is such a policy, use it to resolve the BGP next hop.
   c. If no SR policy is found in the previous steps and the C*n* color-extended community has its CO bits set to 01 or 10 then search for an SR policy in the TTMv4 for which the following are true:
      - endpoint = null (0.0.0.0)
      - color = C*n*

      If there is such a policy, use it to resolve the BGP next hop.
4. If there is no interface route or static route that is available or allowed to resolve the BGP next hop and the next hop uses IPv4-mapped-IPv6, then:
   a. Look for an SR policy in the TTMv4 for which the following are true:
      - end-point = BGP next-hop address
      - color = C*n*

      If there is such an SR policy then use it to resolve the BGP next hop.
   b. If no SR policy is found in the previous steps and the C*n* color-extended community has its CO bits set to 01 or 10, then search for an SR policy in the TTMv4 for which the following are true:
      - endpoint = null (0.0.0.0)
      - color = C*n*

      If there is such a policy, use it to resolve the BGP next hop.
   c. If no SR policy is found in the previous steps and the C*n* color-extended community has its CO bits set to 01 or 10, then search for an SR policy in the TTMv6 for which the following are true:
      - endpoint = null (0::0)
      - color = C*n*

      If there is such a policy, use it to resolve the BGP next hop.
5. If no SR policy is found in the previous steps but there is a non-SR-policy tunnel in the TTMv6 (the next hop uses IPv6) or in the TTMv4 (the next hop uses IPv4-mapped IPv6) that is allowed by the resolution options and for which endpoint = BGP next-hop address (and for which the admin-tag meets the admin-tag-policy requirements applied to the BGP route, if applicable), then use this tunnel to resolve the BGP next hop if it has the highest TTM preference.

2.3.4.5 Resolving EVPN-MPLS routes to segment routing policy tunnels

The next-hop resolution for all EVPN-VXLAN routes and for EVPN-MPLS routes without a color-extended community is unchanged by this feature.

When the resolution options associated with the **auto-bind-tunnel** configuration of an EVPN-MPLS service (VPLS, B-VPLS, R-VPLS, or Epipe) allow SR policy tunnels from the TTM, the next-hop resolution of EVPN-MPLS routes (RT-1 per-EVI, RT-2, RT-3 and RT-5) with one or more color-extended communities (C1, C2, .. C*n*, where C*n* is the highest value) is based on the following rules.

Note:
Contrary to *draft-filsfils-segment-routing-05*, BGP only resolves a route with multiple color-extended communities to an SR policy using the color-extended community with the highest value.

1. If the next hop uses IPv6 and there is an SR policy in the TTMv6 for which the following are true, then use this tunnel to resolve the BGP next hop.
   - end-point = BGP next-hop address
   - color = C*n*
2. Otherwise, if the next hop uses IPv4 or IPv4-mapped IPv6 and there is an SR policy in the TTMv4 for which the following are true, then use this tunnel to resolve the BGP next hop.
   - end-point = BGP next-hop address (or the IPv4 address extracted from the IPv4-mapped IPv6 BGP next-hop address)
   - color = C*n*
3. If no SR policy is found in the previous steps but there is a non-SR policy tunnel in the TTMv4 (the next hop uses IPv4 or IPv4-mapped IPv6) or TTMv6 (the next hop uses IPv6) that is allowed by the resolution options and for which endpoint = BGP next-hop address, then use this tunnel to resolve the BGP next hop if it has the highest TTM preference.

2.3.4.6 VPRN auto-bind-tunnel using segment routing policy tunnels

When the resolution options associated with the **auto-bind-tunnel** configuration of VPRN service allow SR policy tunnels from the TTM, next-hop resolution of VPN-IPv4 and VPN-IPv6 routes that are imported into the VPRN and have one or more color-extended communities (C1, C2, .. C*n*, where C*n* is the highest value) is based on the following rules.

Note:
Contrary to *draft-filsfils-segment-routing-05*, BGP only resolves a route with multiple color-extended communities to an SR policy using the color-extended community with the highest value.

1. If the next hop uses IPv6 and there is an SR policy in the TTMv6 for which the following are true, then use this tunnel to resolve the BGP next hop.
   - end-point = BGP next-hop address
   - color = C*n*
2. Otherwise, if the next hop uses IPv4 or IPv4-mapped IPv6 and there is an SR policy in the TTMv4 for which the following are true, then use this tunnel to resolve the BGP next hop.

- , end-point = BGP next-hop address (or the IPv4 address extracted from the IPv4-mapped IPv6 BGP next-hop address in the case of VPN-IPv6 routes)
- color = C*n*

3. If no SR policy is found in the previous steps but there is a non-SR policy tunnel in the TTMv4 (the next hop uses IPv4 or IPv4-mapped IPv6) or TTMv6 (the next hop uses IPv6) that is allowed by the resolution options and for which endpoint = BGP next-hop address, then use this tunnel to resolve the BGP next hop if it has the highest TTM preference.

2.3.5 Seamless BFD and end-to-end protection for SR policies

Seamless BFD (S-BFD) is a form of BFD that requires significantly less state and reduces the need for session bootstrapping as compared to LSP BFD. See "Seamless Bidirectional Forwarding Detection (S-BFD)" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide*. S-BFD requires centralized configuration of a reflector function, as well as a mapping at the head-end node between the remote session discriminator and the IP address for the reflector by each session. This configuration and the mapping are described in the *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide*.

This section describes the application of S-BFD to SR polices and the configuration required for this feature. See Seamless BFD for SR-TE LSPs for details of the application of S-BFD to SR-TE LSPs.

By default, S-BFD operates in asynchronous mode where the reflector encapsulates and routes IP/UDP encapsulated S-BFD packets back to the initiator using the IGP shortest path. However, some applications also support a controlled return TE path for S-BFD reply packets, where S-BFD operates in echo mode and the far-end router forwards packets back toward the initiator on a specified labelled path using, for example, an SR policy. This enables a specific TE return path to be configured for each S-BFD session on an SR policy at the initiating node. In this case, the reflector function at the tail end of the SR policy is bypassed.

S-BFD provides a connectivity check for the datapath of a segment list in an SR policy and can determine whether the segment list is up. In addition, the router also supports two protection modes for an SR policy that are distinguished by the datapath programming characteristics and whether uniform failover is required between segment lists in the same SR policy candidate path (ECMP protected mode), or between the programmed candidate paths (linear mode). These protection modes are driven by the S-BFD session state on the programmed segment lists of an SR policy.

2.3.5.1 ECMP protected mode

ECMP protected mode programs all segment lists of the top-two candidate paths of an SR policy in the IOM. ECMP protected mode allows establishment of S-BFD on all of those segment lists. All of the segment lists of a specified candidate path are in the same protection group, but different candidate paths are not in the same protection group. Switchover between candidate paths is triggered by the control plane. A segment list is only included in the ECMP set of segment lists if its S-BFD session is up (user traffic is forwarded on a segment list whose S-BFD session is down). See Figure 31: ECMP protected SR policy with S-BFD.

Figure 31: ECMP protected SR policy with S-BFD

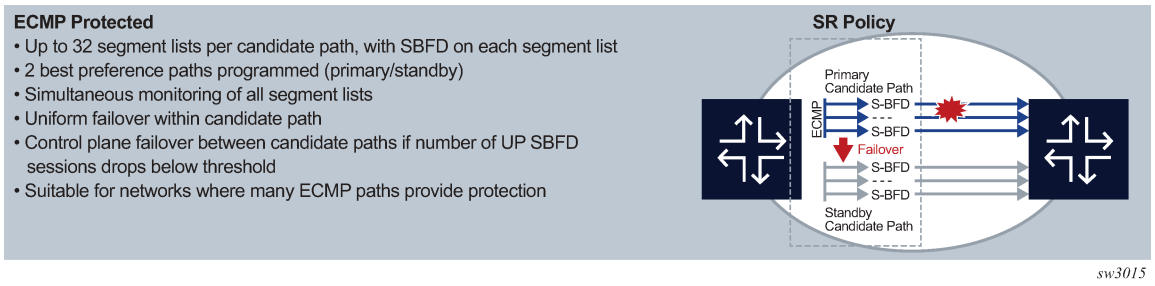
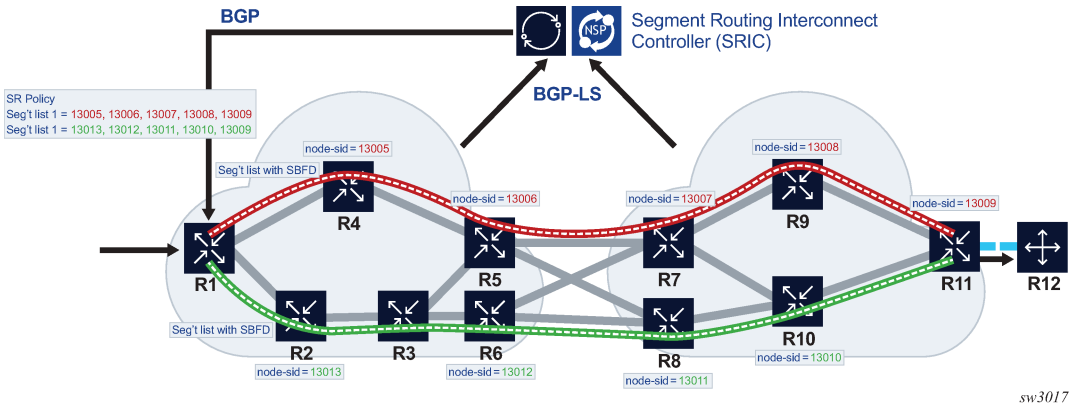


Figure 32: Example application of ECMP protected mode with S-BFD depicts an application for S-BFD on SR policies with ECMP protected mode. Here, an SR policy is programmed at R1 by the NSP with two segment lists from R1 to R11. One segment list is using R4/R5/R7R9, and the other segment list is using R2/R3/R6/R8 and R10. These segment lists are using diverse paths and traffic that is sprayed across both of them according to the configured hashing algorithm. Separate S-BFD sessions are run on each segment list and allow the rapid detection of datapath failures along the whole segment list path. R1 is able to rapidly remove a segment list from the ECMP set if S-BFD goes down, and is also able to failover to a backup SR policy (not shown) (or fall back to a less preferred LSP) if more than a specific number of the S-BFD sessions go down.

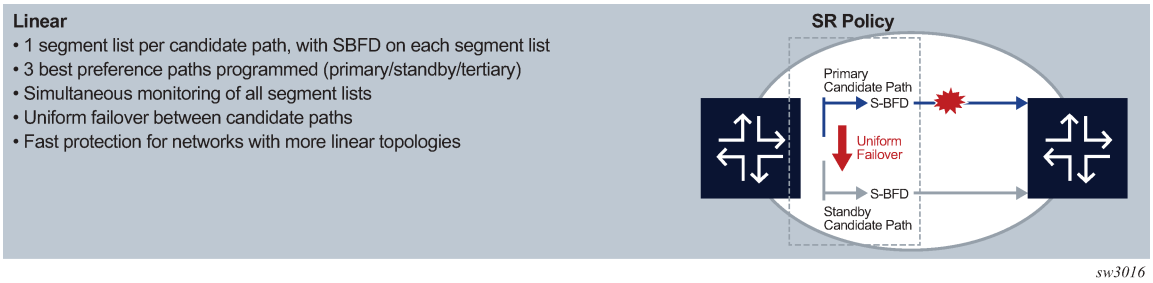
Figure 32: Example application of ECMP protected mode with S-BFD



2.3.5.2 Linear mode

This mode is termed "linear" because it is similar in operation to traditional 1-for-1 linear protection switching. It is intended to allow one or more backup paths to protect a primary path, with fast failover between candidate paths. Uniform failover is supported between candidate paths of the same SR policy. Only one segment list from each of the top-three preference candidate paths is programmed in the IOM. All of the programmed candidate paths of a specified SR policy are in the same protection group. See the following figure.

Figure 33: Linear protected SR policy with S-BFD



2.3.5.3 S-BFD for SR policies detailed description

This section describes the S-BFD for SR policies, support for primary and backup candidate paths, and configuration steps for S-BFD and protection for SR policies.

2.3.5.3.1 S-BFD for SR policies with MPLS or IPv6 data plane

S-BFD is supported on segment lists for both static SR policies and BGP SR policies by binding a maintenance policy containing an S-BFD configuration to an imported SR policy route or a static SR policy. S-BFD packets are encapsulated on the SR policy segment lists in the same way as for SR-TE LSP paths. As in the case of SR-TE LSPs, S-BFD requires the discriminator of the local node as well as a mapping to the far-end reflector node discriminators. BFD sets the remote discriminator at the initiator of the S-BFD session based on a lookup in the S-BFD reflector discriminator using the endpoint address of the SR policy candidate path. A candidate path of an SR policy is only treated as available if the number of up S-BFD sessions equals or exceeds a configured threshold.

Note:
When an SR policy candidate path is first programmed, a 3-second initialization hold timer is triggered. This timer allows the establishment of all the S-BFD sessions for all programmed paths before it decides which candidate path to activate. Eligible candidate paths have a number of segment lists with S-BFD sessions in the up-state that is higher or equal to a configured threshold.

Because this timer is set to 3 seconds, Nokia recommends that the transmit and receive control packet timers are set to no more than 1 second with a maximum multiplier of 3 for S-BFD sessions.

S-BFD control packet timers, which can be configured as low as 10 ms, are supported for specific SR OS platforms with CPM network processor support.

S-BFD can be configured to operate in the following modes:

- **Routed return path**
  In this mode, BFD packets are sent on each segment listed in the SR policy from the session initiator node toward the reflector node. The reflector node then sends the BFD reply packet back to the initiator using a routed return path.
- **Controlled return path**
  For SR policies with an MPLS data plane, this mode is enabled by configuring a return path label for the BFD session. In this mode, the router pushes an additional MPLS label on the S-BFD packets at

the bottom of the stack and the BFD session operates in echo mode. The return path label refers to an MPLS binding SID of a reverse-path SR policy or SR-TE LSP programmed at the far end of the SR policy.

For SR policies with an SRv6 data plane, this mode is enabled by configuring a return path BFD SID for the BFD session. In this mode, the router pushes an additional SRv6 SID in the Segment Routing Header (SRH) on the S-BFD packets as the penultimate SID in the SRH and the BFD session operates in echo mode. The return path BFD SID refers to an SRv6 binding SID of a reverse-path SRv6 policy programmed at the far end of the SR policy.

The return path SR or SRv6 policy can be used to forward S-BFD reply packets along an explicitly traffic-engineered path back to the initiator, avoiding the IGP shortest path. Different SR or SRv6 policies at the initiating end can be configured with different return path labels or return path BFD SIDs, referring to different SR policies, SR-TE LSPs (with Binding SIDs), or SRv6 policies at the far end. These can have segment lists with different paths, ensuring that the BFD reply packets from different head-end SR or SRv6 policy paths do not share the same outcome. S-BFD packets on these sessions bypass the reflector at the far end of the SR or SRv6 policy. Therefore, there is no need to configure a reflector discriminator for these sessions.

2.3.5.3.2 Support for primary and backup candidate paths

End-to-end protection of static and BGP SR policies is supported using ECMP-protected or linear mode.

If an SR policy for a specified (head end, color, endpoint) combination is imported (by BGP) or configured (in the static case) and is selected for use, then the best (highest) preference candidate path is treated as the primary path while the next preference candidate preference policy is treated as the standby path. In linear mode, if a third path is present, it is treated as a tertiary standby path. All of the valid segment lists for these paths are programmed in the IOM and made available for forwarding S-BFD packets, subject to a limitation in linear mode of one segment list per candidate path. In ECMP protected mode, the two best preference candidate paths are programmed in the IOM (up to 32 segment lists per path), while in linear mode, the three best preference candidate paths are programmed in the IOM (one segment list per candidate path).

In each case, segment lists of the best preference path are initially programmed as forwarding NHLEs while the others are programmed as non-forwarding. If the maximum number of programmed paths for a specified mode has been reached (for example, two for ECMP protected mode, and three for linear mode), and a consistent new path is received with a better preference than the existing active path, then this new path is only considered if or when the route for one of the current programmed paths is withdrawn or deleted. However, if the maximum number of programmed paths for the mode has not been reached, then the new path is programmed and any configured revert timer is started. The system switches to that better preference path immediately or when the revert timer expires (as applicable).

Failover is supported between the currently active path and the next best preference path if the currently active path is down because of S-BFD. Similar to the case of SR-TE LSPs, if ECMP protected or linear mode is configured, the system switches back to the primary (best preference) SR policy path as soon as it recovers, by default. This can happen when the number of up S-BFD sessions equals or exceeds a threshold and a hold-down timer has expired. However, it is possible to configure a revert timer to control reversion to the primary path.

All candidate paths of an SR policy must have the same binding SID when one of these two modes is applied.

2.3.5.3.3 Configuration of S-BFD and protection for SR and SRv6 policies

About this task

S-BFD and protection for SR and SRv6 policies is configured using the following steps.

See the *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide* for more information about steps 1 and 2. See the following sections for more information about steps 3 and 4:

- Configuration of SR and SRv6 policy S-BFD and mode command options
- Application of S-BFD and protection command options to static SR policies
- Application of S-BFD and protection command options to BGP SR-policies

Procedure

Step 1. Use the commands in the following context to configure an S-BFD reflector and the mapping for the remote reflector.

```
configure router bfd seamless-bfd
```

Step 2. Use the commands in the following context to configure one or more BFD templates that define the BFD session command options:

- **MD-CLI**

  ```
  configure bfd bfd-template
  ```

- **classic CLI**

  ```
  configure router bfd bfd-template
  ```

Step 3. Use the following command to configure protection and BFD for SR policies using a named maintenance policy.

```
configure router segment-routing maintenance-policy
```

Step 4. Depending on whether a static SR policy or a dynamic BGP SR policy is being configured, perform one of the following:

- For static SR policies, use the following command to apply a named maintenance policy to the static SR policy.

  ```
  configure router segment-routing sr-policies static-policy maintenance-policy
  ```

- For dynamic BGP SR policies, perform the following steps:

  a. Use the following command options to configure a policy statement entry to match on a specific route or set of routes belonging to the segment routing policy IPv4 or IPv6 address family:

  Note: Named entries are only supported in the MD-CLI.

  – **MD-CLI**

  ```
  configure policy-options policy-statement entry from family sr-policy-ipv4
  ```

```
configure policy-options policy-statement entry from family sr-policy-ipv6
configure policy-options policy-statement named-entry from family sr-policy-
ipv4
configure policy-options policy-statement named-entry from family sr-policy-
ipv6
```

– **classic CLI**

```
configure router policy-options policy-statement entry from family sr-policy-
ipv4
configure router policy-options policy-statement entry from family sr-policy-
ipv6
```

b. Use the following command options to configure an accept action or default action for the matched routes belonging to the segment routing policy IPv4 or IPv6 address family:

– **MD-CLI**

```
configure policy-options policy-statement entry action action-type accept
configure policy-options policy-statement named-entry action action-type
 accept
configure policy-options policy-statement default-action action-type accept
```

– **classic CLI**

```
configure router policy-options policy-statement entry action accept
configure router policy-options policy-statement default-action accept
```

c. Apply a named SR maintenance policy to the dynamic BGP SR policy.

2.3.5.3.3.1 Configuration of SR and SRv6 policy S-BFD and mode command options

S-BFD and protection mode command options are configured in a named maintenance policy. This is applied to SR policy paths that are imported by BGP as a policy statement action or by binding to a static SR policy configuration.

Use the commands in the following context to configure maintenance policies.

```
configure router segment-routing maintenance-policy
```

Use the following command to enable or disable BFD on all segment lists of the candidate path that are installed in the datapath:

- **MD-CLI**

```
configure router segment-routing maintenance-policy bfd-liveness
```

- **classic CLI**

```
configure router segment-routing maintenance-policy bfd-enable
```

Note: Seamless BFD is disabled by default.

Use the following command to configure an already-existing named BFD template for seamless BFD to use.

```
configure router segment-routing maintenance-policy bfd-template
```

Use the following command to configure the programming of the datapath and the behavior of the router when the number of BFD sessions up is less than the threshold and the hold-down timer has expired.

```
configure router segment-routing maintenance-policy mode
```

All of the paths in the set must have the same mode (see SR policy route and candidate path consistency). All of the allowed segment lists of the SR policy path are programmed in the IOM. By default, no mode is configured.

In both the **linear** mode and **ecmp-protected** modes, if two or more SR policy paths with the same {headend, color, endpoint} have the same mode, the highest preference path is treated as an effective primary path while the next highest path preference is treated as the standby path. If a third path is present in the **linear** mode, this is treated as a tertiary path and also programmed in the IOM.

In the **ecmp-protected** mode, all the segment lists of the top two best preference paths are programmed in the IOM. However, in **linear** mode, the lowest index segment list of each of the top three preference paths is programmed in the IOM and linear protection is supported between that set. All of the segment lists of the programmed paths are made available for forwarding S-BFD packets.

If the currently active path becomes unavailable because of S-BFD, the system fails over to the next best preference candidate path that is available. If all programmed candidate paths are unavailable, the SR policy is marked as down in TTM.

The **linear** mode supports uniform failover between candidate paths (policy routes) of the same SR policy. If the **linear** mode is configured, the following rules apply:

- Only one segment list is allowed per SR policy path. If more than one is configured, only the lowest index segment list is programmed in the datapath.
- Of the valid paths that belong to the same SR policy, the top three paths with the best preference are programmed in the IOM and are assigned to the same protection group. Uniform failover is supported between these paths.

Use the following command to configure the minimum number of S-BFD sessions that must be up for the SR policy candidate path to be considered up.

```
configure router segment-routing maintenance-policy threshold
```

If the number of S-BFD sessions is below the threshold, the SR policy candidate path is marked as BFD degraded by the system. The **threshold** command can only be used on conjunction with the **ecmp-protected** mode (a threshold of 1 is implicit in the **linear** mode).

Use the following command to enable controlled return path for an SR policy with an MPLS data plane.

```
configure router segment-routing maintenance-policy return-path-label
```

The **return-path-label** command pushes an additional MPLS label at the bottom of the label stack for the S-BFD packet. It also enables echo mode for the S-BFD session. The return-path label refers to a binding SID on an SR policy or other MPLS path configured on the far-end router. The S-BFD packet is returned to the initiator using this MPLS return path, instead of being routed using the IGP path. The **return-path-label** command is ignored if the maintenance policy is applied to an SRv6 policy.

Note: The user can configure a return path label for controlled return path for an SR policy with an MPLS data plane using the following respective commands for an action or default action in the policy statement:

- **MD-CLI**

```
configure policy-options policy-statement entry action sr-return-path-bfd-label
configure policy-options policy-statement named-entry action sr-return-path-bfd-
label
configure policy-options policy-statement default-action sr-return-path-bfd-label
```

- **classic CLI**

```
configure router policy-options policy-statement entry action sr-return-path-bfd-
label
configure router policy-options policy-statement default-action sr-return-path-bfd-
label
```

See the subsequent sections for more information about configuring controlled return path for S-BFD on SRv6 policies.

Use the following command to configure the router to start a revert timer when the primary path recovers (for example, after the number of S-BFD sessions that are up becomes greater than or equal to the threshold and the hold-down timer has expired) that switches back when the timer expires.

```
configure router segment-routing maintenance-policy revert-timer
```

If **revert-timer** is not configured, the system reverts to the primary path for the policy when it is restored.

The revert timer immediately reverts to the primary path, if the following applies:

- A secondary or tertiary path is currently active, and the revert timer is started (because of the recovery of the primary path).
- The secondary path has subsequently gone down because the number of up S-BFD sessions is less than the threshold.
- No better preference standby path is available.

However, if a better preference standby path is available and up, the revert timer is not canceled and the system switches to the better preference standby path, then switches back to the primary path when the revert timer expires. If the hold-down timer is currently active on a better-preference path, the system immediately switches to the primary path. If the system needs to switch to the primary path but the hold-down timer is still active on the primary path, the system cancels the timer and switches immediately.

Use the following command to configure the hold-down timer for the maintenance policy.

```
configure router segment-routing maintenance-policy hold-down-timer
```

The **hold-down-timer** command is intended to prevent bouncing of the SR policy path state if one or more BFD sessions associated with segment lists flap and cause the threshold to be repeatedly crossed in a short period of time. The hold-down timer is started when the number of S-BFD sessions that are up drops below the threshold. The SR policy path is not considered to be up again until the hold-down timer has expired and the number of S-BFD sessions that are up equals or exceeds the threshold.

A maintenance policy can only be deleted or changed if the maintenance policy is administratively disabled. A maintenance policy can only be administratively enabled if S-BFD is enabled, a BFD template

is referenced, and the mode is configured. All associated SR policy paths are deleted from the IOM if a maintenance template is administratively disabled.

2.3.5.3.3.2 Application of S-BFD and protection command options to static SR policies

A named maintenance policy is applied to a static SR policy using the **maintenance-policy** command, as follows:

```
config router segment-routing sr-policies
         static-policy <name>
            head-end local
            binding-sid <number>
            maintenance-policy <name>
            ...
```

A maintenance policy can only be configured if the static SR policy **head-end** is configured with the **local** command option. Policies with an IP address that is not local to the node are not programmed in the SR database and cannot have S-BFD sessions established on them by this node because they are not the head end for the SR policy path.

S-BFD needs an endpoint address for the session so that the S-BFD reflector discriminator can be looked up as a part of the session addressing. A maintenance policy cannot be configured on an SR policy with a null endpoint.

2.3.5.3.3.3 Application of S-BFD and protection command options to BGP SR-policies

S-BFD and protection command options can be applied to matching imported SR policy routes. Match criteria in the route import policy for the color, endpoint, and route distinguisher of a policy enable matching on a specific SR policy route belonging to the segment routing policy IPv4 or IPv6 address family.

Note: For routes with the same matching distinguisher, only those with the best criteria are pushed to the SR database.

Matching a unique SR policy requires a fully qualified set of match criteria. Use the following command options to configure the required match criteria:

Note: The **family** command must be configured to either **sr-policy-ipv4** or **sr-policy-ipv6**.

- **MD-CLI**

```
configure policy-options policy-statement entry from family
configure policy-options policy-statement entry from distinguisher
configure policy-options policy-statement entry from color
configure policy-options policy-statement entry from endpoint
```

- **classic CLI**

```
configure router policy-options policy-statement entry from family
configure router policy-options policy-statement entry from distinguisher
configure router policy-options policy-statement entry from color
configure router policy-options policy-statement entry from endpoint
```

Users may only require general match criteria (for example, to apply the same maintenance template to all imported SR policy IPv4 routes, irrespective of color or endpoint).

Use the following command options to apply an SR maintenance policy to matching SR policy routes as an action on a specific entry:

- **MD-CLI**

```
configure policy-options policy-statement entry action action-type accept
configure policy-options policy-statement named-entry action action-type accept
configure policy-options policy-statement entry action sr-maintenance-policy
configure policy-options policy-statement named-entry action sr-maintenance-policy
```

- **classic CLI**

```
configure router policy-options policy-statement entry action accept
configure router policy-options policy-statement entry action sr-maintenance-policy
```

Use the following command options to apply an SR maintenance policy to matching SR policy routes as the default action:

- **MD-CLI**

```
configure policy-options policy-statement default-action action-type accept
configure policy-options policy-statement default-action sr-maintenance-policy
```

- **classic CLI**

```
configure router policy-options policy-statement default-action accept
configure router policy-options policy-statement default-action sr-maintenance-policy
```

If controlled return path is required for S-BFD on an SR policy with an MPLS data plane, either configure a **return-path-label** in the maintenance policy, or configure **sr-return-path-bfd-label** as the **action** or **default-action** in the policy statement. The **sr-return-path-bfd-label** command can only be configured if a maintenance policy is also configured.

The **sr-return-path-bfd-label** takes precedence over the **return-path-label** value configured in the maintenance policy.

The named SR maintenance policy must exist on the system when the commit is executed for the routing policy. If parameterization of actions is used and the named SR maintenance policy exists, the router still validates.

A change in policy options action deletes all programmed paths for that route and, based on the new action, redownloads applicable routes to the IOM.

2.3.5.3.3.4 SR policy route and candidate path consistency

An SR policy consists of a set of one or more candidate paths. Each candidate path may be described by an SR policy route, which may be a static SR policy that is configured under the **config>router>segment-routing>sr-policies** context or a dynamic route imported by BGP. The router checks the consistency of the following BFD and protection command options across all of the SR policy routes for a specified SR policy.

```
    bfd-enable
    bfd-template <name>
    mode {linear | ecmp-protected}
```

```
    revert-timer <timer-value>
```

{Maintenance-policy existence}

Maintenance-policy existence covers the case where the existing programmed route is an SR policy with no maintenance policy, and the new route has a maintenance policy, and the other way around.

Consistency is enforced across all of the static SR policy candidate paths and dynamic SR policy routes that make up a segment routing policy. Because SR policy routes or paths are imported sequentially and cannot be considered together, inconsistencies are handled as follows:

```
First policy route imported/configured:
Check: valid set of command options
Action: If OK, program in datapath and activate

Second policy route imported/configured:
Check: valid set of command options, consistency with existing activated policy route
Action If OK, program in datapath and activate, else hold in CPM but do not program

Third policy route imported/configured:
Check: valid set of command options, consistency with existing activated policy route (s)
Action If OK, program in datapath and activate, else hold in CPM but do not program
```

Inconsistent policy routes (paths) are only programmed if their command options are valid and any programmed routes for that SR policy are deleted.

By using the same maintenance policy for all of the SR policy's routes, inconsistencies between the BFD and protection command options of SR policy routes belonging to a specified SR policy can be avoided.

2.3.5.3.3.5 Configuration of controlled return path for SRv6 policy S-BFD

For static SRv6 policies, the S-BFD controlled return path is configured using the following context.

```
configure router segment-routing sr-policies static-policy segment-routing-v6 return-path-bfd-
sid
```

The SRv6 policy must also have a maintenance policy configured. When the commands are applied to a static SRv6 policy, the router attempts to establish an echo mode S-BFD session on the segment lists of the SRv6 policy (when programmed) in accordance with the command options of the maintenance policy.

The S-BFD controlled return path for BGP SRv6 policies is enabled by applying both the **maintenance-policy** and **srv6-return-path-bfd-sid** actions in the import policy statement. When both actions are applied to an SRv6 policy, the router attempts to establish an echo mode S-BFD session on the segment lists of the SRv6 policy (when programmed) in accordance with the command options of the maintenance policy. All other commands applicable to SRv6 policies in the maintenance policy apply, such as the protection specification.

2.3.6 Traffic statistics for segment routing policies

SR policies provide the ability to collect statistics for ingress and egress traffic. In both cases, traffic statistics are collected without any forwarding class or QoS distinction.

Traffic statistics collection is enabled as follows:

- **configure router segment-routing sr-policies ingress-statistics**

Ingress traffic collection only applies to **binding-sid** SR policies as the statistic index is attached to the ILM entry for that label. The traffic statistics provide traffic for all the instances that share the binding SID. The statistic index is released and statistics are lost when ingress traffic statistics are disabled for that binding SID or when the last instance of a policy using that label is removed from the database.

- **configure router segment-routing sr-policies egress-statistics**

  Egress traffic statistics are collected globally for all policies at the same time. Both static and signaled policies are subject to traffic statistics collection. Statistic indexes are allocated per segment list, which allows for a fine grain monitoring of traffic evolution, but are only allocated at the time the segment list is effectively programmed. The system allocates up to 32 statistic indexes across all the instances of a policy.

  If an instance of a policy is deprogrammed and a more preferred instance is programmed, the system behaves as follows:

  – If the segment list IDs of the preferred instance are different from any of the segment list IDs of any previously programmed instance, the system allocates new statistic indexes. While that condition holds, the statistics associated with a segment list of an instance strictly reflect the traffic that used that segment list in that instance.

  – If some of the segment list IDs of the preferred instance are equal to any of the segment list IDs of any previously programmed instance, the system reuses the indexes of the preferred instance and keeps the associated counter value and increment. In this case, the traffic statistics provided per segment list not only reflect the traffic that used that segment list in that instance, but incorporates counter values of at least another segment list in another instance of that policy.

In all cases, the aggregate values provided across all instances truly reflect traffic over the various instances of the policy.

Statistic indexes are not released at deprogramming time. They are, however, released when all the instances of a policy are removed from the database or when the **egress-statistics** command is disabled.

When the **egress-statistics** command is enabled, the user can configure rate computation on egress. The traffic rate is determined by an accounting policy configuration that uses the **combined-sr-policy-egress** command option and then references the accounting policy in the following context.

```
configure router segment-routing sr-policies egress-statistics
```

The minimum collection interval is 5 minutes. Rate statistics are determined per segment list and accessible using the **show snmp** command as well as via YANG or NETCONF.

3 Segment routing with IPv6 data plane (SRv6)

3.1 Introduction to SRv6

Segment routing steers packets by encoding the packet-processing instructions for each intermediate and destination router directly in the packet header.

The datapath pushes a list of instructions in the form of Segment Identifiers (SIDs) onto the packet, to forward the payload packet directly to the destination using the shortest path or using source routing via one or more transit routers. Each router that terminates a SID in the segment list performs the instructions related to that SID.

Segment Routing standards specify the following two methods for programming the datapath of the encapsulated packet:

- **Segment Routing MPLS (SR-MPLS)**

  The SID is encoded in 32 bits and programmed as an MPLS label; this method provides a tunnel to both IPv4 and IPv6 destinations. See Segment routing with MPLS data plane (SR-MPLS).

- **Segment Routing IPv6 (SRv6)**
  The SID is encoded in 128 bits and programmed as an IPv6 address; this method provides a tunnel to IPv6 destinations.

SRv6 datapath encapsulation models each SID using a 128-bit address. In shortest path routing, the destination SID is encoded in the Destination Address (DA) field of the outer IPv6 header. In source routing, the SIDs of the nodes the packet must traverse are encoded as a SID list in a Segment Routing Header (SRH), which is a new type of routing header compliant with RFC 8200. The next SID in a segment list to which the packet is to be forwarded is copied from the SRH into the DA field of the outer IPv6 header.

SRv6 provides more than IPv6 transport with shortest path and source-routing capabilities; it provides a framework for programmability of IPv6 networks that takes advantage of the large IPv6 address space.

The following figure shows the SRv6 SRH format and fields (excerpt from RFC 8986).

Figure 34: SRv6 SRH format and fields

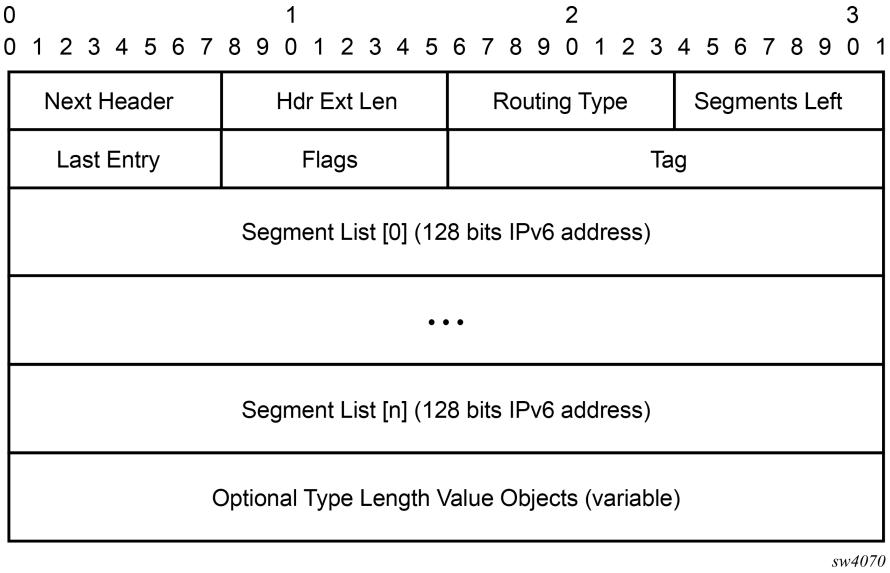


Table 17: SRv6 field descriptions

Field name	Description
Next Header	Defined in RFC 8200, Section 4.4
Hdr Ext Len	Defined in RFC 8200, Section 4.4
Routing Type	4
Segments Left	Defined in RFC 8200, Section 4.4
Last Entry	Contains the index (zero based), in the Segment List, of the last element of the Segment List
Flags	RFC 8754, Section 8.1 creates an IANA registry for new flags to be defined.
Tag	The Tag field is used to mark a packet as part of a class or group of packets; for example, packets sharing the same set of properties. When the Tag field is not used at the source, it must be set to zero on transmission. When the Tag field is not used during SRH processing, it is ignored. The Tag field is not used when processing the SID, as defined in RFC 8754, Section 4.3.1. It may be used when processing other SIDs that are not defined in this document. The allocation and use of tag is outside the scope of this document.
Segment List[0..n]	128-bit IPv6 addresses representing the nth segment in the segment list. The Segment List is encoded starting from the last segment of the SR policy. That is, the first element of the segment list (Segment List[0]) contains the last segment of the SR policy, the second

Field name	Description
	element contains the penultimate segment of the SR policy, and so on.
TLV	Type Length Value (TLV); see RFC 8754, Section 2.1, and Figure 36: SRv6 SID encoding.

The following flags are defined for the SRv6 SRH Flags field.

Figure 35: 8-bits of flags

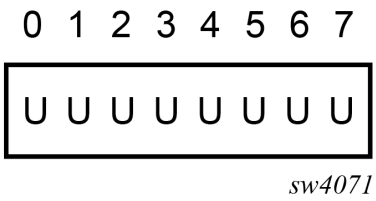


Table 18: Flag descriptions

Flag	Description
U	Unused and for future use. Must be 0 on transmission and ignored on receipt.

The following figure shows the SRv6 Segment Identifier (SID) encoding format and fields.

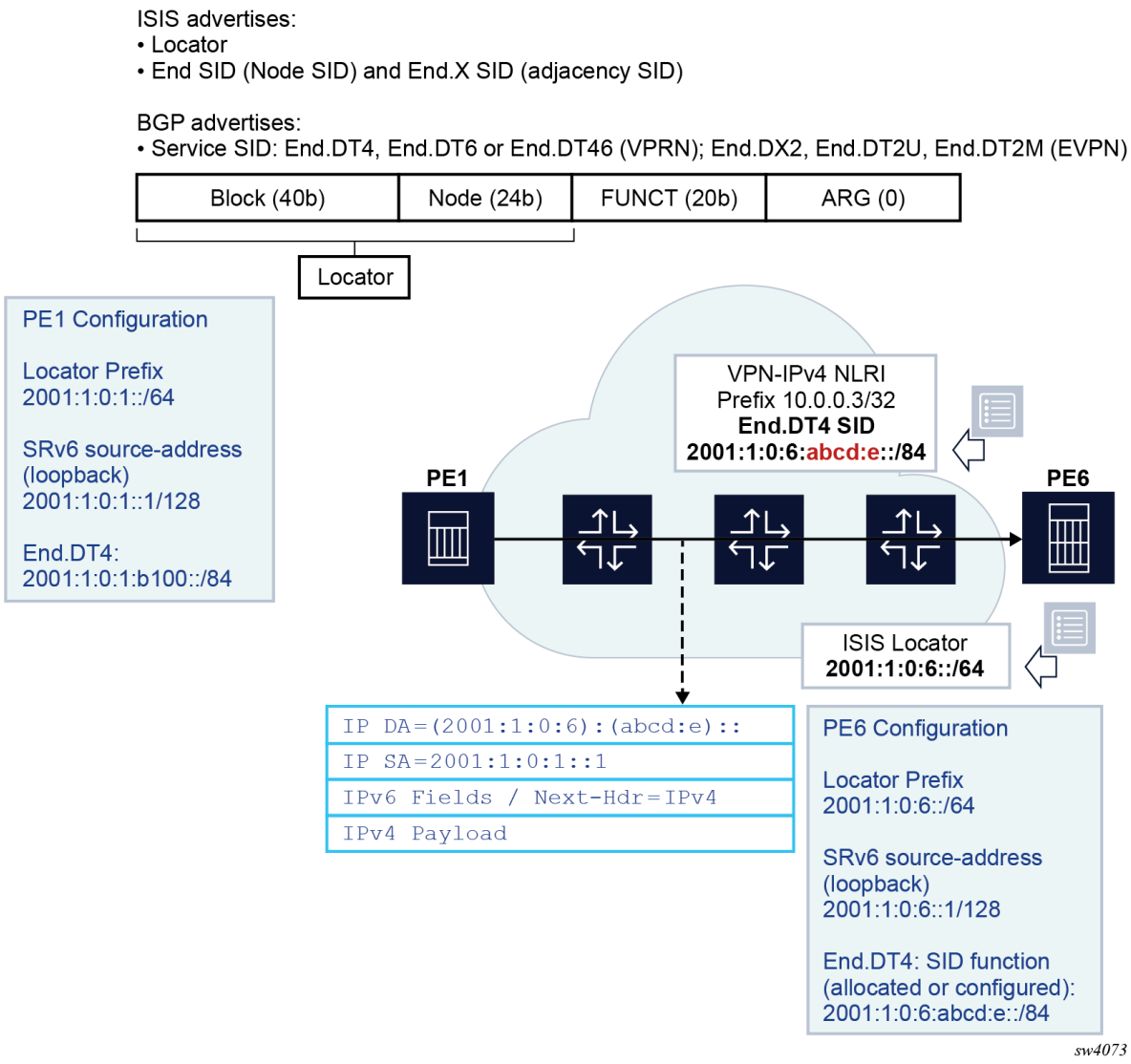
Figure 36: SRv6 SID encoding

The 128-bit address of an SRv6 SID is split into a three-field structure: {LOCATOR:FUNCTION:ARGUMENT}. The size of these fields is configurable.

- LOCATOR field - encodes the transport or reachability information
- FUNCTION field - encodes the node SID function (End SID), an adjacency SID function (End.X SID), or a service function that is the equivalent of a service label in SR-MPLS
- ARGUMENT field - typically encodes a value that identifies the source Ethernet Segment for EVPN services that require multihoming or Etree procedures; can be used to carry limited service or application metadata

The following figures shows the operation of the data and control planes when an IP-VPN route is resolved to an SRv6 tunnel.

Figure 37: SRv6 data and control plane operation



The PE6 egress router advertises the locator route that contains its locator prefix and, optionally, a local node SID (End) in IS-IS. It also advertises the SID of each adjacency (End.X) to its IS-IS neighbors.

The locator prefix provides the route to reach PE6 and is used by other routers to forward an SRv6 packet destined for any SID owned by PE6. Other routers use the End and End.X SIDs to create the repair tunnel for the Remote LFA and TI-LFA backup paths.

In BGP, PE6 advertises a VPN-IPv4 route and includes the End.DT4 service SID, which is equivalent to the SR-MPLS service label in the label per-VRF model. Unlike the SR-MPLS service label, the SRv6 End.DT4 SID contains both the function value that identifies the specific VRF-ID in PE6 and the locator prefix that provides the reachability to router PE6.

The PE1 router resolves the received VPN-IPv4 route by validating the next hop and checking the reachability of the locator prefix of PE6 in the routing table. When PE1 receives an IPv4 packet from a CE node, it pushes an outer IPv6 header that contains the End.DT4 SID in the DA field and looks up the

address in the routing table. The packet is then forwarded to one of the next hops of the route of the locator prefix of PE6.

SR OS also supports micro-segment SRv6. The system supports operating in both modes (regular and micro) concurrently on the same platform, but requires the SIDs of each type to come from different SID blocks.

Micro-segment SRv6 provides the same functionality as regular SRv6 but uses 16-bit SIDs while regular SRv6 uses 128-bit SIDs. The 16-bit SIDs are referred to as micro-SIDs.

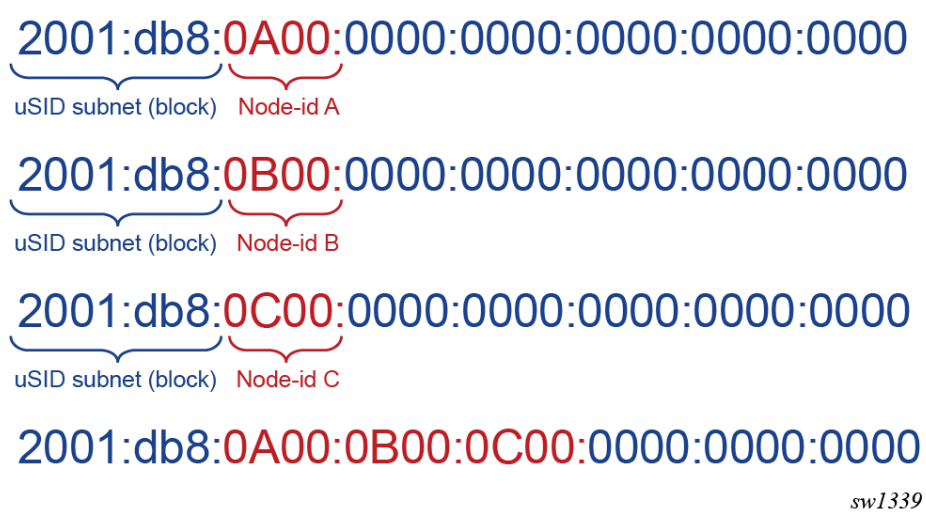
Micro-SIDs follow the general structure of regular SIDs (LOCATOR:FUNCTION:ARGUMENT), although the following differentiates the two.

In regular SRv6, identifiers are assigned to nodes and form, together with the SID block, the LOCATOR part of any SID. SIDs are assigned per node and associated with a specific behavior (or function), resulting in the LOCATOR:FUNCTION structure. Micro-segment SRv6 introduces a specific function (uN) which acts both as a locator (when combined with the block) and as a function (corresponding to END in regular SRv6) but without using any FUNCTION bits. In other words, the LOCATOR and END constructs of regular SRv6 correspond to a unique construct in micro-segment SRv6: <block><uN>.

Micro-segment SRv6 allows micro-SIDs to be compressed. Compression consists in coalescing several micro-SIDs into a 128-bit structure called a container.

The following figure shows three micro-SIDs, each identifying a different node in a network. The figure shows the result of compressing these three micro-SIDs in a container. Compression is possible because all micro-SIDs belong to the same block.

Figure 38: Compression of three micro-SIDs in a container



A container can be placed in the DA field of an IPv6 header and in a segment of the segment list of an SRH.

Compression allows the router to build longer paths with less overhead than with regular SRv6. However, it leads to specific datapath behaviors. See Datapath support for more information.

Micro-segment SRv6 introduces specific micro-segment SRv6 functions that correspond to functions defined for regular SRv6 (see Micro-segment SRv6 and Overview of the BGP requirements).

In general, any SR OS capability that applies to regular SRv6 also applies to micro-segment SRv6. In the documentation, only the differences are highlighted and commonalities are not repeated using micro-segment SRv6 specific terminology.

3.2 Configuring the locator and SIDs

3.2.1 Configuring the SRv6 locator and SIDs

This section describes configuration of the SRv6 locator.

An SRv6 SID is a 128-bit IPv6 address which follows the structure defined by RFC 8986:

SRv6 SID={LOCATOR:FUNCTION:ARGUMENT}.

The user must configure the main SRv6 subnet for this node. This is the locator and is, essentially, an IPv6 subnet (prefix and length) that provides reachability (longest prefix match) to all the SIDs originated by this node. The prefix is encoded in the LOCATOR field of the SRv6 SID and can have a length of either 64 or 96 bits.

The locator is further subdivided into a SID block and a node ID. For example, locator 3FFE:0:0:A1::/64 has a SID block of 3FFE:0 and node ID of 0:A1.

All nodes participating in an SRv6 domain must draw their locator and SIDs from the same SID block. In the previous example, the SID block is subnet 3FFE:0::/32.

Use the following commands to configure the SRv6 locator.

- **MD-CLI**

```
*[pr:/configure router "Base" segment-routing segment-routing-v6 locator "test"]
A:admin@test# tree detail
+-- admin-state <keyword>
+-- algorithm <number>
+-- apply-groups <reference>
+-- apply-groups-exclude <reference>
+-- block-length <number>
+-- function-length <number>
+-- label-block <reference>
+-- prefix
|   +-- ip-prefix <global-unicast-ipv6-address-and-prefix>
+-- static-function
|   +-- label-block <reference>
|   +-- max-entries <number>
+-- termination-fpe <reference>
+-- origination-fpe <reference>
+-- source-address <global-unicast-ipv6-address>
```

- **classic CLI**

```
configure+--router
   +--segment-routing
      +--segment-routing-v6
         +--origination-fpe* <fpe>
         +--source-address <ipv6-address>
         +--locator* <locator-name>
            +--admin-state (enable|disable)
```

```
            +--termination-fpe <fpe>
            +--algorithm <0, 128-255>
            +--prefix
               +--ip-prefix <ipv6-address/prefix-length>
            +--block-length <0-96>
            +--function-length <16 | 20-96>
            +--label-block <block-name>
            +--static-function
                      +--max-entries <integer>
                      +--label-block <block-name>
```

The two reserved label ranges (**locator>label-block** and **locator>static-function>label-block**) are mutually exclusive.

If **locator>static-function>label-block** is configured, then the labels associated with the service functions are drawn as follows:

- from the label block for static service function
- from the dynamic label range for dynamic service functions

If **locator>label-block** is configured, labels associated with service functions are drawn from the label block for both static and dynamic service functions.

Configuring a function length of 16 requires the use of **locator>label-block**. Function lengths of 16 are not supported on **locator>static-function>label-block**.

One locator is required for algorithm 0 and one for each IGP flexible algorithm (128-255). The same locator can be shared by multiple IGP instances for the same algorithm number.

The locator prefix,(in this example, 3FFE:0:0:A1::/64), is advertised in the SRv6 Locator TLV in IS-IS in both algorithm 0 and any configured flexible algorithm number, as defined in *draft-ietf-lsr-isis-srv6-extensions*. It is also advertised as a prefix in IP Reachability TLV (IS-IS TLV 236) in algorithm 0, so the routers that do not support SRv6 can still route the packet to the next hop of the locator and, eventually, to its destination node.

The FUNCTION field is user-configurable. The ARGUMENT field is set to all zeros and is not configurable. The ARGUMENT length must be signaled as zero in the IS-IS End and End.X SIDs and in the BGP service SIDs. Also, the sum of the LOCATOR and FUNCTION lengths must be less than or equal to 128.

Within algorithm 0 and each IGP flexible algorithm, the locator function (FUNCTION field) assigns the values of the End SID and End.X SID which correspond to the node SID and the adjacency or adjacency SET SID, respectively.

The locator function also assigns the value of the service SIDs owned by this node and advertised in the BGP control plane (End.DT4, End.DT6, End.DT46, and End.DX2).

The FUNCTION field can be subdivided into a static and a dynamic subrange. The user can draw from the static subrange to manually assign an SRv6 SID to a node, a local adjacency, or a service. IS-IS and BGP can draw from the dynamic subrange to assign a SID to a local adjacency or a service. The SID of an adjacency to an IS-IS neighbor, over a broadcast interface (LAN End.X), is always dynamically assigned and is not configurable.

The following CLI commands enable the allocation of an SRv6 SID function value. Manual allocation of a static function value is supported for nodes (End SIDs), adjacencies over a P2P interface (End.X SIDs), and service SIDs. Auto-allocation is supported for adjacencies over a P2P interface (End.X SIDs) and service SIDs.

- **MD-CLI**

```
[pr:/configure router "Base" segment-routing segment-routing-v6]
```

```
A:admin@test# tree detail
+-- base-routing-instance
|   +-- apply-groups <reference>
|   +-- apply-groups-exclude <reference>
|   +-- locator <reference>
|       +-- function
|           +-- end <number>
|           |   +-- srh-mode <keyword>
|           +-- end-x-auto-allocate <keyword> protection <keyword>
|           +-- end-x <number>
|           |   +-- interface-name <reference>
|           |   +-- protection <keyword>
|           |   +-- srh-mode <keyword>
|           +-- end-dt4
|           +-- end-dt46
|           +-- end-dt6
```

- **classic CLI**

```
configure
+--router
+---segment-routing
|   +---segment-routing-v6
|   |   +--- base-routing-instance
|   |   |   |   locator <locator-name>
|   |   |   |   +---function
|   |   |   |   |   |   end <integer>
|   |   |   |   |   |   +---srh-mode <psp | usp>
|   |   |   |   |   +---end-x-auto-allocate <psp|usp> <protected|unprotected>
|   |   |   |   |   +---end-x <integer>
|   |   |   |   |   |   +---interface <name>
|   |   |   |   |   |   +---srh-mode <psp | usp>
|   |   |   |   |   |   +---protection <protected|unprotected>
|   |   |   |   |   +---end-dt4 [<integer>]
|   |   |   |   |   +---end-dt6 [<integer>]
|   |   |   |   |   +---end-dt46 [<integer>]
```

3.2.2 Configuring the micro-segment locator and SIDs

Use the commands in the following contexts to configure micro-segment locators and SIDS.

```
configure router segment-routing segment-routing-v6 micro-segment
configure router segment-routing segment-routing-v6 micro-segment-locator
```

The following commands in micro-segment SRv6 apply to any (micro-segment) locator:

- **block-length**
  This command provides the same function as for regular SRv6. The value must be the same on every platform network wide.
- **global-sid-entries**
  This command defines the maximum number of unique micro-segment locators that can be configured network wide. The value must be the same on every platform network wide. This command splits in two the total number of values that can be encoded in 16 bits. A 16-bit field allows values in the range 0x0000 to 0xFFFF. Configuring 16 (default value) for the **global-sid-entries** value splits that range into the following:
  – 0x0000 to 0x3FFF for global identifiers

  - 0x4000 to 0xFFFF for local identifiers

- **sid-length**
  This command defines the length of micro-SIDs. The only supported value is 16 (bits).

To configure a **block**, assign a name to it and configure specific commands associated with the block. Those commands include:

- **label-block**
  The reserved label block serves both static and dynamic service functions. The theoretical maximum number of functions in micro-segment SRv6 is 2^{16}. A label block that provides more values than the theoretical maximum leads to a waste of labels. Because in practice, the number of local functions is less than (2^{16} - **global-sid-entries**), a **label-block** should not need exceed that value.
- **prefix ip-prefix** *ipv6-prefix/prefix-length*
  The IP prefix configures the block as an IPv6 address. The *prefix-length* value must be equal to the **block-length**.
- **static-function max-entries**
  This command sets the maximum number of static functions the user needs.

To configure a **micro-segment locator**, use the following configuration knobs that are specific for micro-segment SRv6:

- **block**
  This command configures a reference (by name) to a previously configured block.
- **un value**
  This command creates a node identifier as an IPv6 address composed of the block part followed by a 16-bit SID. The value of the SID is the n^{th} global SID entry. The value, which must be unique network wide, is configured as part of the locator because it acts as a locator.

The configuration of micro-SIDs follows the same logic as for regular SRv6 SIDs except that:

- The user configures SIDs that are specific to micro-segment SRv6 (uA, uDT4, uDT6, uDT46).
- The configured value only needs to be unique on the system and not network wide.
- The resulting SID value is derived from the local range.
- Although the uN function is equivalent to the regular SRv6 END function, it is not configurable under the Base instance because it is configured in the **micro-segment-locator** context.

3.3 IS-IS control plane extensions

Use the commands in the following context to enable SRv6 in the IS-IS instance and assign a locator to each algorithm (0 or flexible algorithm).

```
configure router isis segment-routing-v6 locator
```

The IS-IS control plane extensions in support of SRv6 are defined in *draft-ietf-lsr-isis-srv6-extensions*.

The IS-IS control plane advertises the SRv6 capabilities sub-TLV and the SRv6 Locator TLV. The latter includes the End function sub-TLV (equivalent to the prefix SID sub-TLV in SR-MPLS). IS-IS also advertises the End.X function sub-TLV (equivalent to the adjacency SID sub-TLV for a P2P link and a LAN in SR-MPLS) in the Extended IS Reachability TLV (top-level link TLV).

The weight fields in the End.X and LAN End.X sub-TLVs are not filled in on transmit and are ignored on receipt of the link TLV.

The following table describes the supported IS-IS SRv6 TLVs in SR OS.

Table 19: SRv6 IS-IS TLVs

SRv6 TLV/sub-TLV	Codepoint	IS-IS context TLV	Description	SR OS support
SRv6 Capabilities sub-TLV	25	Router CAPABILITY TLV (242)	Indicates SRv6 support	Yes
SR-Algorithm sub-TLV	19	Router CAPABILITY TLV (242)	Indicates base algorithm 0 and Flex-Algo 128-255 support	Yes
Maximum Segments Left MSD Type sub-TLV	41	Router CAPABILITY TLV (242)	Indicates how deep a node terminating current segment can process Segments Left field of the SRH to move the next SID to outer IPv6 header DA field	Yes (Advertised value = 10 SIDs) Received TLV is displayed in the LSDB but is not used for any purpose
Maximum End Pop MSD Type sub-TLV	42	Router CAPABILITY TLV (242)	Maximum number of SIDs in a SRH when a node removes the SRH (Penultimate Segment Pop (PSP) or Ultimate Segment Pop (USP) modes of SRH)	Yes (Advertised value = 9 SIDs) Received TLV is displayed in the LSDB but is not used for any purpose
Maximum H.Encaps MSD type sub-TLV	44	Router CAPABILITY TLV (242)	Indicates the maximum number of SIDs in an SRH that a router can push when forwarding an IP or L2 packet over a SRv6 policy	Yes (Advertised value = 7 SIDs) Received TLV is displayed in the LSDB but is not used for any purpose
Maximum End D MSD Type Sub-TLV	45	Router CAPABILITY TLV (242)	The maximum number of SIDs in an SRH when a node removes the SRH and performs the End.DX2/4/6 or End.DT4/6 function (USP mode of SRH)	Yes (Advertised value = 9 SIDs) Received TLV is displayed in the LSDB but is not used for any purpose

SRv6 TLV/sub-TLV	Codepoint	IS-IS context TLV	Description	SR OS support
SRv6 Locator TLV	27	Is a Top-level IS-IS TLV	Advertises the locator prefix configured on this node to terminate SIDs in algorithm 0 and flex-algo 128-255	Yes
SRv6 End SID sub-TLV	5	SRv6 Locator TLV	Advertises the SID for the endpoint or End function (equivalent to the prefix SID sub-TLV in SR-MPLS)	Yes
Prefix Attribute Flags Sub-TLV	4	SRv6 Locator TLV (Also in IP Reach TLV 236)	Provides attributes of a prefix that is leaked between IS-IS levels	Yes
SRv6 End.X SID sub-TLV	43	Top-level Extended IS reachability TLV (22)	Advertises the SID for the adjacency over a P2P link (equivalent to the adjacency SID sub-TLV for P2P link in SR-MPLS)	Yes
SRv6 LAN End.X SID sub-TLV	44	Top-level Extended IS reachability TLV (22)	Advertises the SID for the adjacency over a LAN (equivalent to the adjacency SID sub-TLV for LAN in SR-MPLS)	Yes
SRv6 SID Structure Sub-Sub-TLV	1	SRv6 End SID Sub-TLV SRv6 End.X SID Sub-TLV SRv6 LAN End.X SID Sub-TLV	Provides the length of each field (Block, Locator, Function, and Argument) of the SRv6 SID that it is advertised with	No SR OS does not advertise this sub-sub-TLV. If received from other vendor's implementation, it is not displayed in the Link-State database and is also not propagated with the locator TLV

When both SR-MPLS and SRv6 are enabled on the same IS-IS instance, an MPLS node SID cannot be configured for a prefix of the locator or an End SID. This is because the SRv6 locator subnet cannot be added to a network interface and an MPLS node SID is configurable against a network interface only.

However, both the SRv6 locator tunnel and the SR-MPLS tunnel are programmed if IS-IS receives a /128 prefix that has both a locator TLV and a prefix SID TLV (with the node flag enabled) from a third-party

router implementation. If the prefix SID is for a subnet larger than /128, only the SRv6 locator tunnel is programmed and the SR-MPLS tunnel is not.

Each function encoded in a SRv6 SID has its own endpoint behavior codepoint as listed in Table 20: SRv6 SID function endpoint behavior codepoints.

Note:
SRv6 standards provide the flexibility to advertise transport and service SIDs in both IS-IS and BGP. In SR OS, the IS-IS control plane only advertises the transport SID functions shown in Table 20: SRv6 SID function endpoint behavior codepoints and only uses the transport SIDs advertised in IS-IS for building LFA repair tunnels.

SR OS advertises service SID functions in the BGP control plane only as described in BGP service control plane extensions.

For the SRH processing and removal at the SID termination, the following modes of operation are associated with the termination of the End or End.X SID. These modes are sometimes referred to as SID flavors and IS-IS assigns a unique codepoint for each mode of the End or End.X SID of a given adjacency.

- **Basic or unflavored SRH mode**

  The router that terminates an End or End.X SID and the Segments-Left field in the received packet is 0 before decrementing, keeps the SRH in the packet, and processes the packet identified by the next-header in the SRH. Typically, the next-header indicates another SRH and the packet is then forwarded based on the lookup of that next-SID; SR OS does not support this mode.

- **Ultimate Segment Pop (USP) SRH mode**

  The egress PE terminates the last segment in the outer IPv6 header, removes the SRH and processes the inner service or control plane packet as indicated by the SRH Next-Header field. SR OS supports this mode.

- **Penultimate Segment Pop (PSP) SRH mode**

  The router that terminates the End or End.X segment before the last in the segment list, meaning the Segments-Left field before decrementing has a value of 1, removes the SRH on behalf of the egress PE. SR OS supports this mode.

- **PSP&USP mode**

  This is a combination of both the USP and PSP modes. The router that terminates the End or End.X segment applies the corresponding behavior for value 0 and 1 of the Segments-Left field. SR OS does not support this mode.

- **Ultimate Segment Decapsulation (USD) mode (PSP&USD, USP&USD, and PSP&USP&USD flavors)**

  This is a variant of the USP mode in which the node removes the outer IPv6 header and associated SRH and moves directly to process the inner IPv6 packet as indicated by the SRH Next-Header field. This mode is used to terminate a TI-LFA or a Remote LFA repair tunnel originated with the H.Encaps.Red encapsulation.

  An SRv6 ingress PE or transit P router normally uses the H.Insert.Red behavior to build an LFA (remote LFA or a TI-LFA) repair tunnel. This inserts a dedicated SRH to carry the additional node and adjacency SIDs of the repair tunnel. SR OS supports originating and terminating repair tunnels using H.Insert.Red behavior.

  Some third party implementations use the H.Encaps.Red behavior, which inserts a dedicated outer IPv6 header and SRH to carry the additional SIDs of the LFA repair tunnel. While SR OS does not support

originating an LFA repair tunnel with the H.Encaps.Red behavior, it does support terminating the repair tunnel.

To allow third party implementations to activate the H.Encaps.Red repair tunnel, IS-IS in SR OS can signal support for the USD mode of the outer IPv6 header and SRH processing with End and End.X SIDs in classic SRv6, and with uN and uA in micro-segment SRv6.

The user can configure the advertisement of the PSP&USD, USP&USD, or PSP&USP&USD flavor of the USD mode for each configured and auto-allocated node SID or adjacency SID. The PSP&USP&USD behavior is similar to the PSP&USD behavior because the datapath in SR OS always performs the USP SRH mode when Segments-Left = 0 on a received SRv6 packet.

Note: On the 7750 SR, when handling a uN+uA, the implementation always applies PSP&USP, regardless of the flavors associated with the uN and the uA.

Table 20: SRv6 SID function endpoint behavior codepoints

SID function endpoint behavior	Codepoint (per RFC 8986)	SID type: End.SID	SID type: End.X SID	SID type: LAN End.X SID	Advertising protocol	Supported
End (PSP, USP, USD)	1-4, 28-31	Yes	No	No	IS-IS	Yes IS-IS only (PSP value = 2, USP value = 3) (PSP&USD value = 29, USP&USD value = 30) (PSP&USP&USD value = 31)
End.X (PSP, USP, USD)	5-8, 32-35	No	Yes	Yes	IS-IS	Yes IS-IS only (PSP value = 6, USP value = 7) (PSP&USD value = 33, USP&USD value = 34) (PSP&USP&USD value = 35)
End.T	9-12, 36-39	Yes	No	No	IS-IS	No[19]

[19] IS-IS saves SID sub-TLVs for endpoint behavior values that it does not support, if received from a third-party implementation. However, it only uses End and End.X endpoint behaviors in RLFA and TI-LFA. BGP advertises the supported endpoint behaviors End.DT4, End.DT6, End.DT46, and End.DX2 and accepts any behavior codepoint with a supported NLRI type.

SID function endpoint behavior	Codepoint (per RFC 8986)	SID type: End.SID	SID type: End.X SID	SID type: LAN End.X SID	Advertising protocol	Supported
(PSP, USP, USD)						
End.DX6	16	No	Yes	Yes	BGP or IS-IS	No[19]
End.DX4	17	No	Yes	Yes	BGP or IS-IS	No[19]
End.DT6	18	Yes	No	No	BGP or static	Yes[19] BGP only
End.DT4	19	Yes	No	No	BGP or static	Yes[19] BGP only
End.DT46	20	Yes	No	No	BGP or static	Yes[19] BGP only
End.DX2	21	Yes	No	No	BGP or static	Yes[19] BGP only

3.3.1 Micro-segment SRv6

Use commands in the following context to enable micro-segment SRv6 in the IS-IS instance and to assign a micro-segment locator to each algorithm (zero or flexible algorithm).

```
configure router isis segment-routing-v6 micro-segment-locator
```

The following table lists the TLVs that micro-segment SRv6 supports in addition to the TLVs described in Table 19: SRv6 IS-IS TLVs.

Table 21: SRv6 IS-IS TLVs additionally supported by micro-segment SRv6

SRv6 TLV/sub-TLV	Codepoint	IS-IS context TLV	Description	SR OS support
SRv6 SID Structure sub-sub-TLV	1	SRv6 uN SID Sub-TLV, SRv6 uA SID Sub-TLV, SRv6 LAN uA SID Sub-TLV	Provides the length of each field (block, locator, function, and argument) of the SRv6 uSID it is advertised with	Yes

The following table lists the endpoint behavior codepoint for each SID function encoded in a micro-segment SRv6 SID.

Table 22: SRv6 SID function endpoint behavior codepoints

SID function endpoint behavior	Codepoint	SID type: End.SID	SID type: End.X SID	SID type: LAN End.X SID	Advertising protocol	Supported
uN	42-50	Yes	No	No	IS-IS	Yes [20] IS-IS only (PSP value = 44, USP value = 45) (PSP&USD value = 48, USP&USD value = 49) (PSP&USP&USD value = 50)
uA	51-59	No	Yes	Yes	IS-IS	Yes [20] IS-IS only (PSP value = 53, USP value = 54) (PSP&USD value = 57, USP&USD value = 58) (PSP&USP&USD value = 59)

3.3.2 SRv6 support in IS-IS multitopology

This feature extends the support of SRv6 to multitopology IPv6 (MT2) in IS-IS.

3.3.2.1 Feature configuration

The user first configures a locator for use in IS-IS IPv6 unicast multitopology. Use the following CLI syntax to configure a locator for use in IS-IS IPv6 unicast multitopology:

- **classic CLI**

```
configure
+--router
   +--isis <isis-instance>
```

[20] IS-IS saves SID sub-TLVs for endpoint behavior values that it does not support, if received from a third-party implementation. However, it only uses End and End.X endpoint behaviors in RLFA and TI-LFA. BGP advertises the supported endpoint behaviors End.DT4, End.DT6, End.DT46, and End.DX2 and accepts any behavior codepoint with a supported NLRI type.

```
      +--segment-routing-v6
        +---no adj-sid-hold
        |   adj-sid-hold <seconds>
        +---locator <name>
        |   no locator <name>
        |   +---level {1|2}
        |   |   +---metric <metric>
        |   |   |   no metric
        |   +---level-capability {level-1|level-2|level-1/2}
        |   |   no level-capability
        |   +---multi-topology [mt0] [mt2]
        |   |   no multi-topology
        |   +---no tag
        |   |   tag <tag>
        +---shutdown
        |   no shutdown
```

- **MD-CLI**

```
configure
+--router
   +--isis <isis-instance>
      +--segment-routing-v6
         +-- adj-sid-hold <seconds>
         +--admin-state (enable|disable)
         +--locator <locator-name>
            +--level-capability <level-1|level-2|level-1/2>
            +--level <1|2>
               +--metric <metric>
            +--tag <tag>
        +---multi-topology [mt0] [mt2]
```

The user can enable one or more SRv6 local locators in a specific IS-IS instance. Each locator can be enabled in a single topology of an IS-IS instance, topology 0 (MT0) or topology 2 (MT2). By default, a locator name added to an IS-IS instance is enabled in MT0. A local locator can be used in multiple IS-IS instances and can be assigned to at most one IPv6 topology independently within each IS-IS instance.

Note: To enable processing of local and remote IPv6 prefixes and SRv6 locators in MT0 and MT2, use the **admin-state enable** MD-CLI command in the **configure router isis segment-routing-v6** context. The user must also enable IPv6 routing in MT0 (using the **isis ipv6-routing native** command), in MT2 (using the **isis multi-topology ipv6-unicast** command), or in both to enable SRv6 forwarding in the appropriate topologies.

3.3.2.2 IS-IS control plane changes

When a local locator is enabled in the MT2 IPv6 unicast topology of an IS-IS instance, IS-IS advertises the following routes:

- The prefix in a top level Multitopology Reachable IPv6 Prefixes TLV type 237 with the MT-ID field set to 2.
- A top level SRv6 Locator TLV type 27 that contains the locator prefix as well as the End SID sub-TLVs associated with this local locator. The locator TLV is advertised with the MT-ID field set to 2.
- The End.X or LAN End.X sub-TLV in the top level multitopology Extended IS Reachability TLV (222).
- The TE Application Specific Link Attributes sub-TLV in the top level multitopology Extended IS Reachability TLV (222), which supports the flex-algo feature.

- The Application-Specific Shared Risk Link Group (SRLG) TLV (238).

The following table summarizes the new or modified TLVs in support of SRv6 in IS-IS MT2.

Table 23: SRv6 IS-IS MT2 TLVs

Multitopology TLV	Codepoint	IS-IS context TLV	Description	SR support
Multitopology IPv6 Reach TLV	237	Is a top-level IS-IS TLV	The main prefix TLV MT-ID field set to 2	Yes
SRv6 Locator TLV	27	Is a top-level IS-IS TLV	Indicates that the locator, configured on this node, is used to terminate SIDs in algorithm 0 and flex-algo 128-255 MT-ID field set to 2	Yes
SRv6 End SID Sub-TLV	5	SRv6 Locator TLV	Advertises the SID for the endpoint or End function (equivalent to the prefix SID in SR-MPLS)	Yes
Prefix Attribute Flags Sub-TLV	4	SRv6 Locator TLV (also in IPv6 Reach TLV 237)	Indicates that the prefix of the locator is anycast	Yes
SRv6 End.X SID Sub-TLV	43	Top-level Multitopology Extended IS Reachability TLV (222)	Advertises the SID for the adjacency or End.X function over a P2P link (equivalent to the adjacency SID sub-TLV for P2P link in SR-MPLS)	Yes
SRv6 LAN End.X SID sub-TLV	44	Top-level Multitopology Extended IS Reachability TLV (222)	Advertises the SID for the adjacency or End.X function over a LAN (equivalent to the adjacency SID sub-TLV for LAN in SR-MPLS)	Yes
SRv6 SID Structure Sub-Sub-TLV	1	SRv6 End SID Sub-TLV, SRv6 End.X SID Sub-TLV, SRv6 LAN End.X SID Sub-TLV	Provides the length of each field (Block, Locator, Function, and Argument) of the SRv6 SID that it is advertised with	No SR OS does not advertise this sub-sub-TLV. If received from other vendor's implementation, it is not displayed in Link-State database and is also not propagated with the locator TLV.

Multitopology TLV	Codepoint	IS-IS context TLV	Description	SR support
Application Specific Link Attributes (ASLA) sub-TLV	16	Top-level Multitopology Extended IS reachability TLV (222)	Advertises the link attributes for the flex-algo application	Yes
Application-Specific Shared Risk Link Group (SRLG) TLV	238	Is a top-level IS-IS TLV	Advertises the link SRLG attribute for the flex-algo application	No The SRLG constraint is not supported in IS-IS MT0 or MT2. IS-IS does not advertise this TLV and does not use it for any purpose if received from the network.

In prior releases of SR OS, only local locator routes and local prefix routes were redistributed with an export policy between topologies MT0 and MT2 of the same IS-IS instance or different IS-IS instances. These local locators were redistributed as a prefix TLVs and not as a locator TLVs. In addition, the tag of the local locator was not redistributed. Routes of remote locators and remote prefixes were not redistributed.

To extend SRv6 at a boundary of two IS-IS instances operating IPv6 in different topologies, the SRv6 support in MT2 feature enhances this behavior as follows:

- It allows the concurrent enabling of a local locator in multiple IS-IS instances and in either MT0 or MT2 topology of each of these IS-IS instances.
- It allows the redistribution with an export policy of remote locator routes between MT0 and MT2 topologies of different IS-IS instances. The locator routes are redistributed as locator TLVs. In algorithm 0, a prefix TLV is advertised in addition to the locator TLV.

Redistribution between MT0 and MT2 topologies of the same IS-IS instance is not allowed for either remote locator or remote prefix routes.

3.3.2.3 Locator and SID resolution

The MT2 local locator, remote locator, SID resolution, and programming into the tunnel table and datapath tunnel follow the same rules as those for enabling SRv6 in the single topology (MT0).

If a remote IPv6 prefix route is received in both MT0 and MT2, the route with the lowest cost (IGP metric) is selected. If both routes have the same metric, the MT0 route is selected. The selected route is programmed in the route table and FIB. If that prefix also advertised a locator TLV, the corresponding SRv6 route is updated in the route table and FIB to point to the SRv6 tunnel which is programmed in the tunnel table.

Note:
The preference of the MT0 route over the MT2 route is solely based on comparing the cost of each route. So, a remote IPv6 prefix route without a locator TLV can win over a remote IPv6 route with a locator TLV. The programmed route in the route table is a regular IS-IS route and does not have an SRv6 tunnel associated with it.

The same selection rule applies to a locator TLV advertised with a flex-algo number in both MT0 and MT2. The selection is based on comparing the cost of the routes using the metric of that specific algorithm (IGP, TE, or latency metric). The selected SRv6 route is added to the tunnel table.

3.3.3 SRv6 locator summarization with IS-IS

If an IS-IS domain exists out of multiple areas, the operator must redistribute SRv6 locators between areas for inter-area SRv6-based transport services. Each SRv6 locator is associated with an applied topology algorithm as follows:

- algorithm 0 for the base SPF topology
- algorithm in the range of 128 to 255 for flexible algorithms

Scaling is impacted when all existing SRv6 locators are redistributed between all existing areas. SRv6 locator summarization with IS-IS reduces the sizes of the LSDB and the IPv6 routing table and increases network stability.

SRv6 locators can be summarized when they are redistributed from one area into another area. This helps to reduce the number of SRv6 locators existing in each area. A summary SRv6 algorithm-aware locator address is configured with the following syntax:

```
summary-address {ip-prefix/ip-prefix-length | ip-prefix netmask} [level] [tag tag] [algorithm <algo-
id>]
no summary-address {ip-prefix/ip-prefix-length | ip-prefix netmask}
```

The configuration parameters are as follows:

- *ip-prefix/ip-prefix-length* | *ip-prefix netmask* contains the summary locator address of the route
- *level* configures the level where the summary route is applied
- *tag* assigns a route tag to the summary address
- *algo-id* identifies the flex-algorithm topology for the summary address

  The following considerations apply:

  – When an algorithm-aware summary address is generated, all matching algorithm-aware member prefixes are suppressed.

  – When the algorithm is not explicitly configured, the summary address is only generated for algorithm 0 in an IP Reachability TLV, and all more specific prefixes are suppressed from both the IP reach TLV and the SRv6 locator TLV for algorithm 0.

  – When the algorithm is configured in the range 128 to 255 or 0, the summary address is generated only for member prefixes belonging to the configured algorithm (including explicit algorithm 0 configurations). Only member prefixes within the configured algorithm are summarized and suppressed, while member prefixes that belong to a different algorithm are not summarized and not suppressed.

The following apply when the summary address is configured and the route table has matching member prefixes:

- For algorithm 0, when algorithm 0 is not explicitly configured, the summary is inserted as an IS-IS IP Reachability TLV and all more specific prefixes from both IP Reachability TLV and SRv6 locator TLV are suppressed.
- For algorithm 0, when algorithm 0 is explicitly configured, the summary is inserted as an IS-IS IP Reachability TLV and as an SRv6 locator TLV, and all more specific prefixes from both the IP Reachability TLV and the SRv6 locator TLV are suppressed.
- For algorithms in the range of 128 to 255, the summary is inserted as an SRv6 locator TLV and all more specific locator prefixes in the SRv6 locator TLV are suppressed.
- The summary address SRv6 locator does not contain any END SIDs of member SRv6 locator prefixes in the SRv6 locator TLV.

The following apply when using administrative tags for SRv6 locator summaries:

- When SRv6 locators are summarized from one IS-IS level into another IS-IS level, special care must be taken to avoid re-redistributing them back into the original IS-IS level and potentially causing routing loops. Routing filters must be used to prevent such routing loops.
- Existing filters can use 32-bit administrative tags to match upon routes and avoid routing loops. This route tag can be set using the **summary-address** command when originating an algorithm-aware SRv6 summary locator.
- A routing policy with a match tag supports matching to both classic IPv6 prefix tags and SRv6 locator tags.

3.4 Configuring IS-IS Flex-Algorithm for SRv6

SRv6 introduces flexible algorithms to the IPv6 data plane.

A router is provisioned with topology- or algorithm-specific locators for each topology or algorithm pair supported by that node. Each locator is a covering prefix for all SIDs provisioned on that router that have the matching topology or algorithm. Locators associated with flexible algorithms are not advertised in a Prefix Reachability TLV (236 or 237). However, locators associated with algorithm 0 are advertised in a Prefix Reachability TLV (236 or 237), which allows legacy routers that do not support SRv6 to install a forwarding entry for algorithm 0 SRv6 traffic.

Each SRv6 locator is associated with an algorithm (either algorithm 0 or a flexible algorithm in the range of 128 to 255) and each algorithm represents a topologically-constrained forwarding construct. The M-flag within the flexible algorithm prefix metric sub-TLV is not applicable to prefixes advertised as SRv6 locators. The metric field in the locator TLV is used regardless of the M-flag in the FAD advertisement.

A router configured to participate in a flexible algorithm must use the selected FAD to compute the corresponding routing table. The available options are as follows:

- Algorithm 0 (legacy routing table entries) is constructed from information advertised as a traditional IP Reachability TLV or as an SRv6 locator TLV (27). When IP reach TLV and SRv6 locator TLV contain conflicting information, then the IP Reachability TLV information is used.
- Algorithms ranging from 128 to 255 (Flex-Algorithm routing table entries) are constructed from information advertised and constructed from locators found in the SRv6 locator TLV (27).

For route leaking of flexible algorithm-aware SRv6 locators between IS-IS areas, the following rules apply when a topology TLV (IP Reachability TLV or SRv6 locator TLV) is leaked, including leaked locators and end SIDs:

- For algorithm 0, this SRv6 locator route is programmed as regular IS-IS route. If an IS-IS route is readvertised and also has an SRv6 locator TLV, it is readvertised as a regular IP Reachability TLV and SRv6 locator TLV.
- For algorithms ranging from 128 to 255, if locator leaking is enabled, the original SRv6 locator TLV is readvertised as a SRv6 locator TLV into the other area.
- The default locator leaking behavior between levels is as follows:
  - For Level 1 to Level 2, leaking is enabled by default.
  - For Level 2 to Level 1, leaking is disabled by default.
  - Changing the default leaking behavior requires an export policy where the **prefix-list** keyword behavior is configured to match upon prefixes or locators found in the routing table regardless of the associated algorithm. The **prefix-list** keyword allows combined support for algorithm 0 locators (regular prefixes) and algorithm locators ranging from 128 to 255 (Flex-Algorithm prefixes).

```
configure
+---router
    +---policy-options
        +---[no] policy-statement <name>
            +---from
                +---[no] level [1|2]
                +---[no] prefix-list
                +---[no] protocol
                +---[no] tag
            +---to
                 +---<snip>
```

If a locator is associated with a flexible algorithm and the LFA is enabled, then LFA paths to the locator prefix must be calculated using the flexible algorithm in the corresponding topology to guarantee that they follow the same constraints as the calculation of the primary paths. LFA paths must only use SRv6 SIDs advertised specifically for the flexible algorithm. The LFA configuration is inherited from algorithm 0. The anycast behavior of SRv6 flexible algorithms is inherited from the standard algorithm 0 (standard SPF) SRv6 configuration.

The IS-IS neighbor advertisements are topology-specific and not algorithm-specific. Therefore, the SRv6 End.X SIDs inherit topology from the associated neighbor advertisement, but the algorithm is specified in the individual SID. All End.X SIDs are a subnet of a locator with matching topology and algorithm which is advertised by the same node in an SRv6 locator TLV. The End.X SIDs that do not meet this requirement are ignored. All End.X SIDs must find a supernet by the subnet of a locator with the matching algorithm which is advertised by the same router in an SRv6 locator TLV. The End.X SIDs that do not meet this requirement are ignored.

IS-IS protocol limitations affect enabling SRv6 flexible algorithms on a broadcast network. On a broadcast network, the LAN End.X SIDs of all neighbors for all participating flexible algorithms need to be advertised in a single LSP fragment because each IS-IS TE-NBR with all its TLV blocks must be advertised in one IS-IS LSP fragment. The amount of information inserted by segment routing for SRv6 into the LSP fragment depends upon the number of the flexible algorithms used, the number of static or auto-end.X configured per locator, and if both SRv6 and SR-MPLS are deployed.

3.5 BGP service control plane extensions

This section provides an overview of the BGP service control plane extensions.

3.5.1 Overview of the BGP requirements

The BGP service control plane required extensions are specified in RFC 9252, *BGP Overlay Services Based on Segment Routing over IPv6 (SRv6)*. BGP requires some changes in the IPv6, VPN-IPv4, VPN-IPv6, and EVPN family routes so that the egress PE can signal the following End programming behaviors to the ingress PE:

- Layer 2 SRv6 service SIDs

  End.DX2 Layer 2 decapsulation and cross-connect to an Epipe egress SAP, signaled by AD per-EVI routes

- Layer 3 SRv6 service SIDs

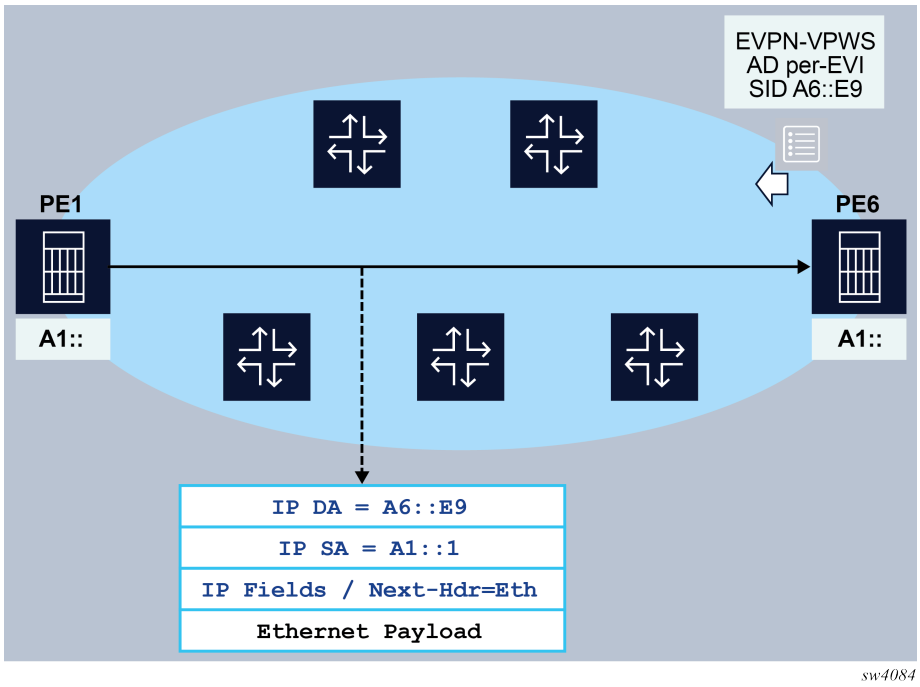
  End.DT4 a VPRN (or GRT) route-table lookup, signaled by VPN-IPv4 or EVPN in Interface-less (EVPN-IFL) IPv4 prefix routes (also by IPv4)

  End.DT6 a VPRN (or GRT) route-table IPv6 lookup, signaled by VPN-IPv6 or EVPN-IFL IPv6 prefix routes (also by IPv6)

  End.DT46 a VPRN route-table lookup for IPv4 or IPv6 prefixes, signaled by VPN-IPv4 or VPN-IPv6 or EVPN-IFL IPv4 or IPv6 prefix routes

The following figure shows an example for the End.DX2 behavior for EVPN-VPWS services.

Figure 39: End.DX2 behavior for EVPN-VPWS



The ingress and egress PEs behave as follows:

- The egress PE (PE6) advertises an A-D per EVI route with the SRv6 Service SID that identifies the End.DX2 behavior. The service SID includes the configured locator in the Epipe (A6::), as well as the allocated function (E9), which identifies the Epipe at the egress PE.

- The ingress PE (PE1) imports the A-D per EVI route and creates an EVPN destination in the corresponding Epipe to A6::E9.
- When PE1 receives frames at the access SAP, it encapsulates the frames into an SRv6 packet using the configured IP SA. The IP DA is the EVPN destination SID.
- Shortest path forwarding is considered in the example shown in Figure 39: End.DX2 behavior for EVPN-VPWS, and therefore the EVPN destination SID is encoded in the IP DA. If TI-LFA is required, PE1 modifies the encapsulation to include an SRH and additional SIDs.
- When the SRv6 packet arrives at PE6, the SID encoded in the IP DA identifies the packet for termination on PE6 and the Epipe for decapsulation and forwarding.

Similar procedures are followed for the other required services.

The following table lists the functions that micro-segment SRv6 implements to support the requirements.

Table 24: Micro-segment SRv6 functions

SID function endpoint behavior	Codepoint	SID type: End.SID	SID type: End.X SID	SID type: LAN End.X SID	Advertising protocol	Supported
uDT6	62	Yes	No	No	BGP or static	Yes[21]
uDT4	63	Yes	No	No	BGP or static	Yes [21]
uDT46	64	Yes	No	No	BGP or static	Yes [21]
uDX2	65	Yes	No	No	BGP or static	Yes [21]
uDT2U	67	Yes	No	No	BGP or static	Yes [21]
uDT2M	68	Yes	No	No	BGP or static	Yes [21]

3.5.2 BGP extensions

The following BGP extensions are supported, as defined in RFC 9252:

- The following SRv6 Service TLVs:
  - SRv6 Service TLV encoded in the BGP Prefix-SID attribute
  - SRv6 SID Information Sub-TLV (SRv6 Service Sub-TLV type 1) encoded in the SRv6 Service TLV

[21] BGP advertises the supported endpoint behaviors and accepts any behavior codepoint with a supported NLRI type

  - SRv6 SID Structure Sub-Sub-TLV (SRv6 Service Data Sub-Sub-TLV type 1)
- Transposition of 16 or 20 bits of the FUNCTION to the Label field of the NLRI
- Arg.FE2 arguments used for split-horizon filtering in EVPN multihoming, advertised in the EVPN AD per-ES routes with 16 bits of the argument transposed to the label field of the ESI Label extended community.

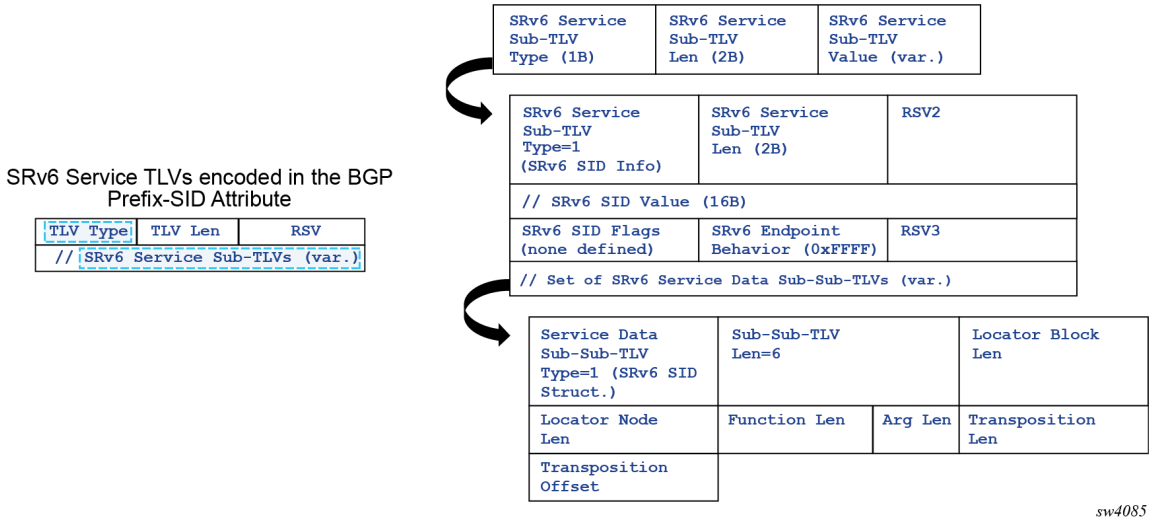
The BGP extensions are applied to the following routes by setting the behavior field in the SRv6 Services TLV, as defined in RFC 8986:

- VPN-IPv4
- VPN-IPv6
- IPv4
- IPv6
- EVPN AD per-EVI
- EVPN AD per-ES
- EVPN MAC/IP
- EVPN Inclusive Multicast Ethernet Tag

3.5.3 Advertising SRv6 service TLVs

The EVPN, VPN-IPv4, VPN-IPv6, IPv4, and IPv6 routes for the SRv6-enabled services are advertised along with the SRv6 Service TLV. The TLV format is described in RFC 9252 and shown in the following figure.

Figure 40: SRv6 service TLV format



The SRv6 Service TLV encoded in the BGP Prefix-SID attribute can have two different types:

- Type 5 is used for Layer 3 service SIDs or the SIDs signaled for VPRN services with VPN-IP or EVPN-IFL routes. Layer 3 service SIDs are also supported for the base router along with IPv4 or IPv6 routes.

- Type 6 is used for Layer 2 service SIDs or the SIDs signaled for Epipe or VPLS services. Type 6 is supported along with AD per-EVI and per-ES routes for Epipes.

The SRv6 Service TLV may contain an unordered list of sub-TLVs, but currently the SRv6 Service TLV is advertised with only one sub-TLV: the SRv6 SID Info sub-TLV (type 1). This sub-TLV encodes the following information:

- For the SID value, the entire 128-bit SID is allocated to the service, including the locator configured for the service and the allocated FUNCTION (which can be dynamically allocated or statically configured on the service). The ARGUMENT is always 0.
- SID flags are all zero.
- Endpoint behavior encodes the behavior as in RFC 8986, in decimal values. The following values are relevant for SR OS:
  - 18 – End.DT6
  - 19 – End.DT4
  - 20 – End.DT46
  - 21 – End.DX2
  - 24 – End.DT2m (only for AD per-ES routes)
- One SID Structure Sub-Sub-TLV (Service Data Sub-Sub-TLV type 1)

The SID Structure Sub-Sub-TLV is always included in routes with label fields and always uses the following values when advertised:

- Locator Block Length - encodes the length of the block configured in the locator for the service
- Locator Node Length - the length of the node configured in the locator for the service
- Function Length - configurable in the range 20 to 96
- Argument Length - 0 (default) or 16 (configured)
- Transposition Length (TL) - 20 for EVPN and VPN-IP routes with full SIDs and 16 for micro segments and 0 for IP routes in the base router
- Transposition Offset (TO)
  - for EVPN and VPN-IP routes:
    - If the Function Length equals 20 or 16, the Transposition Offset (TO) value equals the prefix length configured in the locator
    - If the Function Length is greater than 20, the TO value equals:

      (prefix length configured in the locator) + (FunctLength 20)
  - for IP routes in base router, TO value is always 0
- For IP routes in the base router, the TO value is always 0.

3.5.4 Transposition procedures when advertising service routes

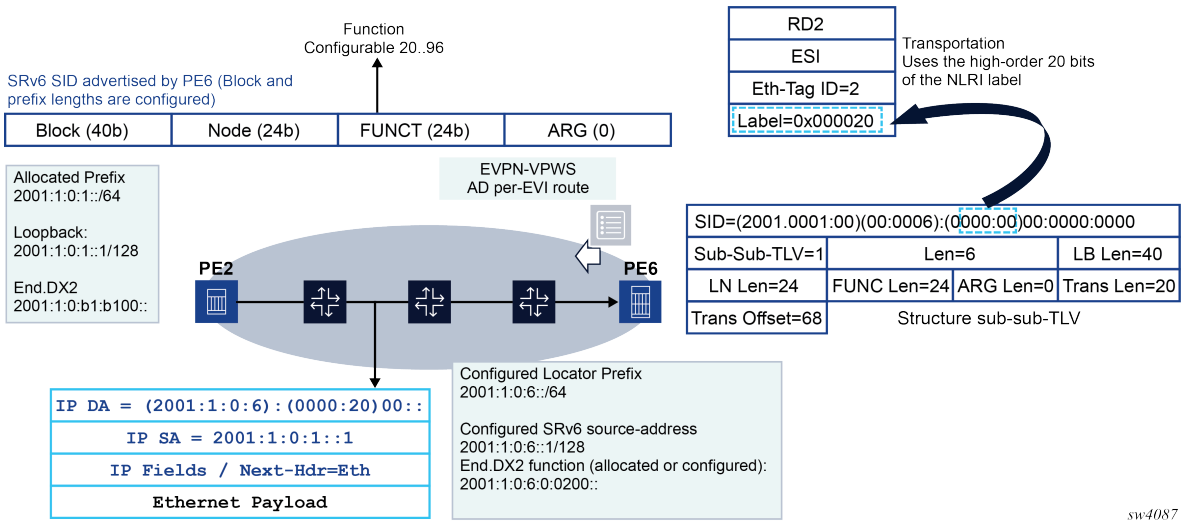
The purpose of the SID Structure Sub-Sub-TLV is twofold:

- Advertise the structure of the SRv6 SID used in the service, including the length of the Locator block, node, function and argument.

- Support transposition procedures for efficient service route packing. The FUNCTION is transposed into the label field in the route's NLRI. Because the rest of the SID is common for routes of the same type in the service, this transposition operation supports efficient packing of routes into the same BGP update.

The following figure shows how the FUNCTION part of the SID is transposed. This example illustrates how transposition works for EVPN-VPWS, and it would be similar for VPN-IP routes.

Figure 41: Transposition of the FUNCTION into the NLRI



In the preceding figure, PE6 is configured with an Epipe that uses a configured locator with LB length = 40 bits and LN length = 24 bits. The Function length is set at 24, and 20 bits are always transposed into the NLRI (non-configurable). In the preceding figure, the following rules apply:

- On reception, the router can build any SID out of the received route, irrespective of transposition, as long as the lengths are correctly encoded.
- On transmission, the system performs a transposition for VPN-IP and EVPN service routes as follows:
- – If Function Length is greater than 20 in the Locator configuration, the function bits are put at the right-most bits of the L bits. For example, if LB LEN is 40 bits and the LN Len is 24 bits:
  – If Function Length = 20, the entire function is transposed into the label field, and the following is signaled in the route:

  Length [LBL, LNL, FL, AL] : [40, 24, 20, 0]

  TL:20, TO:64

```
*A:PE-4>config>router>segment-routing>srv6>locator# info
----------------------------------------------
                        shutdown
                        block-length 40
                        termination-fpe 1
                        prefix
                            ip-prefix cafe:1:0:4::/64
                        exit
                        static-function
                            max-entries 10
                        exit
----------------------------------------------
*A:PE-4>config>router>segment-routing>srv6>locator# no shutdown
```

```
3 2021/01/19 08:21:16.827 UTC MINOR: DEBUG #2001 Base Peer 1: 2001:db8::3
"Peer 1: 2001:db8::3: UPDATE
Peer 1: 2001:db8::3 - Send BGP UPDATE:
    Withdrawn Length = 0
    Total Path Attr Length = 102
    Flag: 0x90 Type: 14 Len: 33 Multiprotocol Reachable NLRI:
        Address Family VPN_IPV4
        NextHop len 12 NextHop 192.0.2.4
        10.0.0.3/32 RD 192.0.2.4:20 Label 524254
    Flag: 0x40 Type: 1 Len: 1 Origin: 0
    Flag: 0x40 Type: 2 Len: 0 AS Path:
    Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
    Flag: 0xc0 Type: 16 Len: 8 Extended Community:
        target:64500:20
    Flag: 0xc0 Type: 40 Len: 37 Prefix-SID-attr:
       SRv6 Services TLV (37 bytes):-
         Type: SRV6 L3 Service TLV (5)
         Length: 34 bytes, Reserved: 0x0
         SRv6 Service Information
         Service Information sub-TLV Type 1
            Type: 1 Len: 30 Rsvd1: 0x0
            SRv6 SID: cafe:1:0:4::
            SID Flags: 0x0 Endpoint Behavior: 0x14 Rsvd2: 0x0
            SRv6 SID Sub-Sub-TLV
               Type: 1 Len: 6
               BL:40 NL:24 FL:20 AL0 TL:20 TO:64
```

– If Function Length = 32, part of the function is transposed into the label field, and the following is signaled in the route:

  Length [LBL, LNL, FL, AL] : [40, 24, 20, 0]

  TL:20, TO:76

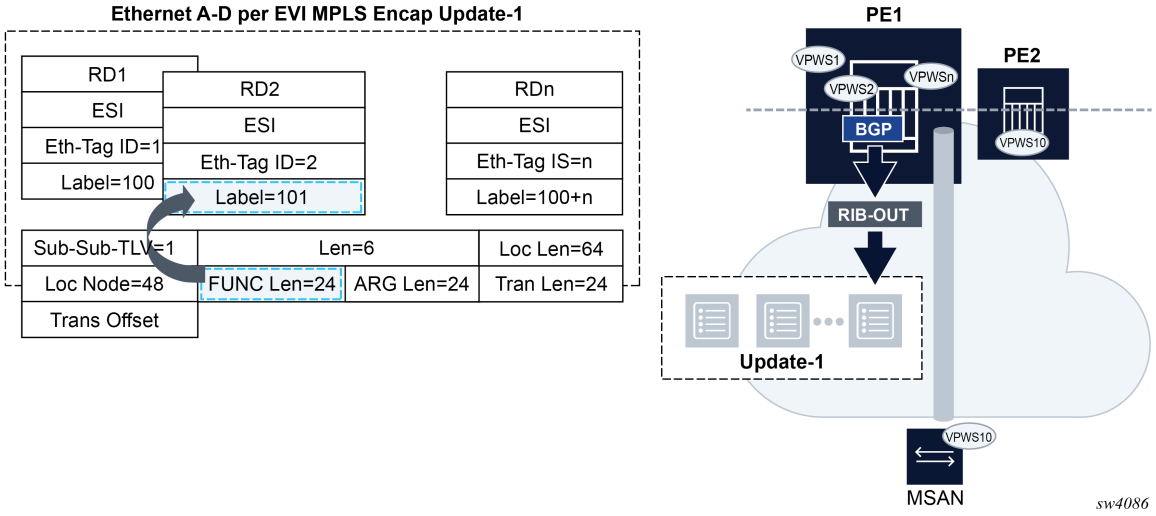
```
*A:PE-4>config>router>segment-routing>srv6>locator# function-length 32
*A:PE-4>config>router>segment-routing>srv6>locator# info
----------------------------------------------
                        shutdown
                        block-length 40
                        function-length 32
                        termination-fpe 1
                        prefix
                            ip-prefix cafe:1:0:4::/64
                        exit
                        static-function
                            max-entries 10
                        exit
----------------------------------------------
*A:PE-4>config>router>segment-routing>srv6>locator# no shutdown

8 2021/01/19 08:27:09.318 UTC MINOR: DEBUG #2001 Base Peer 1: 2001:db8::3
"Peer 1: 2001:db8::3: UPDATE
Peer 1: 2001:db8::3 - Send BGP UPDATE:
    Withdrawn Length = 0
    Total Path Attr Length = 102
    Flag: 0x90 Type: 14 Len: 33 Multiprotocol Reachable NLRI:
        Address Family VPN_IPV4
        NextHop len 12 NextHop 192.0.2.4
        10.0.0.3/32 RD 192.0.2.4:20 Label 524254
    Flag: 0x40 Type: 1 Len: 1 Origin: 0
    Flag: 0x40 Type: 2 Len: 0 AS Path:
    Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
    Flag: 0xc0 Type: 16 Len: 8 Extended Community:
```

```
        target:64500:20
    Flag: 0xc0 Type: 40 Len: 37 Prefix-SID-attr:
       SRv6 Services TLV (37 bytes):-
         Type: SRV6 L3 Service TLV (5)
         Length: 34 bytes, Reserved: 0x0
         SRv6 Service Information
         Service Information sub-TLV Type 1
            Type: 1 Len: 30 Rsvd1: 0x0
            SRv6 SID: cafe:1:0:4::
            SID Flags: 0x0 Endpoint Behavior: 0x14 Rsvd2: 0x0
            SRv6 SID Sub-Sub-TLV
               Type: 1 Len: 6
               BL:40 NL:24 FL:32 AL0 TL:20 TO:76
```

- The label field of the NLRI (VPN-IP and EVPN routes) encodes the FUNCTION that is dynamically or statically allocated for the service.

With the transposition procedure, multiple NLRIs with the same common SRv6 SID (minus the function) can be packed into the same BGP update, as is done for regular VPN-IP or EVPN route for MPLS tunnels. The packing benefit is illustrated in the following figure.

Figure 42: Transposition and route packing



Note: The transposition procedures do not apply to service SIDs in the base router, advertised via IPv4 and IPv6 families.

For micro-segment SRv6, the following applies:

- The Transposition Length (TL) is always 16 for EVPN and VPN-IP routes.
- For IP routes in the base router, the TL is always 0.
- Because FL is fixed to 16. the Transposition Offset (TO) = LBL + LNL.

3.5.5 Supported service routes for SRv6

The supported service routes for SRv6 are:

- VPRN services, configured for SRv6:
  - VPN-IPv4 routes
  - VPN-IPv6 routes
- Base router
  - IPv6 routes, if configured for SRv6 in IPv6 family
  - IPv4 routes, if configured for SRv6 in IPv4 family
- Epipe services, configured for SRv6:
  - EVPN AD per-EVI routes
  - EVPN AD per-ES routes, for Epipes that make use of Ethernet Segments
- VPLS services, configured for SRv6:
  - EVPN AD per-EVI routes
  - EVPN AD per-ES routes, including the advertisement of arguments
  - EVPN MAC/IP Advertisement routes
  - EVPN Inclusive Multicast Ethernet Tag routes

3.5.6 BGP next hop for SRv6 service routes

As specified in RFC 9252, *BGP Overlay Services Based on Segment Routing over IPv6 (SRv6)*, the egress PE may set the next hop to any of its IPv6 addresses. When the IPv6 address value is not covered by the SRv6 locator from which the SRv6 Service SID is allocated, the ingress PE performs reachability checks for the SRv6 Service SID in addition to the BGP next-hop reachability procedures.

Next hop and locator resolution considerations include the following:

- On reception of a BGP SRv6 service route, both the locator and the next hop are resolved independently in the route table.
- For base instance routes (not service routes), the following command triggers the independent resolution of the next hop and the locator reachability (and collates their states that drive route programming):
  - **MD-CLI**

    ```
    configure router bgp segment-routing-v6 family ignore-received-srv6-tlvs false
    ```

  - **classic CLI**

    ```
    configure router bgp segment-routing-v6 family no ignore-received-srv6-tlvs
    ```

  The locator state is considered only if the preceding command is configured.
- Whether the router does **next-hop-self** or **next-hop-unchanged**, there is no affect on the RIB-IN processing and reachability because the next-hop behavior is a RIB-OUT parameter.
- If a received route has a resolved next hop but unresolved locator, the **show router bgp routes** command shows "valid/best" but not "used" in the route flags. The following command displays the locator and the resolution of the locator.

  ```
  show router bgp next-hop 192.0.2.3 detail vpn-ipv4
  ```

Output example

```
===============================================================================
 BGP Router ID:192.0.2.4        AS:64500       Local AS:64500
===============================================================================

===============================================================================
BGP VPN Next Hop
===============================================================================
-------------------------------------------------------------------------------
VPN Next Hop          : 192.0.2.3
Autobind              : gre
Labels                : --
Admin-tag-policy      : --
Strict-tunnel-tagging : N
Color                 : --
Locator               : cafe:1:0:3::/64
-------------------------------------------------------------------------------
Resolving Prefix : 192.0.2.3/32
Preference       : 18                   Metric           : 10
Reference Count  : 6                    Owner            : GRE
Fib Programmed   : Y
Resolved Next Hop: 192.168.34.1
Egress Label     : n/a                  TunnelId         : 4294967293
Locator State    : Resolved
-------------------------------------------------------------------------------
Next Hops : 1
===============================================================================
```

The ingress PE can use the MT-ISIS Unreachable Prefix Announcement (UPA) feature for expedited BGP next-hop reachability checks when SRv6 route summarization is used toward the BGP next hops. Use the following command to activate UPA for IPv4 or IPv6 over SRv6.

```
configure router bgp segment-routing-v6 family upa-trigger next-hop frr
```

Use the following command to activate UPA for BGP-IPVPN.

```
configure service vprn bgp-ipvpn segment-routing-v6 upa-trigger next-hop frr
```

When a user applies the preceding commands, the ingress SRv6 BGP router can react with Fast Reroute (FRR) when it detects a matching UPA.

When BGP SRv6 UPA-based SRv6 FRR is activated, BGP starts listening to entries added to the UPA routing table to trigger BGP next-hop resolution when a longest prefix matching UPA is detected.

The BGP FRR processing sequence supports UPA-based next-hop unreachability checks. When BGP detects that a next hop disappears due to a matching UPA entry, BGP sends a trigger to the data plane to switch to the backup path, however:

- BGP does not withdraw the Network Layer Reachability Information (NLRI) from the BGP table
- BGP does not begin a new best-path calculation
- BGP does not send a withdraw request to its neighbors

Instead, BGP waits for the BGP control plane to converge as normal to detect through the traditional control plane that the NLRI is no longer valid. This BGP sequence results in BGP behaving with greater grace after UPA failures. This process allows UPA to serve as a stimulus for swift transitioning from the primary to the backup path, while the control plane remains unaffected. Consequently, BGP requires time to converge and withdraw the NLRI at the legacy speed. The effects of a UPA are contained, and this

subdues the reactive nature of BGP. This is advantageous, especially during periods of network instability and the resultant routing fluctuations.

3.5.7 Route policy support for matching and modifying BGP SRv6 service routes

Route matching criteria and route modifying actions that are specific to BGP routes carrying SRv6 TLVs can be configured for route policies.

SR OS supports the following route matching criteria:

- **SRv6 TLV presence as match criterion**

  Use the following command in the **from** clause of a policy-statement entry to configure whether a BGP route with associated SRv6 TLVs matches the entry:

  Note: Named entries are only supported in the MD-CLI.

  – **MD-CLI**

  ```
  configure policy-options policy-statement entry from srv6-tlv
  configure policy-options policy-statement named-entry from srv6-tlv
  ```

  – **classic CLI**

  ```
  configure router policy-options policy-statement entry from srv6-tlv
  ```

  This match criterion is supported in BGP import policies, BGP export policies, and VRF or VSI import policies.

- **SRv6 SID prefix as match criterion**

  Use the following command in the **from** clause of a policy-statement entry to configure a SID or micro-segment (uSID) value in the SRv6 TLV (including the locator and function) to be used as match criterion for a BGP route:

  – **MD-CLI**

  ```
  configure policy-options policy-statement entry from srv6-sid-prefix
  configure policy-options policy-statement named-entry from srv6-sid-prefix
  ```

  – **classic CLI**

  ```
  configure router policy-options policy-statement entry from srv6-sid-prefix
  ```

  This match criterion uses longest prefix match logic and is supported in BGP import policies, BGP export policies, and VRF or VSI import policies.

SR OS supports the following route modifying actions:

- **SRv6 locator modification**

  Use the following command in the **action** clause of a policy-statement entry to modify the SRv6 TLV in a BGP route to use a different locator than the default or the locator added using the command options in the **configure router bgp segment-routing-v6 family add-srv6-tlvs** context:

– **MD-CLI**

```
configure policy-options policy-statement entry action srv6-locator
configure policy-options policy-statement named-entry action srv6-locator
```

– **classic CLI**

```
configure router policy-options policy-statement entry action srv6-locator
```

This route modifying action is supported in BGP export policies and SRv6-specific VRF export policies.

- **SRv6 uSID locator modification**

  Use the following command in the **action** clause of a policy-statement entry to modify the SRv6 TLV in a BGP route to use a different uSID locator than the default or the uSID locator added using the command options in the **configure router bgp segment-routing-v6 family add-srv6-tlvs** context:

  – **MD-CLI**

  ```
  configure policy-options policy-statement entry action srv6-micro-segment-locator
  configure policy-options policy-statement named-entry action srv6-micro-segment-locator
  ```

  – **classic CLI**

  ```
  configure router policy-options policy-statement entry action srv6-micro-segment-locator
  ```

  This route modifying action is supported in BGP export policies and SRv6-specific VRF export policies.

3.6 Route table, FIB table, and tunnel table support

The following tables and information are needed to process a SRv6 packet at service origination, service termination, and transit router roles.

3.6.1 Route table and FIB

SRv6 locator and SID resolution is performed in the RTM and forwarding of all SRv6 packets is performed in the FIB.

The TTM is used to save details of the SRv6 tunnel but is not used directly to forward user or CPM originated packets.

The RTM and FIB are programmed with the routes of the local and remote locators, the local End.X SIDs, and the local End SIDs.

When a policy is applied to export SRv6 routes from the route table to another IS-IS instance, only the IP Reachability TLV and the locator TLV, along with the End SID sub-TLVs, are advertised by the receiving IS-IS instance. Local End, End.X, and LAN End.X routes are not exported nor advertised as separate routes.

- **remote locator (route owner = SRv6 IS-IS)**

  All routers in the SRv6 domain populate a resolved remote locator prefix received in the SRv6 Locator TLV in the route table and the FIB.

  A SRv6 packet is always forwarded out in the datapath using the FIB.

For algorithm 0, the same prefix is advertised with the IP reach prefix TLV and the SRv6 Locator TLV. A single route entry is programmed in the route table and the FIB.

The prefix of an IGP flexible algorithm locator TLV is never advertised with an IP reach prefix TLV. Therefore, the route of the locator TLV is programmed in the route table and the FIB.

– **remote locator with up to 64 ECMP next hops**

IS-IS models a remote locator prefix with two or more ECMP next hops as an IGP route with tunneled next hops using a protected NHLFE with hardware PG-ID per tunneled next hop.

This implementation provides uniform failover in ECMP. IS-IS allocates a hardware PG-ID to each next hop it establishes an adjacency with. That PG-ID is then used when programming SRv6 routes of a remote locator and of a local adjacency that resolve to this next hop.

The route table manager programs the route into the FIB. IS-IS creates an SRv6 tunnel for the locator prefix. The tunnel is added to the tunnel table. The IS-IS route entries in the route table and the FIB point to the tunnel ID of this tunnel in the tunnel table.

Weighted ECMP, when enabled on the interfaces of this IS-IS instance, is supported when forwarding packets over the locator next hops.

– **remote locator with primary or backup next hops**

To provide uniform failover, IS-IS models a remote locator prefix with a primary next hop or a primary and LFA backup next-hop pair as an IGP route with a tunneled next hop using a protected NHLFE with a hardware PG-ID.

The route table manager programs the route into the FIB. IS-IS creates a SRv6 tunnel for the locator prefix. The tunnel is added to the tunnel table. The IS-IS route entries in the route table and the FIB point to the tunnel ID of this tunnel in tunnel table.

- **local locator (route owner = SRv6)**

  All routers in the SRv6 domain populate a route entry in the route table and the FIB to terminate packets destined for the local locator. This is modeled like any other local route but with the SRv6-specific route owner.

- **local adjacency SID (route owner = IS-IS)**

  All routers in the SRv6 domain populate a route entry in the route table and FIB for each local End.X and LAN End.X adjacency SID with primary and backup next hops.

  The route table, and FIB entries are modeled like a remote locator prefix with primary and backup next hops.

- **local End SID (route owner = SRv6)**

  All routers in the SRv6 domain populate a route entry in the route table and the FIB to terminate packets destined for each local End SID. This is modeled like any other local route but with the SRv6-specific route owner.

With micro-segment SRv6, entries pertaining to local services populate the RTM and FIB. In the FIB, those entries are aggregated. At most, 15 entries are needed to cover the whole service addressing space offered by micro-segment SRv6. The route owner is SRv6.

3.6.2 TTM

The tunnel table is not used directly in the SRv6 locator resolution, in SID resolution, or in packet forwarding. All resolution is performed in the RTM and forwarding is performed in the FIB.

Each resolved remote locator or local adjacency creates an entry in TTM (ROUTE_OWNER_SRV6_ISIS). The entries for remote locators and local adjacencies in the RTM and FIB point to the tunnel ID of this tunnel. The TTM entry is not used directly for forwarding SRv6 packets. However, the route resolution behavior for SRv6 services can be modified to preferentially resolve in TTMv6. See SRv6 policy support for Layer 2 and Layer 3 services for more information.

3.6.3 Users of route table SRv6 routes

The SRv6 locator and adjacency routes in route table can be used to forward the following user and CPM originated packets:

- user packets
- ICMPv6 echo request and echo reply packets as described in *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide*
- UDP traceroute packets as described in *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide*

Forwarding and terminating of any other CPM originated packets are not supported. Specifically, received management protocol and control plane protocol packets that are encapsulated in SRv6 are dropped.

If the user configures the address of a BGP neighbor, an LDP peer, or an RSVP-TE LSP destination to match a locator prefix or a SID, packets are forwarded over the SRv6 tunnel but are dropped at the destination router.

3.7 Datapath support

This section describes packet processing in the datapath on ingress PE, egress PE, and transit P router roles.

SR OS supports both regular SRv6 and micro-segment SRv6. Both modes can operate concurrently on the same platform, but it requires that the SIDs of each type come from different SID blocks.

3.7.1 Service origination and termination roles

The SRv6 processing is performed inline in the IP datapath when forwarding service packets over a shortest path SRv6 tunnel at service origination or when terminating an SRv6 packet with the service SID in the DA field of the outer IPv6 header.

SRv6 processing is performed in a specialized SRv6 Forwarding Path Extension (FPE) when forwarding service packets over an SRv6 policy or when terminating an SRv6 packet with a local node SID in the DA field and the service SID in the SRH.

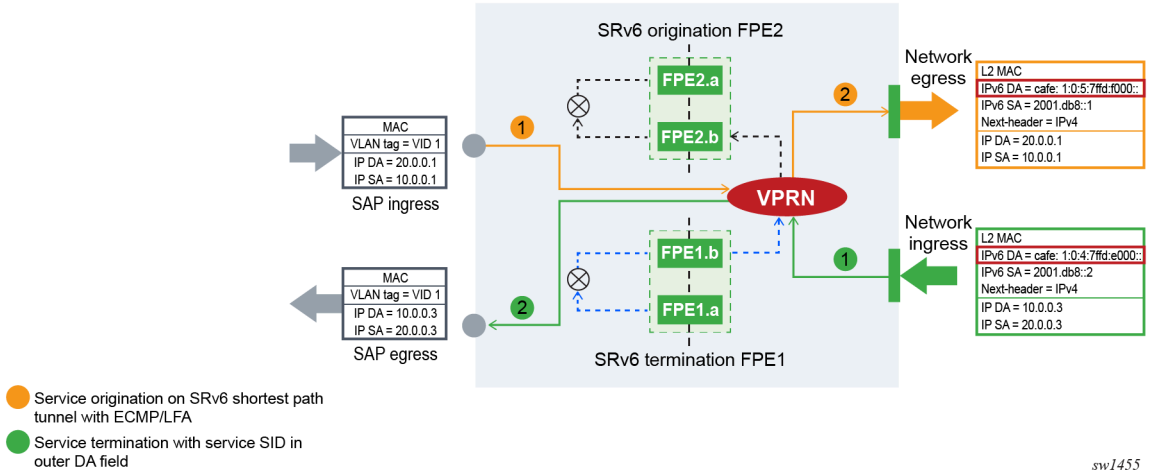
Note: The SRv6 FPE also processes service packets forwarded over an SRv6 shortest path tunnel when the service SID in the DA field matches a non-locator IPv6 route in the FIB. A non-locator IPv6 route is an IS-IS route of the locator prefix advertised without a locator TLV, a static route, an OSPF route, or a BGP route of the locator prefix.

The SRv6 origination and termination cannot share the same FPE. A single FPE can be configured for SRv6 origination. One or more FPEs can be configured for SRv6 termination.

Caution: Regardless of whether packets require SRv6 FPE processing, the user must configure the SRv6 origination FPE at the SRv6 ingress PE and one or more SRv6 termination FPEs at the SRv6 egress PE. On a per-packet basis, the datapath makes a determination if SRv6 FPE processing is required. Transit SRv6 routers do not need SRv6 FPEs except at an SRv6 policy BSID expansion node. In the latter, the user must configure an SRv6 origination FPE. See Segment routing policies with an IPv6 data plane for more information about the SRv6 policy feature.

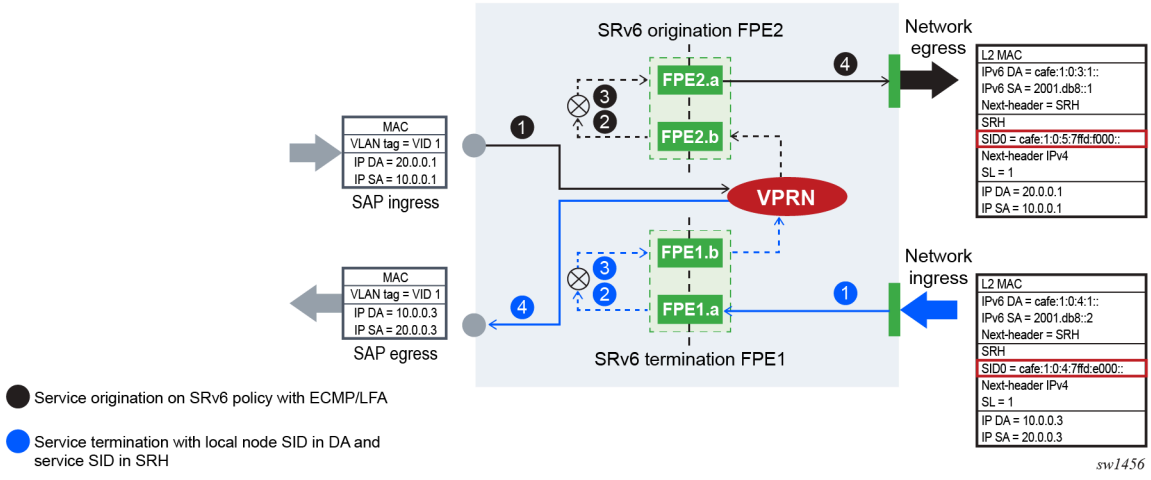
The following figure shows a packet walk-through including the service origination and termination without SRv6 FPE.

Figure 43: Packet walk-through showing service origination and termination without SRv6 FPE



The following figure shows a packet walk-through including the service origination and termination with SRv6 FPE.

Figure 44: Packet walk-through showing service origination and termination with SRv6 FPE



Micro-segment SRv6

The datapath behavior for micro-segment SRv6 slightly differs from regular SRv6. The general behavior of performing two lookups applies but the FIB entries are not the same.

After the parallel lookups, the same procedures as for regular SRv6 apply with the following exception in the behavior in step 1.b. The notion of matching only on a locator does not exist in micro-segment SRv6 because locators also realize an END/uN function. Therefore, a match on a locator is a match on a uN (shift or END). The packet is not sent to the FPE if the first match is on a local locator/uN, but rather if the first match is on a local locator/uN and the second match is on a service SID.

3.7.1.1 At the ingress PE

The following occurs at the ingress PE:

- **Packet forwarded to a shortest path SRv6 tunnel**

  The datapath pushes the SRv6 encapsulation header on the received Layer 2 or Layer 3 service packet.

  – The Outer DA field is set to service SID (End.DT4, End.DT6, End.DT46, or End.DX2 SID).

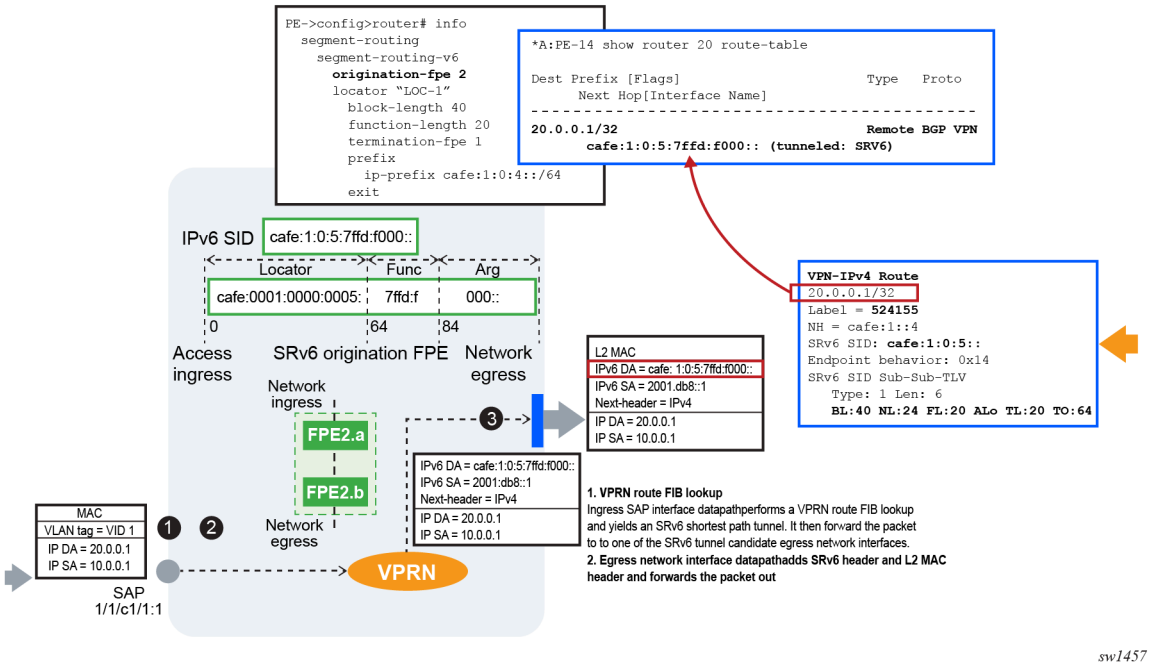
  – The Outer next-header field is set to IPv4, IPv6, or Ethernet.

  – The **hop-limit** field in the outer IPv6 header of the SRv6 tunnel is set to 255 for all transit IPv4, IPv6, and Ethernet packets encapsulated into SRv6. The hop-limit for OAM packets originated by the CPM on the router is set according to the specific OAM probe. See the *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide* for more information.

  The service ingress datapath forwards the packet to one of the SRv6 tunnel candidate egress network IP interfaces based on a hash of the inner packet headers. See Using flow label in load-balancing of IPv6 and SRv6 encapsulated packets for more information about the spraying of service packets over SRv6 tunnel candidate next hops.

  The following diagram displays details of the datapath processing of a service packet that is originated on a VPRN and forwarded over an SRv6 shortest path tunnel.

Figure 45: Walk-through SRv6 datapath without SRv6 FPE at service origination node



- **Packet forwarded to a SRv6 policy**

  The SRv6 FPE egress datapath receives the Layer 2 or Layer 3 service packet and pushes the SRv6 encapsulation header:

  – The Outer DA field is set to the top SID of the SRv6 policy.

  – The Outer next-header field is set to SRH.

  – The Service SID (End.DT4, End.DT6, End.DT46, or End.DX2 SID) is encoded in the SRH.

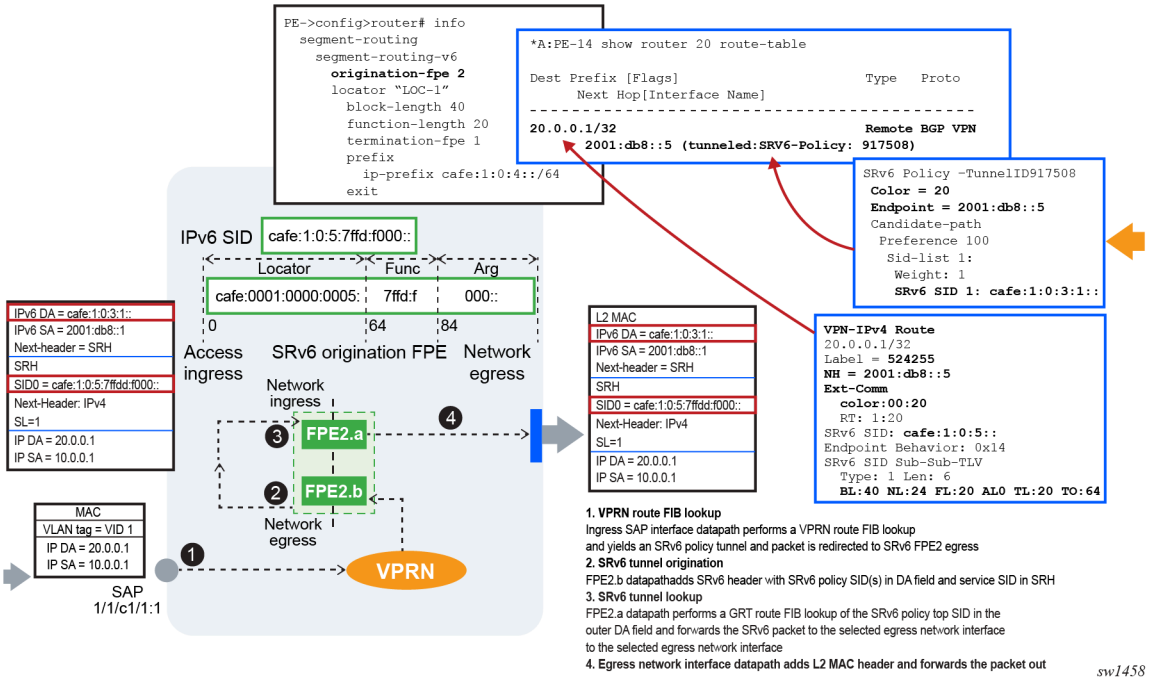
  – The SRH next-header field is set to IPv4, IPv6, or Ethernet.

  – The **hop-limit** field in the outer IPv6 header of the SRv6 tunnel is set to 255 for all transit IPv4, IPv6, and Ethernet packets encapsulated into SRv6. The hop limit for OAM packets originated by the CPM on the router is set according to the specific OAM probe. See the *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide* for more information.

  The SRv6 FPE ingress datapath does the lookup on the outer DA field and forwards the packet to one of the candidate egress network IP interfaces based on the flow label and the SA or DA or both fields of the outer IPv6 packet header. See Using flow label in load-balancing of IPv6 and SRv6 encapsulated packets for more information about the spraying of SRv6 packets.

  The following diagram displays details of the datapath processing of a service packet that is originated on a VPRN and forwarded over an SRv6 policy.

Figure 46: Walk-through SRv6 datapath with SRv6 FPE at service origination node



3.7.1.2 At the egress PE

1. The procedure in this step is common to the transit router role and the service termination router role.

   On the ingress IP network interface, the SRv6 feature concurrently performs two IPv6 address lookups on a received IPv6 packet:

   - A first (longest prefix match) lookup checks if the address in the outer header DA field matches either an SRv6 local locator subnet, a local service SID, a local End function or a local End.X function. This first lookup is for the current SID.
   - A second lookup is performed on the next SID in the SRH (when the IPv6 packet has an SRH). The SRv6 feature reads next SID using the index value after decrementing the Segments-Left field.

   The subsequent processing depends on the outcome of the first lookup:

   a. **If the match is on a local service SID:**

      - If the payload type is IPv4, IPv6, or Ethernet, the packet is forwarded to the service function processing; see step 2 for more details.

        The payload type refers to the value of the last next-header field in the processing chain of the packet. This could be the next-header field of the outer IPv6 packet, if there is no SRH. This could also be the next-header field of an expired SRH (**Segments-Left** = 0) for which the last SID matches the service SID.
      - If the payload type indicates any other protocol, including ICMPv6 (ICMP ping packet) and UDP (potential traceroute message when hop-limit field has a value of 1), the packet is redirected to the CPM for more processing. Protocol matching ICMPv6 ping and UDP traceroute have their

packets processed as described in the *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide*. Other protocol packets are dropped.

The CPM generates a specific ICMPv6 message to the address in the SA field of the processed or dropped packet depending on the protocol type and the result of the match of the address in the DA field of the packet. These ICMPv6 reply messages are summarized in Table 25: ICMPv6 reply messages to extracted SRv6 packets.

b. If the match is on a local locator only:

- If the payload type is IPv4, IPv6, or Ethernet, the packet is forwarded to the SRv6 FPE for potential service function processing; see step 2 for more details.
- If the payload type indicates any other protocol, including ICMPv6 (ICMP ping packet) and UDP (potential traceroute message when hop-limit field has a value of 1), the packet is redirected to the CPM for more processing. Protocol matching ICMPv6 ping and UDP traceroute have their packets processed as described in the *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide*. Other protocol packets are dropped.

  The CPM generates a specific ICMPv6 message to the address in the SA field of the processed or dropped packet depending on the protocol type and the result of the match of the address in the DA field of the packet. These ICMPv6 reply messages are summarized in Table 25: ICMPv6 reply messages to extracted SRv6 packets.

c. If the match is on a specific local End function and the next SID lookup is not a local locator, the packet is processed as per the transit router role for these functions as detailed in Transit router role with or without segment termination.

d. If the match is on a specific local End function and the next SID resulted in a match on a local locator, the packet is processed as described in step 1.b with the next-header field used in the processing is that of the SRH.

e. If the match is on a specific local End.X function, regardless of the next SID match outcome the packet is processed in accordance with the transit router role for these functions; see Transit router role with or without segment termination for more information.

f. If the match is on a regular IPv6 route or there is no match, the packet is forwarded or dropped. For forwarded packets, the destination address could match the locator prefix or a regular IPv6 prefix of a remote node.

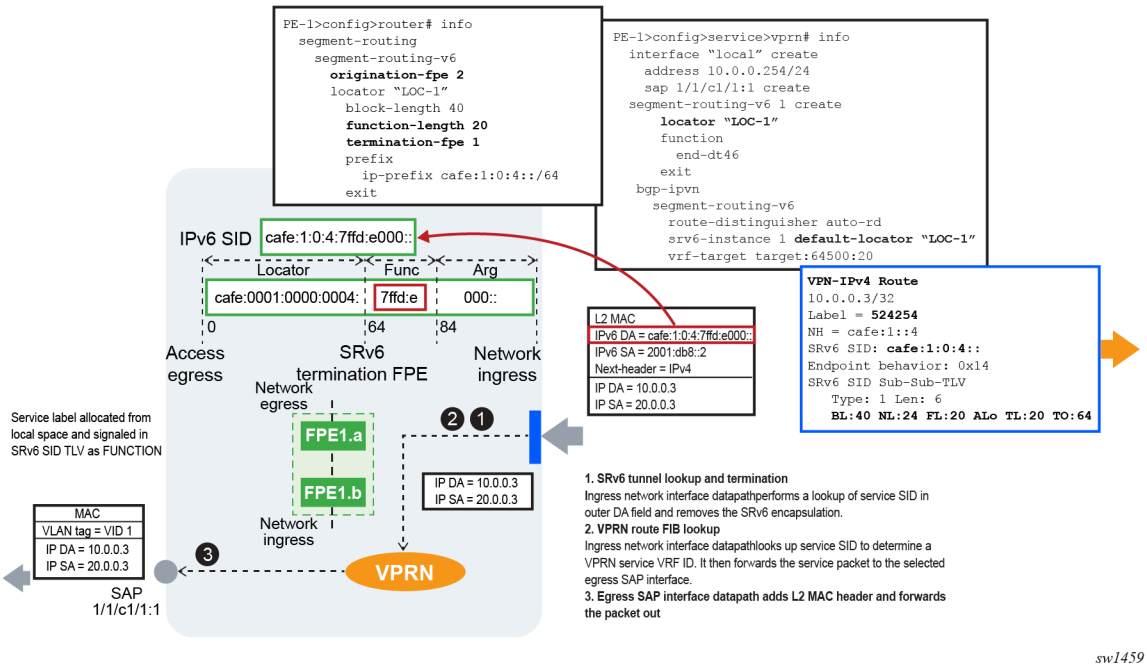
2. The procedure in this step is specific to the service termination router role.

a. When the first lookup match is on a local service SID (service SID encoded in outer DA field), service packet processing is performed inline in the network interface ingress datapath.

- The datapath performs the detailed processing of the specific SID function as per https://tools.ietf.org/html/rfc8986. It then removes the SRv6 encapsulation headers, including SRH if any.
- It decrements and propagates, into the IPv4 TTL field or IPv6 hop-limit field of the forwarded inner packet, the minimum of the incoming outer header hop-limit and inner header hop-limit (or TTL) values.
- It performs a lookup of the service SID and forwards the packet to the service context for further processing.

The following figure displays a walk-through of the SRv6 datapath without SRv6 FPE at the service termination node.

Figure 47: Walk-through SRv6 datapath without SRv6 FPE at service termination node

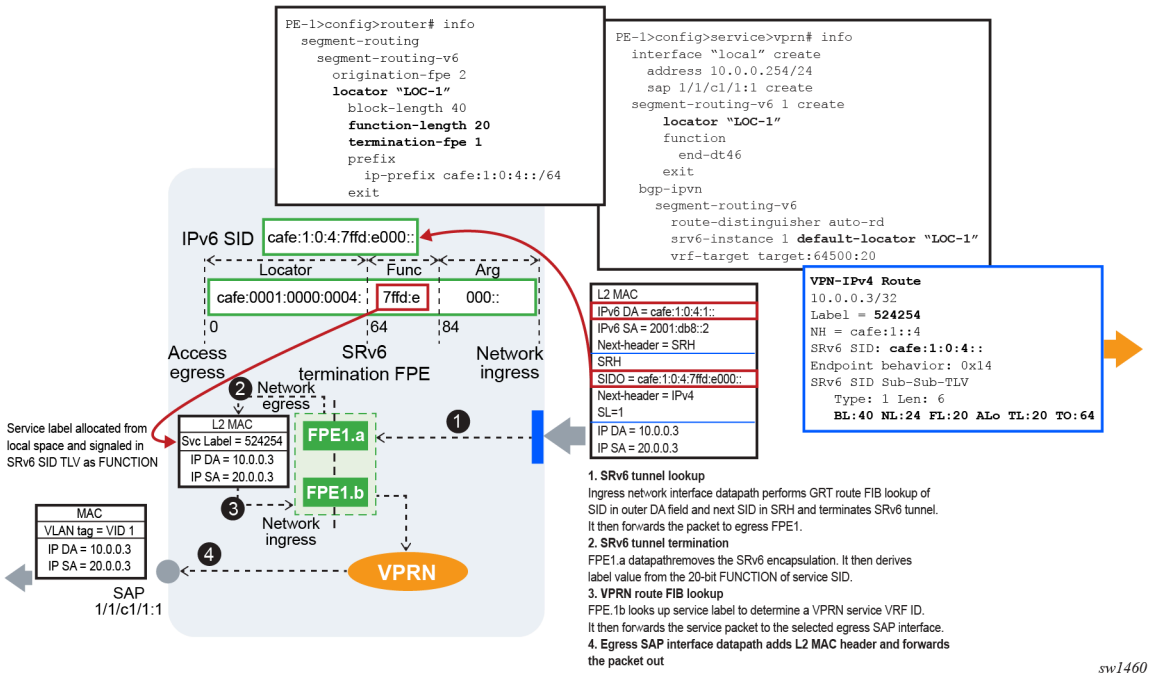


b. When the first lookup match is on a local locator entry of the FIB, or on a local End function and second lookup match is on a local locator, the service SID must be in the SRH and service packet processing is performed by the SRv6 termination FPE.

- The egress SRv6 FPE datapath receives the SRv6 encapsulated packet and performs the detailed processing of the specific SID function as per https://tools.ietf.org/html/rfc8986. It then removes the SRv6 encapsulation headers, including SRH if any, and inserts a service label, with the value derived from the FUNCTION field value, into the inner service packet.
  - The egress SRv6 FPE decrements and propagates, into the IPv4 TTL field or IPv6 hop-limit field of the forwarded packet, the minimum of the incoming outer header hop-limit and inner header hop-limit (or TTL) values.
  - The ingress SRv6 FPE does an ILM lookup on the service label and forwards the packet to the service context for further processing.

The following figure shows a walk-through of the SRv6 datapath with SRv6 FPE at the service termination node.

Figure 48: Walk-through SRv6 datapath with SRv6 FPE at service termination node



3.7.2 Transit router role with or without segment termination

The transit router role does not require the use of an SRv6 FPE except at SRv6 policy BSID expansion node. See Segment routing policies with an IPv6 data plane for more details on the SRv6 policy feature.

The following steps summarize the packet processing for the transit router role. For more information about the specific processing of the SID function see https://tools.ietf.org/html/rfc8986.

1. The procedure in this step is common to the transit router role and the service termination router role.

   On the ingress IP network interface, the SRv6 feature concurrently performs two IPv6 address lookups on a received IPv6 packet:

   - A first (longest prefix match) lookup checks if the address in the outer header DA field matches either an SRv6 local locator subnet, a local service SID, a local End function or a local End.X function. This first lookup is for the current SID.
   - A second lookup is performed on the next SID in the SRH (when the IPv6 packet has an SRH). The SRv6 feature reads next SID using the index value after decrementing the **Segments-Left** field.

   The subsequent processing depends on the outcome of the first lookup:

   a. If the match is on a local service SID:

      - If the payload type is IPv4, IPv6, or Ethernet, the packet is forwarded to the service function processing; see step 2 in section At the egress PE for more details.

        The payload type refers to the value of the last next-header field in the processing chain of the packet. This could be the next-header field of the outer IPv6 packet, if there is no SRH. This could also be the next-header field of an expired SRH (**Segments-Left** = 0) for which the last SID matches the service SID.

- If the payload type indicates any other protocol, including ICMPv6 (ICMP ping packet) and UDP (potential traceroute message when hop-limit field has a value of 1), the packet is redirected to the CPM for more processing. Protocol matching ICMPv6 ping and UDP traceroute have their packets processed as described in the *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide*. Other protocol packets are dropped.

  The CPM generates a specific ICMPv6 message to the address in the SA field of the processed or dropped packet depending on the protocol type and the result of the match of the address in the DA field of the packet. These ICMPv6 reply messages are summarized in the table that follows.

Table 25: ICMPv6 reply messages to extracted SRv6 packets

Protocol	Destination IP address match result	ICMPv6 reply (Type/code)
ICMP echo request/reply (See *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide* for ICMPv6 Ping support in SRv6)	locator prefix [function | any arg]	echo reply/ping successful
UDP / TCP (See *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide* for UDP Traceroute support in SRv6)	locator prefix [function | any arg]	dest unreachable, port unreachable
Any other protocol	locator prefix [function | any arg]	ICMP Parameter Problem/SR Upper-layer Header Error
All protocols including above	locator prefix | unsupported function	dest unreachable, communication prohibited

b. If the match is on a local locator only:

- If the payload type is IPv4, IPv6, or Ethernet, the packet is forwarded to the SRv6 FPE for potential service function processing; see step 2 in section At the egress PE for more information.
- If the payload type indicates any other protocol, including ICMPv6 (ICMP ping packet) and UDP (potential traceroute message when hop-limit field has a value of 1), the packet is redirected to the CPM for more processing. Protocol matching ICMPv6 ping and UDP traceroute have their packets processed as described in the *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide*. Other protocol packets are dropped.

  The CPM generates a specific ICMPv6 message to the address in the SA field of the processed or dropped packet depending on the protocol type and the result of the match of the address in the DA field of the packet. These ICMPv6 reply messages are summarized in Table 25: ICMPv6 reply messages to extracted SRv6 packets.

c. If the match is on a specific local End function and the next SID lookup is not a local locator, the packet is processed as per the transit router role for these functions as detailed in step 2 below.

d. If the match is on a specific local End function and the next SID resulted in a match on a local locator, the packet is processed as described in the preceding step 1.b with the next-header field used in the processing is that of the SRH.

e. If the match is on a specific local End.X function, regardless of the next SID match outcome, the packet is processed in accordance with the transit router role for these functions; see 2 below.

f. If the match is on a regular IPv6 route or there is no match, the packet is forwarded or dropped. For forwarded packets, the destination address could match the locator prefix or a regular IPv6 prefix of a remote node.

2. The procedure in this step is specific to the transit router role.

If the match is on a local End or End.X SID, the SID termination processing is performed on the packet.

a. If the End or End.X SID is the last SID in the packet encapsulation, meaning there is no SRH or there are only expired SRHs (**Segments-Left** = 0), the packet is sent to the CPM for further processing.

Note: The CPM processes ICMPv6 ping packets and UDP traceroute packets but drops any other protocol type. See the *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide* for more information about the processing of ICPMv6 echo request and reply packets and UDP traceroute packets.

b. If the next-header in the IPv6 header is an SRH, the **Segments-Left** field is zero, and the next-header in the SRH is another SRH, the current SRH is removed and the remaining steps are applied on the next SRH.

c. If the **Segments-Left** field is 1 and the SRH mode of the terminated SID is PSP, the SRH is removed. Otherwise, the **Segments-Left** field is decremented and used to read and copy the next SID into the DA field of the outer IPv6 header.

d. Decrement the incoming outer IPv6 header hop-limit and write it into the outer IPv6 header hop-limit field of the outgoing packet.

e. If the first SID lookup of the current SID in the FIB matched an End function, use the outcome of the second SID lookup of the next SID to forward the packet to the next hop of the next SID (in the DA field of the outer IPv6 header).

f. If the first SID lookup of the current SID in the FIB matched an End.X function, override the outcome of the second SID lookup of the next SID with the set of next hops of the adjacency and forward the packet.

g. If both the current and the next SIDs match a local End or End.X SID, the packet is forwarded as indicated in Table 26: Forwarding behavior for back-to-back local SIDs.

Table 26: Forwarding behavior for back-to-back local SIDs

Current SID match	Next SID match	Forwarding action
End	End	Packet is extracted to the CPM, which drops it if the next SID in not the last SID. An ICMPv6 packet (type: dest unreachable, code: communication prohibited) is sent to the address in the SA field

Current SID match	Next SID match	Forwarding action
		If the next SID is the last SID, the bounced packet is processed as per Table 25: ICMPv6 reply messages to extracted SRv6 packets.
End	End.X	Packet is forwarded over the adjacency of the next SID to the downstream neighbor, which forwards it back to this node for the next SID processing. The bounced packet is forwarded again over the same adjacency. If the next SID is the last SID, the bounced packet is processed as per Table 25: ICMPv6 reply messages to extracted SRv6 packets.
End.X	End	Packet is forwarded over the adjacency of the current SID to the downstream neighbor, which forwards it back to the current node for the next SID processing If the next SID is the last SID, the bounced packet is processed as per Table 25: ICMPv6 reply messages to extracted SRv6 packets.
End.X	End.X	Packet is forwarded over the adjacency of the current SID to the downstream neighbor, which forwards it back to the current node for the next SID processing If the next SID is the last SID, the packet is processed as per Table 25: ICMPv6 reply messages to extracted SRv6 packets.

3.7.3 Transit router role in micro-segment SRv6

The datapath behavior in transit SRv6-enabled routers distinguishes the following use cases:

- The DA field of the IPv6 header contains a single SID (shortest path case).
- Multiple SIDs are used in the DA field and eventually in the SRH (LFA repair tunnel or SRv6 policy).

The DA field of the IPv6 header contains a single SID is of the form `2001:db8:00d1:d547::`

where

- <2001:0db8> is the SID block
- <00d1> is the identifier associated with a specific node
- <d547> is an identifier for a specific local service assigned by the node

<2001:0db8><00d1> acts both as a locator for the specific node and as a uN function. A packet with this address in the DA field of the IPv6 header is routed toward the specific node with each node along the path matching on 2001:0db8:00d1::/48 and forwarding to the predetermined next hop.

Multiple SIDs in the DA field and eventually in the SRH illustrate the following micro-segment SRv6 behavior.

Suppose that a source node wants to send traffic for a specific service to the destination node D via the nodes B and F. For that example, consider the following:

- The micro-segment ID block is 2001:0db8::/32.
- The node identifiers (uN) are 0x00b1 for node B, 0x00f1 for node F, and 0x00d1 for node D.
- Node D has advertised 2001:0db8:00d1:d457:: for the specific service.

The source node does the following:

- It constructs the following container: 2001:0db8:00b1:00f1:00d1:d457::
- It places this container in the DA field of the IPv6 header.
- It forwards the packet to the next-hop determined by an LPM on that address.

On receiving the packet, nodes B and F perform a shift operation (bound to the uN).

Figure 49: Transit routers in micro-segment SRv6 illustrates the preceding example.

Node B matches on 2001:0db8:00b1::/48 (as it has that entry in its FIB) and performs the operation associated with that micro-segment ID:

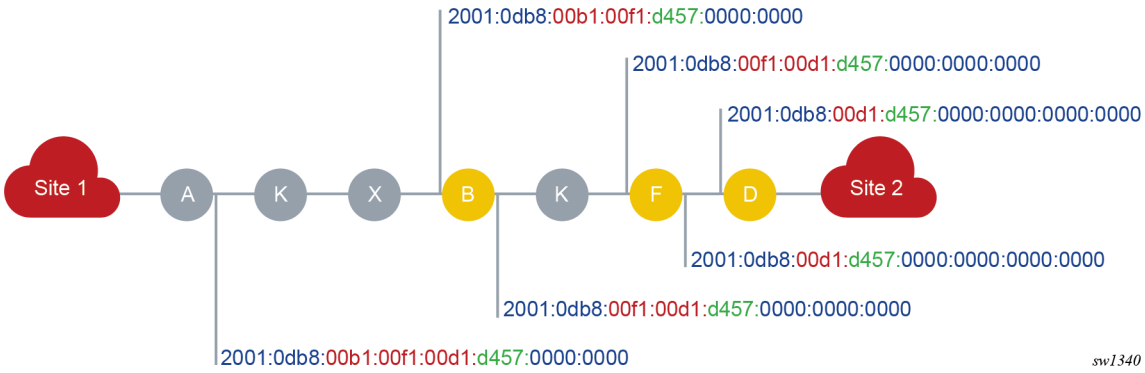
- It removes the 16-bit from the address.
- It shifts the right part toward the MSB.
- It adds a 16-bit block of zeros, the so called End of Container (EoC), as the LSB.

Node B forwards the packet to the next-hop based on a lookup of the resulting DA field.

Node F does the same (based on matching on its own identifier).

Node D does the same but processes the packet in the service corresponding to the identifier.

Figure 49: Transit routers in micro-segment SRv6



It can be that the path is longer than the number of micro-SIDs that can be compressed in 128 bits. In that case, an SRH is used to convey the rest of the path. Each container in the SRH must be of the same form (a block part followed by a sequence of micro-SIDs). At some node along the path, because of the shift operations, the container in the IPv6 DA expires. At that node, micro-segment SRv6 implements a regular SRv6 END function (or END.X function if the match is on a uA).

For example, if node B receives a packet with `2001:0db8:00b1::` in the DA field, it detects that its own identifier is followed by an EoC. Node B copies the next segment or container from the SRH in the DA field.

3.7.4 Using flow label in load-balancing of IPv6 and SRv6 encapsulated packets

When a service is bound to a SRv6 shortest path tunnel, the ingress service SAP or interface sprays the packets over the ECMP next hops of the SRv6 tunnel and LAG links of the outgoing network interfaces.

The default hash calculation on the ingress service SAP or interface is based on the existing hash procedures of an IPv4, IPv6, or an Ethernet packet. For IPv6 service packets, an option is provided to include the packet's Flow Label field, when not zero, and to hash on the triplet {SA, DA, Flow Label}. The **flow-label-load-balancing** command is used to enable this behavior on an access or network interface.

The ingress service SAP or interface copies the output of the hash on the inner packet headers into the flow label field of the outer IPv6 header that it pushes on the SRv6 encapsulated packet. This is regardless of whether the flow label is used or not in the computation of the hash on the service packet

When a service is bound to an SRv6 policy, the service packets are first forwarded to the egress network interface of the SRv6 origination FPE to build and push the SRv6 encapsulation. The packets are then handed in to the ingress network interface of the SRv6 origination FPE which sprays the packets over the ECMP next hops of the SRv6 policy and LAG links of the outgoing network interfaces.

The SRv6 origination FPE egress network interface copies the output of the hash on the inner packet headers into the flow label field of the outer IPv6 header that it pushes on the SRv6 encapsulated packet. This is regardless of whether the flow label is used or not in the computation of the hash on the service packet.

The SRv6 origination FPE ingress network interface does not require the **flow-label-load-balancing** command to be enabled. All SRv6 packets are automatically sprayed to the ECMP next hops of the SRv6 tunnel and LAG links of the outgoing network interfaces using a hash on the triplet {SA, DA, Flow Label} in the SRv6 packet's outer IPv6 header.

On a transit router, the hashing of SRv6 encapsulated packets can also use the Flow Label field in the outer IPv6 header to provide more entropy to the load-balancing process of SRv6 packets. The **flow-**

label-load-balancing command can be configured on a network interface to hash on the triplet {SA, DA, Flow Label}. By default, a transit router only hashes on the tuple {SA, DA} in the header of a received IPv6 packet with a non-zero flow label field, including when the packet is SRv6. The description of the **flow-label-load-balancing** command and the detailed behavior of the hash feature based on the IPv6 packet flow label field and its general application to access and network interfaces is described in section *IPv6 Flow Label Load Balancing* of the *7450 ESS, 7750 SR, 7950 XRS, and VSR Interface Configuration Guide*.

3.7.5 Interaction with other datapath features

The following describes the interaction of the SRv6 feature with other datapath features:

- When SRv6 is enabled on the base router, datapath enables forwarding and receiving SRv6 encapsulated packets on all network interfaces of the router. The datapath, however, drops an SRv6 encapsulated packet if received from or needs forwarding to an access interface (IES, VPRN). A service packet received on either a network interface or an access interface can be encapsulated into a SRv6 packet and forwarded to a network interface.
- SRv6 feature performs concurrently a couple of IPv6 address lookups on a packet received with an SRH. The first lookup is for the current SID in the DA field in the header of the received SRv6 packet and the second is for the next SID in the SRH.

  The datapath repurposes the IPv6 unicast RPF (IPv6 uRPF) check for the next SID lookup, which means the IPv6 uRPF feature cannot be performed on all IPv6 packets received on that interface. CLI enforces this interaction and so SRv6 cannot be enabled in the base router context (**config**>**router**>**segment-routing**>**segment-routing-v6**) if the IPv6 uPRF check (**config**>**router**>**if**>**ipv6**>**urpf**) is enabled on one or more network interfaces.

  Conversely, the IPv6 uRPF check cannot be enabled on a network interface if SRv6 is enabled in the base router context.

  Note: SRv6 does not impact uRPF checks of IPv4 packets received on a network interface.

- When SRv6 is enabled in one or more IGP instances, a transit router cannot check the SA field in the outer IPv6 header of a SRv6 encapsulated packet received on a network interface and that also has an SRH header. Normally an IPv6 packet which uses 0::0 or a link-local address format should be dropped. All other IPv6 packets, including a SRv6 encapsulated packet that does not have an SRH header, are checked for these two situations.
- Policy Based Routing (PBR) is allowed on flows of packets of an SRv6 tunnel. In other words, the user can apply an ACL filter on a network interface which matches on the outer SA and DA fields of the SRv6 packet and execute an action such as a redirection.

  The operator must ensure that SRv6 matching packets are directed to a router that can process and forward the SRv6 packets.

  Note: Redirecting an SRv6 packet even to an SRv6-capable router is not recommended because the processing of the SID list in the SRH can create loops for any of the SIDs in the outer DA and SRH.

3.8 LFA support

Note: This section and the following IS-IS procedures and Datapath procedures sections are applicable to regular and micro-segment SRv6, unless stated otherwise.

LFA, remote LFA, and TI-LFA are supported in the following router roles:

- service originating role
- transit role with segment termination
- transit role without segment termination

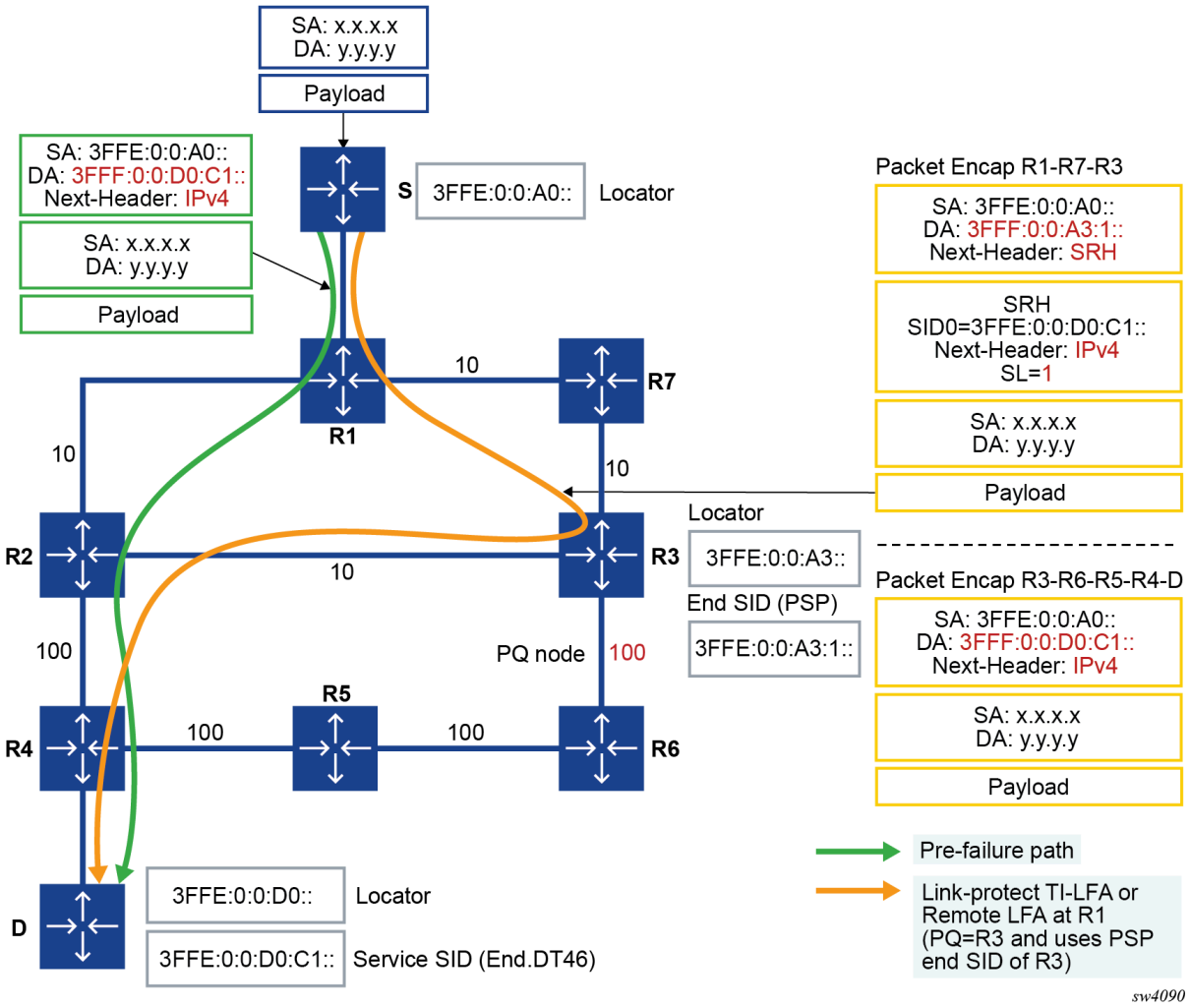
The backup is computed and programmed for each remote SRv6 node SID, service SID (for example, End.DT4, End.DT6, End.DT46, and End.DX2), and for each local adjacency or LAN adjacency SID.

A base LFA backup path or a TI-LFA backup path that uses a direct IP next hop (not a repair tunnel), requires configuring a next hop that is different from the primary path and does not modify the SID list pushed on the primary path.

When the RLFA or TI-LFA backup path uses a repair tunnel (source routed or not), the additional SIDs of the repair tunnel must be inserted into the packet when the backup is activated. This requires the insertion of an LFA dedicated SRH into the packet. The SRv6 behavior is referred to as H.Insert.Red and is described in *draft-filsfils-spring-srv6-net-pgm-insertion-04*. The application of this behavior to the LFA repair tunnel is described in *draft-voyer-6man-extension-header-insertion-10*.

The following figure illustrates the packet encoding for the primary and remote LFA backup path using the H.Insert.Red behavior and pushing a dedicated reduced SRH for the repair tunnel. The figure uses regular SIDs but is equally applicable to micro SIDs.

Figure 50: LFA repair tunnel packet encoding



The LFA backup path for the prefix of a remote node SID and of a remote service SID is programmed in the route table and tunnel table with the entry of the prefix of the locator of that SID. This practice is required because forwarding to these SIDs requires looking up the remote locator they were derived from.

The LFA backup for a local adjacency SID or a local LAN adjacency SID is programmed in route table with the specific entry corresponding to this SID.

3.8.1 IS-IS procedures

The base LFA, remote LFA, and TI-LFA features operate with SRv6 tunnels in the same way as with SR-MPLS tunnels.

Use the commands in the following context to enable the base LFA:

- **MD-CLI**

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

- **classic CLI**

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

Use the commands in the following contexts to enable remote LFA and TI-LFA:

- **MD-CLI**

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

- **classic CLI**

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

The **max-srv6-frr-sids** command option configures the maximum number of SRv6 SIDs allowed in the segment list of a TI-LFA repair tunnel. A higher **max-srv6-frr-sids** value results in better coverage by TI-LFA, at the expense of increased packet encapsulation overhead. The SR OS implementation can insert up to 1 additional regular SID or up to 3 additional micro SIDs in the segment list of a TI-LFA repair tunnel.

The **max-srv6-frr-sids** parameter is used by TI-LFA to limit the search for the P-Q set in the post-convergence path. Users can configure the following values:

- **0**

  The IGP LFA SPF restricts the search to TI-LFA backup next hop, which does not require a repair tunnel. This means that the P and Q nodes are the same and match a neighbor of the LFA SPF computing router.

- **1 (default)**

  This option corresponds to a repair tunnel to the node SID of a PQ node (P=Q) or to an adjacency SID between adjacent P and Q nodes. With regular 128-bit SIDs, an SRv6 adjacency is globally routable, therefore there is no need to insert the node SID of the P node; the adjacency SID between the P and Q nodes is sufficient.

  With micro-segment SRv6, an adjacency SID is globally routable only if it is combined with the corresponding node SID (<uN,uA>). In that case and in all other cases, the combination counts as a single SID toward the limit configured by **max-srv6-frr-sids**.

- **2 to 3**

  When protecting a node SID, the IGP LFA SPF widens the search to include a repair tunnel to a P-Q set in which P and Q nodes are interconnected with two to three adjacencies. IGP can also protect an adjacency using the backup path of the neighbor’s node SID. In this case, the neighbor’s node SID is added to the SIDs of the adjacencies between the P and Q nodes, for a total of two to three SIDs.

  With micro-segment SRv6, the addition of the neighbor's node SID can exceed the **max-srv6-frr-sids** value as long as it fits in the same LFA SRH container.

The following is a description of the SRv6 IS-IS tunnel backup path computation. A compression of the SIDs is applied to minimize the SID list of the computed backup path.

The backup path of a remote locator is as follows:

- **TI-LFA**

  IS-IS adds the segment list of node and adjacency SIDs of the P-Q set to the repair tunnel. The datapath encodes the top SID of this segment list into the DA field of the outer IPv6 header and moves the received SID value in the DA field and the remaining SIDs of the repair tunnel segment list into the

LFA SRH. The protected locator prefix backup next hop in the route table points to the tunnel ID of the locator prefix of the top SID of the P-Q set segment list.

- **base LFA**

  If an alternate equal-cost parallel link to the LFA neighbor exists, IS-IS programs it as a regular IP next hop of the protected locator.

  If the alternate parallel link to the LFA neighbor is not equal-cost, IS-IS programs a null SID repair tunnel. The datapath does not push an LFA SRH in this case. The protected locator prefix backup next hop in the route table points to the tunnel ID of the local adjacency over the outgoing link to the LFA neighbor.

- **remote LFA**

  IS-IS adds the node SID of the PQ node to the repair tunnel. The datapath encodes this SID into the DA field of the outer IPv6 header and moves the received SID value of the DA field into the LFA SRH. The protected locator prefix backup next hop in the route table points to the tunnel ID of the locator prefix of the PQ node.

Note: If the P node is a neighbor of the computing node, the SID list is empty and the backup next hop of the protected locator prefix in the route table points to the tunnel ID of the local adjacency over the outgoing link to the LFA neighbor.

The backup path of a local adjacency is as follows:

- **TI-LFA**

  IS-IS computes the repair tunnel by using the segment list of the P-Q set plus a node SID of the neighbor node on the other side of the protected adjacency. If the Q node is the same as the neighbor node on the other side of the protected adjacency, the repair tunnel is programmed after compressing the SID list to the adjacency SIDs of the links between the P node and the neighbor node. The datapath encodes the top SID of this segment list into the DA field of the outer IPv6 header and moves the received SID value in the DA field and the remaining SIDs of the repair tunnel segment list into the LFA SRH.

- **base LFA**

  IS-IS attempts first to program a null SID repair tunnel by using the adjacency of an alternate parallel link to the neighbor on the other side of the protected adjacency. The datapath does not push an LFA SRH in this case. The protected locator prefix backup next hop in the route table points to the tunnel ID of the local adjacency over the outgoing link to the LFA neighbor.

  If the first option fails, IS-IS programs a repair tunnel by using the node SID of the neighbor on the other side of the protected adjacency. The datapath encodes that node SID of the neighbor into the DA field of the outer IPv6 header and moves the received value of the DA field into the LFA SRH.

- **remote LFA**

  IS-IS computes the repair tunnel by using the node SID of the PQ node plus a node SID of the neighbor node on the other side of the protected adjacency. If the PQ node is one hop from the node on the other side of the protected adjacency, the repair tunnel SID list is compressed to the adjacency SID of the link between the PQ node and the neighbor node. The datapath encodes the top SID of this segment list into the DA field of the outer IPv6 header and moves the received SID value in the DA field and the remaining SIDs of the repair tunnel segment list into the LFA SRH.

Note: IS-IS prefers a PSP over a USP SID when selecting the PQ node SID or the P-Q set adjacency SID of a remote LFA or TI-LFA repair tunnel. In general, if a third-party implementation

signals other SRH modes, IS-IS selects the SID in the ascending order of the SRH mode codepoint for the SID. For example, node SIDs in the regular SRv6 have the following order:

1. End 0x01
2. End_PSP 0x02
3. End_USP 0x03
4. End_PSP_USP 0x04
5. End_PSP_USD 0x1D
6. End_USP_USD 0x1E
7. End_PSP_USP_USD 0x1F

When two or more node SIDs of the same SRH mode exist, IS-IS selects the SID with the lowest function value.

When two or more adjacency SIDs of the same SRH mode exist, IS-IS prefers the SID with protection enabled and selects the SID with lowest function value from the protected or unprotected SID subset.

The TI-LFA algorithm prefers a micro-segment locator over a regular locator in case both exist. As such, a micro-segment path may protect a regular SRv6 path.

3.8.2 Datapath procedures

The datapath procedures for the service origination router role, as described in Service origination and termination roles, are modified as follows when the LFA backup is a repair tunnel using the H.Insert.Red encapsulation.

1. The top SID of the primary path is copied into the LFA SRH.
2. The top regular SID or the whole LFA SRH micro-segment container of the repair tunnel is copied in the DA field of the outer IPv6 header. The Segments-Left field of the LFA SRH is set to 1.
3. If the P-Q set consists of a PQ node only and the PQ node is the same as the node that owns the top SID of the primary path, the LFA SRH insertion is skipped. This is not a datapath procedure per se, but IGP compresses the SID list of the backup path to look the same as that of the primary path.

The datapath procedures for the transit router role, as described in Transit router role with or without segment termination, are modified as follows when the LFA backup is a repair tunnel using the H.Insert.Red encapsulation.

1. The next SID, read from the original SRH after decrementing Segments-Left field, or from the DA field if no SRH exists, is copied into the LFA SRH.
2. The top regular SID or the whole LFA SRH micro-segment container of the repair tunnel is copied in the DA field of the outer IPv6 header. The Segments-Left field of the LFA SRH is set to 1.
3. If the P-Q set consists of a PQ node and the PQ node is the same as the node that owns the next SID, the LFA SRH insertion is skipped. This is not a datapath procedure per se, but IGP compresses the SID list of the backup path to look the same as that of the primary path.

3.9 SRv6 tunnel metric and MTU settings

IGP sets the metric of the SRv6 remote locator prefix route and tunnel or that of a local adjacency SID route to the metric of the computed path of the corresponding route.

The metric of a local End SID route is set to 0, similar to any local route.

The metric of a BGP IPv4/IPv6 or VPN-IPv4/VPN-IPv6 route resolved to a SRv6 tunnel inherits the value of the locator prefix route metric.

The user must configure the network interfaces at the ingress PE and at transit P routers with an MTU value that accounts for the fixed IPv6 header (40 bytes) and the additional LFA SRHs (24 bytes each).

Special attention is required when a service packet is forwarded over the SRv6 origination FPE interface (interface-b) at ingress PE. The datapath accounts for the fixed 40-byte IPv6 header in the check for fragmenting IPv4 packets (DF=0) or for dropping IPv4 (DF=1) and IPv6 packets that are intended to be forwarded over an SRv6 tunnel. If LFA is enabled in IS-IS, the LFA overhead is not accounted for, and therefore the configured MTU for the SRv6 origination FPE interface must account for it.

In addition, there are network deployments where it is not possible to modify the network interface MTU or to set all network interfaces to the same value. In that case, the user must configure the outgoing network interface and the SRv6 origination FPE interface MTU to reflect this worst case MTU in the network, accounting fixed and variable LFA overhead. The following is the CLI to use for this purpose.

```
configure
+--fwd-path-ext
   +--fpe <fpe-id> [create]
         +--srv6 <origination | termination>
            +--interface-b
                +--mtu <1280-9786>
```

The **show router base icmp6** command has a global count for received “Packets Too Big” that captures the dropped packets on network interfaces, including SRv6 origination FPE interface, caused by MTU violation.

3.9.1 MTU configuration examples

The following are examples of MTU settings of the SRv6 origination FPE interface (interface-b):

- **all default values**

  By default, LFA is disabled in IS-IS. The network interface default MTU is 9786 bytes based on default Ethernet MTU of 9800 bytes.

  The SRv6 origination FPE interface MTU defaults to the same value of 9786 bytes; there is no need for further adjustments.

- **all default values and RLFA or TI-LFA enabled**

  The user must adjust the SRv6 origination FPE interface MTU from the first bullet item case to account for the 24 bytes introduced by LFA.

  SRv6 origination FPE Interface MTU is 9786 - 24 = 9762 bytes.

- **a remote constrained network interface and RLFA or TI-LFA enabled**

Assume a specific remote network interface in the SRv6 network is limited to a maximum value of 1500 bytes. The user must adjust the SRv6 origination FPE interface MTU in all the ingress PEs from the first bullet item case, to account for that constrained value, plus the 24 bytes introduced by LFA repair tunnel (Users can adjust for as many LFA SRHs as they need. The following example assumes 1 LFA SRH).

SRv6 origination FPE Interface MTU = 1500 - 24 = 1476 bytes.

3.10 Service extensions

This section describes the service extensions to originate and terminate SRv6 services.

The extensions also apply to micro-segment SRv6. To distinguish a regular locator from a micro-segment locator, specific CLI contexts matching those for regular SRv6 are available.

3.10.1 SRv6 forwarding path extension

SRv6 processing is performed in a specialized SRv6 FPE when forwarding service packets over a SRv6 policy or when terminating a SRv6 packet with a local node SID in the DA field and the service SID in the SRH. The SRv6 FPE also processes service packets forwarded over a SRv6 shortest path tunnel when the service SID in the DA field matches a non-locator IPv6 route in the FIB. A non-locator IPv6 route is an ISIS route of the locator prefix advertised without a locator TLV, a static route, an OSPF route, or a BGP route of the locator prefix.

The following guidelines apply for the SRv6 FPE:

- An internal or external Port Cross-Connect (PXC) can be used for the SRv6 FPE.
- The SRv6 origination or termination FPE cannot be shared with other applications, but the same physical ports can be used when configuring PXC ports for multiple FPEs of different applications.

  As an example of how two FPEs can share the same physical port (therefore the same bandwidth), define two PXC ports, both sharing the same underlaying physical port; for instance:

  FPE 1 is associated with PXC-1 and FPE 2 is associated with PXC-2, where PXC-1 and PXC-2 are both assigned to port 1/1/1
- Some considerations about the SRv6 termination FPE follow:
  - It is configured per locator.
  - Multiple locators can optionally use the same or different FPE.
  - Received SRv6 traffic for a specific (local) locator is redirected to the SRv6 termination FPE interface-a.
- Some considerations about the SRv6 origination FPE follow:
  - There is only one SRv6 origination FPE supported per system.
  - The SRv6 origination and termination FPEs are always different.
- SRv6 FPE redundancy and load-balancing:
  - Each FPE can use a LAG composed of as many PXC ports as needed (there is no specific limitation in the number of PXC members per LAG).
  - LAG members can be PXC ports in the same or a different card.

The following CLI is required to create the FPE of type **srv6** origination or termination and apply it to a locator. All locators may be associated with the same or a different FPE.

```
configure
+--fwd-path-ext
   +--fpe <fpe-id>
      +--application
         +--srv6 <origination|termination>
            +--interface-a
               +--qos <network-policy-id>
            +--interface-b
               +--mtu <1280-9786>
               +--qos <network-policy-id>
```

```
configure
+--router
|    +--segment-routing
|    |    +--segment-routing-v6
|    |    |    +--origination-fpe <fpe>
|    |    |    +--source-address <ipv6-address>
|    |    |    +--locator <locator-name>
|    |    |    |    +--termination-fpe <fpe>
```

3.10.2 SRv6 VPRN services

VPRN services support SRv6 End.DT4, End.DT6, and End.DT46 behaviors. VPRNs support IPv4 and IPv6 routes that are advertised in VPN-IPv4 and VPN-IPv6 families, as well as in EVPN IP prefix routes in the Interface-less mode.

The following classic CLI commands configure a VPRN for SRv6 with the VPN-IP families.

```
configure
+--service
|   +---vprn <service-id>
|   |   +---segment-routing-v6 <instance-id>
|   |   |   +---locator <locator-name>
|   |   |   |   +---function
|   |   |   |   |   +---end-dt4 <integer>
|   |   |   |   |   +---end-dt6 <integer>
|   |   |   |   |   +---end-dt46 <integer>
|   |   |   +---bgp-ipvpn
|   |   |   |   +---segment-routing-v6 <bgp-instance-id>
|   |   |   |   |   +---srv6-instance <id> default-locator <name>
|   |   |   |   |   +---source-address <ipv6-address>
|   |   |   |   |   +---route-distinguisher <rd>
|   |   |   |   |   +---vrf-export
|   |   |   |   |   +---vrf-import
|   |   |   |   |   +---vrf-target
|   |   |   |   |   +---default-route-tag <number>
|   |   |   |   |   +---shutdown
```

The following MD-CLI commands configure a VPRN for SRv6 with the VPN-IP families.

```
configure
+--service
|   +---vprn <service-id>
|   |   +-- segment-routing-v6 <number>
|   |   |   +-- apply-groups <reference>
```

```
|   |   |   +-- apply-groups-exclude <reference>
|   |   |   +-- locator <reference>
|   |   |       +-- apply-groups <reference>
|   |   |       +-- apply-groups-exclude <reference>
|   |   |       +-- function
|   |   |           +-- end-dt4
|   |   |           |   +-- value <number>
|   |   |           +-- end-dt46
|   |   |           |   +-- value <number>
|   |   |           +-- end-dt6
|   |   |           |   +-- value <number>
|   |   +---bgp-ipvpn
|   |   |   +-- segment-routing-v6 <number>
|   |   |       +-- admin-state <keyword>
|   |   |       +-- apply-groups <reference>
|   |   |       +-- apply-groups-exclude <reference>
|   |   |       +-- default-route-tag <number>
|   |   |       +-- domain-id <string>
|   |   |       +-- evi <number>
|   |   |       +-- route-distinguisher <string | keyword>
|   |   |       +-- source-address <global-unicast-ipv6-address>
|   |   |       +-- srv6
|   |   |       |   +-- default-locator <reference>
|   |   |       |   +-- instance <reference>
|   |   |       +-- vrf-export
|   |   |       |   +-- apply-groups <reference>
|   |   |       |   +-- apply-groups-exclude <reference>
|   |   |       |   +-- policy <string | reference>
|   |   |       +-- vrf-import
|   |   |       |   +-- apply-groups <reference>
|   |   |       |   +-- apply-groups-exclude <reference>
|   |   |       |   +-- policy <string | reference>
|   |   |       +-- vrf-target
|   |   |           +-- community <string>
|   |   |           +-- export-community <string>
|   |   |           +-- import-community <string>
```

The following classic CLI commands configure a VPRN for SRv6 with EVPN-IFL.

```
configure
+--service
|   +---vprn <service-id>
|   |   +-- segment-routing-v6 <number>
|   |   |   +-- apply-groups <reference>
|   |   |   +-- apply-groups-exclude <reference>
|   |   |   +-- locator <reference>
|   |   |       +-- apply-groups <reference>
|   |   |       +-- apply-groups-exclude <reference>
|   |   |       +-- function
|   |   |           +-- end-dt4
|   |   |           |   +-- value <number>
|   |   |           +-- end-dt46
|   |   |           |   +-- value <number>
|   |   |           +-- end-dt6
|   |   |           |   +-- value <number>
|   |   +---bgp-evpn
|   |   |   +---segment-routing-v6 <bgp-instance-id>
|   |   |   |   +---srv6-instance <id> default-locator <name>
|   |   |   |   +---source-address <ipv6-address>
|   |   |   |   +---route-distinguisher <rd>
|   |   |   |   +---vrf-export
|   |   |   |   +---vrf-import
|   |   |   |   +---vrf-target
```

```
|   |   |   |   +---default-route-tag <number>
|   |   |   |   +---shutdown
```

The following MD-CLI commands configure a VPRN for SRv6 with EVPN-IFL.

```
configure
+--service
|   +---vprn <service-id>
|   |   +-- segment-routing-v6 <number>
|   |   |   +-- apply-groups <reference>
|   |   |   +-- apply-groups-exclude <reference>
|   |   |   +-- locator <reference>
|   |   |       +-- apply-groups <reference>
|   |   |       +-- apply-groups-exclude <reference>
|   |   |       +-- function
|   |   |           +-- end-dt4
|   |   |           |   +-- value <number>
|   |   |           +-- end-dt46
|   |   |           |   +-- value <number>
|   |   |           +-- end-dt6
|   |   |           |   +-- value <number>
|   |   +---bgp-evpn
|   |   |   +-- segment-routing-v6 <number>
|   |   |       +-- admin-state <keyword>
|   |   |       +-- apply-groups <reference>
|   |   |       +-- apply-groups-exclude <reference>
|   |   |       +-- default-route-tag <number>
|   |   |       +-- domain-id <string>
|   |   |       +-- evi <number>
|   |   |       +-- resolution <keyword>
|   |   |       +-- route-distinguisher <string | keyword>
|   |   |       +-- source-address <global-unicast-ipv6-address>
|   |   |       +-- srv6
|   |   |       |   +-- default-locator <reference>
|   |   |       |   +-- instance <reference>
|   |   |       +-- vrf-export
|   |   |       |   +-- apply-groups <reference>
|   |   |       |   +-- apply-groups-exclude <reference>
|   |   |       |   +-- policy <string | reference>
|   |   |       +-- vrf-import
|   |   |       |   +-- apply-groups <reference>
|   |   |       |   +-- apply-groups-exclude <reference>
|   |   |       |   +-- policy <string | reference>
|   |   |       +-- vrf-target
|   |   |           +-- community <string>
|   |   |           +-- export-community <string>
|   |   |           +-- import-community <string>
```

The associated locator must be configured to enable SRv6 on the VPRN service. In addition, the following rules apply:

- The function value can either be statically configured or dynamically allocated.
- Any Layer 3 function behavior can be configured, although VPN-IPv4 and EVPN-IFL IPv4 routes are advertised with **end-dt4** or **end-dt46** in that preference order (if they exist) and VPN-IPv6 and EVPN-IFL IPv6 routes are advertised with **end-dt6** or **end-dt46** in that preference order.
- The VPRN or label mode is not relevant to SRv6 and setting it has no effect on the behavior of the SRv6 feature.
- The following are supported:

  - BGP-IPVPN and BGP-EVPN (EVPN-IFL) families are simultaneously supported in the same VPRN where SRv6 is enabled.
  - Up to two BGP instances per VPRN are supported.
  - The two BGP instances can be associated with the same family or different families.
  - A family can have two BGP instances of SRv6 encapsulation.
  - In addition, two BGP-IPVPN instances can be configured with SRv6 and MPLS encapsulations respectively. Two BGP-EVPN instances can also be configured with SRv6 and MPLS encapsulations, respectively.
- The following applies to VPRN feature interaction with SRv6:
  - The following commands under the VPRN context only operate on MPLS encapsulations:
    - **class-forwarding**
    - **entropy-label**
    - **hash-label**
    - **label-mode**
    - **ttl-propagate**
  - The following commands under the VPRN context are mutually exclusive with SRv6:
    - **carrier-carrier-vpn**
    - **network-interface**
    - **export-inactive-bgp**
  - The following VPRN commands and features operate on MPLS and SRv6 encapsulations:
    - **vprn-type**
    - **allow-export-bgp-vpn**
    - **ecmp-unequal-cost**
    - **bgp-vpn-backup**
    - Unicast protocols on PE-CE interfaces
    - Commands such as **ipsec**, **nat**, and **subscriber-interfaces** are supported (no interaction).
    - ECMP and edge PIC are supported for SRv6 and also across routes of the same family with different encapsulations; for example, the same prefix is resolved to SRv6 and MPLS tunnels.
    - Route target-based leaking is supported for SRv6 routes in the VPRN.

3.10.3 SRv6 VPRN and BGP path attribute propagation between route table BGP owners

As is the case with existing MPLS encapsulation, BGP Path Attribute Propagation between BGP owners of the same VPRN route table is supported, including routes with SRv6 encapsulation.

- BGP Path Attribute propagation for SRv6 routes does not require enabling a CLI knob
- In cases where multiple BGP owners coexist in the same VPRN route table with a single instance, propagation is supported in the following cases, irrespective of the encapsulation of the route (MPLS or SRv6):
  - VPN-IPv4/6 ← → EVPN-IFL

  - VPN-IPv4/6 ← → VPN-IPv4/6 – when **allow-export-bgp-vpn** is enabled
  - EVPN-IFL← → EVPN-IFL – when **allow-export-bgp-vpn** is enabled
  - VPN-IPv4/6 ← → IPv4/v6
  - EVPN-IFL ← → IPv4/v6
  - VPN-IPv4/6 ← → EVPN-IFF (requires **iff-attribute-uniform-propagation**)
  - EVPN-IFL ← → EVPN-IFF (requires **iff-attribute-uniform-propagation**)
- In the case of multi-instance bgp VPRNs, VPN-IPv4/6 and EVPN-IFL routes received in one instance are readvertised into the other instance, including all the BGP Path Attributes of the original route.
- The following attributes are filtered out before the path attributes are propagated in all the previously described cases:
  - all type 0x06 extended communities
  - the BGP encapsulation extended community
  - the BGP encapsulation attribute
  - the BGP Prefix-SID attribute (which includes SRv6 Service SID TLVs)
  - all Route Target extended communities

3.10.4 Migration from MPLS to SRv6 in VPRN services

To allow a seamless migration from MPLS tunnels to SRv6 in VPRN services, both MPLS auto-bind tunnels and SRv6 are supported in the same VPRN service.

Routes for the same prefix follow regular route table selection in the VPRN. At the end of the selection process, if there are still SRv6 and MPLS routes, the SRv6 route is selected first and the MPLS route is removed from consideration.

3.10.5 SRv6 service SIDs and BGP routes in the base router

In the base router BGP instance, BGP routes may be originated, propagated or received with SRv6 TLVs in the prefix SID attribute. The processing rules are expected to use the following logic:

- A VPN-IPv4 or VPN-IPv6 route that is imported by the base router BGP instance from a VPRN (through the VRF export process), and that already has a prefix SID attribute with an SRv6 TLV at this stage, is advertised to all VPN-IPv4 and VPN-IPv6 peers of the base router BGP instance, with the **next-hop-self** as normal. There is no option to strip the SRv6 TLV or prefix SID attribute toward any of these peers, however, there is an option to drop the entire route toward selected peers or groups. This is similar to the effect of a BGP export policy that drops the route, but route policies cannot match routes based on the presence of an SRv6 TLV.
- An IPv6 route that is imported by the base router BGP instance from another protocol's route that was added to base router route table (static, OSPF, IS-IS, and so on) does not have a prefix SID attribute carrying the SRv6 TLV added to it by default. Adding a prefix SID can only be achieved by configuring the **config**>**router**>**bgp**>**segment-routing-v6**>**family ipv6 add-srv6-tlvs** command; however, this command also has the effect of adding a prefix SID attribute with SRv6 TLV to BGP IPv6 routes received from other peers without the SRv6 TLV and that are propagated to other family IPv6 peers with **next-hop-self** applied.

- Any BGP route received with a prefix SID attribute carrying SRv6 TLVs that is not an imported VPN-IPv4, VPN-IPv6, or EVPN route or an IPv6 unlabeled unicast route is treated the same as an existing route, that is, by ignoring the prefix SID attribute contents, resolving the route based only on its BGP next hop, and propagating the prefix SID attribute to other peers unchanged.
- If an IPv6 unlabeled unicast route is received with a prefix SID attribute carrying an SRv6 TLV and the **config**>**router**>**bgp**>**segment-routing-v6**>**family ipv6 ignore-received-srv6-tlvs** command is configured, that route is treated the same as an existing route, that is, by ignoring the prefix SID attribute contents, resolving the route based only on its BGP next hop, and propagating the attribute to other peers unchanged, irrespective of the **next-hop-self**.
- If an IPv6 unlabeled unicast route is received with a prefix SID attribute carrying an SRv6 TLV and the **config**>**router**>**bgp**>**segment-routing-v6**>**family ipv6 ignore-received-srv6-tlvs** command is set to FALSE, that route is considered resolved only if its BGP next hop is reachable and the locator prefix is reachable. The datapath or FNH programming and IGP cost to reach the next hop (used by the BGP decision process) is based on the route to the locator prefix. The IPv6 route is propagated to other family IPv6 peers with the prefix SID attribute and its SRv6 TLV unchanged, irrespective of the **next-hop-self**.
- An IPv6 unlabeled unicast route that is received without a prefix SID attribute containing an SRv6 TLV does not have a prefix-SID attribute carrying the SRv6 TLV added to it by default, even if it is re-advertised with the **next-hop-self** applied. This can only be achieved by configuring the **config**>**router**>**bgp**>**segment-routing-v6**>**family ipv6 add-srv6-tlvs**; however, this command also has the effect of adding a prefix SID attribute with SRv6 TLV to imported or redistributed routes.

```
configure
+--router
   +--segment-routing
|   |   +--- base-routing-instance
|   |   |   |   locator <locator-name>
|   |   |   |   +---function
|   |   |   |   |   +---end-dt6 <integer>
```

```
configure
+--router
   +--bgp
      +--segment-routing-v6
         +--source-address <ipv6-address>
         +--family*
            +--add-srv6-tlvs
            |  +--locator <locator-name>
            +--ignore-received-srv6-tlvs
```

Note:

- When an SRv6 TLV is added to an IPv6 unlabeled unicast route, the signaled behavior is End.DT6 and the SID Structure Sub-Sub-TLV contains a TO and TL of 0.
- As in the case of VPRNs, a locator is configured for the base router. If the End.DT6 function for that locator is not statically configured, a dynamically-allocated function is reserved for the End.DT6 behavior. This is internally mapped to a label, as in the case of VPRNs.

3.10.6 SRv6 Epipe services

Epipe services support SRv6 End.DX2 behavior. Currently, the SRv6 SID for End.DX2 is signaled by EVPN-VPWS AD per-EVI routes.

Use the following CLI commands to configure an Epipe service for SRv6:

```
+--epipe
   +--segment-routing-v6 <instance-id>
   |  +--locator <locator-name>
   |  |  +--function
   |  |  |  +--end-dx2 <integer>
   |  +--bgp-evpn
   |  |  +-- segment-routing-v6 <number>
   |  |  |   +-- admin-state <keyword>
   |  |  |   +-- default-route-tag <number>
   |  |  |   +-- ecmp <number>
   |  |  |   +-- force-vc-forwarding <keyword>
   |  |  |   +-- oper-group <reference>
   |  |  |   +-- route-next-hop
   |  |  |   |   +-- ip-address <ip-address>
   |  |  |   |   +-- system-ipv4
   |  |  |   |   +-- system-ipv6
   |  |  |   +-- source-address <global-unicast-ipv6-address>
   |  |  |   +-- srv6
   |  |  |       +-- default-locator <reference>
   |  |  |       +-- instance <reference>
```

Where the following conditions apply:

- The SRv6 Epipe is configured with the locator that is used. This determines the SID structure and value that is advertised in the AD per EVI route for the service.
- A single SRv6 instance with a single locator is supported on Epipes.
- The SRv6 Epipe uses an End.DX2 function value that, if not configured, is dynamically allocated by the system, out of the dynamic range available for the locator. If the **end-dx2** function is configured, then this value is used instead of a dynamic value.
- The following is supported in SRv6 Epipes:

  ecmp

  used for aliasing on remote SRv6 EVPN ES destinations

  force-vlan-vc-forwarding

  used to preserve one VLAN tag in the payload of the SRv6 tunnel. On termination, the VLAN tag is transparently passed through the termination SRv6 FPE.

  default-route-tag

  used to allow easy matching of the service routes on export policies

  oper-group

  required for fault-propagation and fate-sharing with the monitoring objects

  route-next-hop

  required to control the advertised BGP next hop for the EVPN AD per-EVI route. When EVPN Multihoming is used, this value must match the ES **originating-ip** and the ES **route-next-hop** value.

EVPN-VPWS is the control plane technology to signal SRv6 Epipes. The **local-attachment-circuit** Ethernet-tag value is advertised in an AD per-EVI route that may have a zero or non-zero ESI (if multihoming is used). The AD per-EVI routes are advertised along with the Layer-2 SRv6 Services TLV encoding with an End.DX2 behavior and using the transposition procedures described in Transposition procedures when advertising service routes. Upon reception the ingress PE creates an EVPN destination, as long as the received route includes the remote expected Ethernet-tag and route-target. The following CLI excerpt shows a configuration example and the created EVPN SRv6 destination on the PE:

```
*A:PE-3# configure service epipe 200
*A:PE-3>config>service>epipe# info
----------------------------------------------
            segment-routing-v6 1 create
                locator "LOC-1"
                    function
                        end-dx2 200
                    exit
                exit
            exit
            bgp
            exit
            bgp-evpn
                local-attachment-circuit ac-23 create
                    eth-tag 23
                exit
                remote-attachment-circuit ac-5 create
                    eth-tag 5
                exit
                evi 200
                segment-routing-v6 bgp 1 srv6-instance 1 default-locator "LOC-1" create
                    source-address 2001:db8::3
                    ecmp 2
                    route-next-hop system-ipv6
                    no shutdown
                exit
            exit
            sap lag-1:200 create
                no shutdown
            exit
            no shutdown
----------------------------------------------
*A:PE-3>config>service>epipe# /show service id 200 segment-routing-v6 destinations

===============================================================================
TEP, SID
===============================================================================
Instance  TEP Address                       Segment Id
-------------------------------------------------------------------------------
1         2001:db8::5                       cafe:1:0:5:c:8000::
-------------------------------------------------------------------------------
Number of TEP, SID: 1
-------------------------------------------------------------------------------
===============================================================================

===============================================================================
Segment Routing v6 Ethernet Segment Dest
===============================================================================
Instance  Eth SegId                      Num. Macs     Last Change
-------------------------------------------------------------------------------
No Matching Entries
===============================================================================
```

SRv6 Epipes support EVPN Multihoming. Their associated Ethernet Segments (ES) can be shared among MPLS services and other SRv6 services.

EVPN Multihoming for SRv6 Epipes follows these guidelines:

- The ES used by SRv6 Epipes is only supported in **evi-rt** mode, therefore separate AD per-ES routes per service are advertised for SRv6 Epipes.
- If **evi-rt-set** is configured in the ES, the SRv6 services are skipped when packing route-targets for the AD per-ES route. This is the same behavior as having a Epipe services with a configured **vsi-export** policy; the system also skips the route-targets for Epipes with **vsi-export** when packing multiple route-targets in the same AD per-ES route.
- The advertised AD per-ES route for the SRv6 Epipe includes a Layer-2 SRv6 Services TLV with a SID value of zero. The used behavior code point is End.DT2M. In addition, the AD per-ES route is also advertised with an ESI Label extended community that encodes the implicit-null value in the ESI label field. A debug example of a received AD per-ES route for SRv6 follows:

```
22 2021/05/21 18:23:10.247 UTC MINOR: DEBUG #2001 Base Peer 1: 2001:db8::2
"Peer 1: 2001:db8::2: UPDATE
Peer 1: 2001:db8::2 - Received BGP UPDATE:
    Withdrawn Length = 0
    Total Path Attr Length = 125
    Flag: 0x90 Type: 14 Len: 48 Multiprotocol Reachable NLRI:
        Address Family EVPN
        NextHop len 16 Global NextHop 2001:db8::2
        Type: EVPN-AD Len: 25 RD: 192.0.2.2:200 ESI: 01:d8:47:ff:00:00:00:00:01:
00, tag: MAX-ET Label: 0
    Flag: 0x40 Type: 1 Len: 1 Origin: 0
    Flag: 0x40 Type: 2 Len: 0 AS Path:
    Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
    Flag: 0xc0 Type: 16 Len: 16 Extended Community:
        target:64500:200
        esi-label:3/Single-Active
    Flag: 0xc0 Type: 40 Len: 37 Prefix-SID-attr:
       SRv6 Services TLV (37 bytes):-
         Type: SRV6 L2 Service TLV (6)
         Length: 34 bytes, Reserved: 0x0
         SRv6 Service Information
         Service Information Sub-TLV Type 1
            Type: 1 Len: 30 Rsvd1: 0x0
            SRv6 SID: ::
            SID Flags: 0x0 Endpoint Behavior: 0x18 Rsvd2: 0x0
            SRv6 SID Sub-Sub-TLV
               Type: 1 Len: 6
               BL:0 NL:0 FL:0 AL0 TL:0 TO:0
```

The creation of ES destinations follows the same rules as in EVPN-VPWS services with MPLS transport, except that the ES destination is resolved to remote SRv6 SIDs, as shown in the following example:

```
*A:PE-5# show service id 200 segment-routing-v6 destinations

===============================================================================
TEP, SID
===============================================================================
Instance  TEP Address                       Segment Id
-------------------------------------------------------------------------------
No Matching Entries
===============================================================================

===============================================================================
Segment Routing v6 Ethernet Segment Dest
```

```
===============================================================================
Instance  Eth SegId                       Num. Macs     Last Change
-------------------------------------------------------------------------------
1         01:d8:47:ff:00:00:00:00:01:00   0             06/18/2021 17:04:15
-------------------------------------------------------------------------------
Number of entries: 1
-------------------------------------------------------------------------------
===============================================================================
*A:PE-5# show service id 200 segment-routing-v6 esi 01:d8:47:ff:00:00:00:00:01:00

===============================================================================
Segment Routing v6 Ethernet Segment Dest
===============================================================================
Instance  Eth SegId                       Num. Macs     Last Change
-------------------------------------------------------------------------------
1         01:d8:47:ff:00:00:00:00:01:00   0             06/18/2021 17:04:15
-------------------------------------------------------------------------------
Number of entries: 1
-------------------------------------------------------------------------------
===============================================================================

===============================================================================
Segment Routing v6 Dest TEP Info
===============================================================================
Instance  TEP Address                   Segment Id          Last Change
-------------------------------------------------------------------------------
1         2001:db8::2                   cafe:1:0:2:c:8000:: 06/18/2021 17:04:15
-------------------------------------------------------------------------------
Number of entries : 1
-------------------------------------------------------------------------------
===============================================================================
```

3.10.7 SRv6 VPLS services

VPLS services support SRv6 End.DT2M behavior for BUM traffic and End.DT2U behavior for known unicast traffic. The SRv6 SID for End.DT2M is signaled by EVPN Inclusive Multicast Ethernet Tag (IMET) routes, whereas the SID for End.DT2U is signaled along with EVPN MAC/IP routes.

Use the following contexts to configure a VPLS service for SRv6:

- **MD-CLI**

```
configure service vpls "1" segment-routing <instance-id>
configure service vpls "1" bgp-evpn segment-routing-v6 <number>
```

- **classic CLI**

```
configure service vpls "1" segment-routing-v6 <instance-id>
configure service vpls "1" bgp-evpn segment-routing-v6 srv6-instance 1 default-locator
```

The following conditions apply for a VPLS service for SRv6:

- The SRv6 VPLS is configured using the locator. This determines the SID structure and value that is advertised along with the EVPN IMET and MAC/IP routes for the service.
- A single SRv6 instance with a single locator is supported on VPLS services.
- SRv6 VPLS services use End.DT2M and End.DT2U SIDs. Their corresponding function values, if not configured, are allocated automatically by the router. Dynamic or static, both function values (End.DT2M and End.DT2U) are needed so the EVPN routes for the services are advertised. While both values

are different locally on the service, the router accepts the same value for End.DT2M and End.DT2U functions from a remote PE for the same service.

- If the following command is configured on the PEs attached to the same service, the service MTU value is advertised in the EVPN Layer 2 Attributes extended community along with the IMET routes.
  - **MD-CLI**

    ```
    configure service vpls bgp-evpn routes incl-mcast advertise-l2-attributes
    ```

  - **classic CLI**

    ```
    configure service vpls bgp-evpn incl-mcast-l2-attributes-advertisement
    ```

  Upon receiving the signaled MTU from an egress PE, the ingress PE compares the received and local MTU. In case of a mismatch, the EVPN SRv6 destination goes operationally down, and the operational flag MTU-Mismatch shows the reason. Use the following command to configure the router to ignore the MTU signaled by the remote PE and bring up the SRv6 destination if there are no other reasons to keep it down.

  ```
  configure service vpls bgp-evpn ignore-mtu-mismatch
  ```

- The following commands are supported in SRv6 VPLS services:
  - **force-vlan-vc-forwarding**

    This command preserves one VLAN tag in the payload of the SRv6 tunnel.
  - **default-route-tag**

    This command matches BGP EVPN routes for the service on export policies, typically to add or modify BGP attributes.
  - **oper-group**

    This command is required for fault-propagation purposes.
  - **protected-src-mac-violation-action discard**

    This command is required for loop avoidance, along with **mac-duplication blackhole**.
  - **split-horizon-group**

    This command is used for seamless integration with spoke SDPs and migration from EVPN-MPLS services (in multi-instance services).
  - **route-next-hop**

    This command is used to control the BGP next-hop used for service routes. If not configured, it is **system-ipv4** by default.
  - **source-address**

    If not configured, it is inherited from the locator’s source address. The source address does not need to be reachable or even exist on a local interface. This is possible because the source address is not looked up in the datapath at the remote PE.
  - **resolution {route-table | tunnel-table | fallback-tunnel-to-route-table}**

    This command is used to set the resolution of SRv6 routes in the route table or tunnel table (needed for SRv6 policy), and even a fallback from tunnel to route-table resolution.

Example: Configuration example and the allocated functions on the PE

```
A:PE-2# configure service vpls 1900
A:PE-2>config>service>vpls# info
            segment-routing-v6 1 create
                locator “LOC-1"
                    function
                        end-dt2u
                        end-dt2m
                    exit
                exit
            exit
            bgp
            exit
            bgp-evpn
                evi 1900
                segment-routing-v6 bgp 1 srv6-instance 1 default-locator "LOC-1" create
                    source-address 2001:db8::2
                    route-next-hop 2001:db8::2
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            no shutdown
```

```
show service id 1900 segment-routing-v6 instance 1
```

Output example

```
===============================================================================
Segment Routing v6 Instance 1 Service 1900
===============================================================================
Locator
  Type         Function  SID                                          Status
-------------------------------------------------------------------------------
LOC-1
  End.DT2U       *504273 cafe:1:0:2:7b1d:1000::                       ok
  End.DT2M       *504272 cafe:1:0:2:7b1d::                            ok
===============================================================================
```

```
show router segment-routing-v6 local-sid end-dt2u end-dt2m
```

Output example

```
===============================================================================
Segment Routing v6 Local SIDs
===============================================================================
SID                                             Type           Function
  Locator
  Context
-------------------------------------------------------------------------------
cafe:1:0:2:7b1d::                               End.DT2M       504272
  LOC-1
  SvcId: 1900 Name: bd-1900-srv6
cafe:1:0:2:7b1d:1000::                          End.DT2U       504273
  LOC-1
  SvcId: 1900 Name: bd-1900-srv6
```

```
-------------------------------------------------------------------------------
SIDs : 2
-------------------------------------------------------------------------------
===============================================================================
```

Example: Remote PEs

If the remote PEs attached to the same VPLS services are configured in a similar way, the EVPN destinations for BUM and unicast traffic are created and can be displayed as follows:

```
show service id 1900 segment-routing-v6 destinations
```

Output example

```
===============================================================================
TEP, SID (Instance 1)
===============================================================================
TEP Address                   Segment Id                      Oper  Mcast Num
                                                              State       MACs
-------------------------------------------------------------------------------
2001:db8::3                   cafe:1:0:3:7b1d:7000::          Up    BUM   1
2001:db8::4                   cafe:1:0:4:7:b1db::             Up    BUM   0
2001:db8::5                   cafe:1:0:5:7b1d:d000::          Up    BUM   0
2001:db8::5                   cafe:1:0:5:7b1d:e000::          Up    None  1
-------------------------------------------------------------------------------
Number of TEP, SID: 4
-------------------------------------------------------------------------------
===============================================================================

===============================================================================
Segment Routing v6 Ethernet Segment Dest
===============================================================================
Instance  Eth SegId                      Num. Macs     Last Change
-------------------------------------------------------------------------------
No Matching Entries
===============================================================================
```

Example: IMET and MAC/IP routes for SRv6 transport, received on a router

The EVPN IMET and MAC/IP routes for SRv6 are advertised and expected to be received with the SRv6 Services TLV, which includes all the information related to the SRv6 SID and behavior that is used by the receiving router. For other services, transposition procedures are followed as described in Transposition procedures when advertising service routes.

```
show router bgp routes evpn incl-mcast community target:64500:1900 next-hop 2001:db8::5 hunt
```

Output example

```
===============================================================================
 BGP Router ID:192.0.2.2        AS:64500       Local AS:64500
===============================================================================
 Legend -
 Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                 l - leaked, x - stale, > - best, b - backup, p - purge
 Origin codes  : i - IGP, e - EGP, ? - incomplete

===============================================================================
BGP EVPN Inclusive-Mcast Routes
===============================================================================
-------------------------------------------------------------------------------
```

```
RIB In Entries
------------------------------------------------------------------------------
Network        : n/a
Nexthop        : 2001:db8::5
Path Id        : None
From           : 2001:db8::5
Res. Nexthop   : fe80::b449:1ff:fe01:1f
Local Pref.    : 100                    Interface Name : int-PE-2-PE-5
Aggregator AS  : None                   Aggregator     : None
Atomic Aggr.   : Not Atomic             MED            : None
AIGP Metric    : None                   IGP Cost       : 10
Connector      : None
Community      : target:64500:1900
Cluster        : No Cluster Members
Originator Id  : None                   Peer Router Id : 192.0.2.5
Flags          : Used Valid Best IGP
Route Source   : Internal
AS-Path        : No As-Path
EVPN type      : INCL-MCAST
Tag            : 0
Originator IP  : 2001:db8::5
Route Dist.    : 192.0.2.5:1900
Route Tag      : 0
Neighbor-AS    : n/a
Orig Validation: N/A
Source Class   : 0                      Dest Class     : 0
Add Paths Send : Default
Last Modified  : 00h14m18s
SRv6 TLV Type  : SRv6 L2 Service TLV (6)
SRv6 SubTLV    : SRv6 SID Information (1)
Sid            : cafe:1:0:5::
Full Sid       : cafe:1:0:5:7b1d:d000::
Behavior       : End.DT2M (24)
SRv6 SubSubTLV : SRv6 SID Structure (1)
Loc-Block-Len  : 32                     Loc-Node-Len   : 32
Func-Len       : 20                     Arg-Len        : 0
Tpose-Len      : 20                     Tpose-offset   : 64
------------------------------------------------------------------------------
PMSI Tunnel Attributes :
Tunnel-type    : Ingress Replication
Flags          : Type: RNVE(0) BM: 0 U: 0 Leaf: not required
MPLS Label     : 8068560
Tunnel-Endpoint: 2001:db8::5
------------------------------------------------------------------------------

------------------------------------------------------------------------------
RIB Out Entries
------------------------------------------------------------------------------
------------------------------------------------------------------------------
Routes : 1
==============================================================================
```

```
show router bgp routes evpn mac community target:64500:1900 next-hop 2001:db8::5 hunt
```

Output example

```
==============================================================================
 BGP Router ID:192.0.2.2        AS:64500       Local AS:64500
==============================================================================
 Legend -
 Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
```

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

===============================================================================
BGP EVPN MAC Routes
===============================================================================
-------------------------------------------------------------------------------
RIB In Entries
-------------------------------------------------------------------------------
Network        : n/a
Nexthop        : 2001:db8::5
Path Id        : None
From           : 2001:db8::5
Res. Nexthop   : fe80::b449:1ff:fe01:1f
Local Pref.    : 100                    Interface Name : int-PE-2-PE-5
Aggregator AS  : None                   Aggregator     : None
Atomic Aggr.   : Not Atomic             MED            : None
AIGP Metric    : None                   IGP Cost       : 10
Connector      : None
Community      : target:64500:1900
                 mac-mobility:Seq:0/Static
Cluster        : No Cluster Members
Originator Id  : None                   Peer Router Id : 192.0.2.5
Flags          : Used Valid Best IGP
Route Source   : Internal
AS-Path        : No As-Path
EVPN type      : MAC
ESI            : ESI-0
Tag            : 0
IP Address     : n/a
Route Dist.    : 192.0.2.5:1900
Mac Address    : 00:ca:ca:de:ba:ca
MPLS Label1    : LABEL 504286           MPLS Label2    : n/a
Route Tag      : 0
Neighbor-AS    : n/a
Orig Validation: N/A
Source Class   : 0                      Dest Class     : 0
Add Paths Send : Default
Last Modified  : 00h14m26s
SRv6 TLV Type  : SRv6 L2 Service TLV (6)
SRv6 SubTLV    : SRv6 SID Information (1)
Sid            : cafe:1:0:5::
Full Sid       : cafe:1:0:5:7b1d:e000::
Behavior       : End.DT2U (23)
SRv6 SubSubTLV : SRv6 SID Structure (1)
Loc-Block-Len  : 32                     Loc-Node-Len   : 32
Func-Len       : 20                     Arg-Len        : 0
Tpose-Len      : 20                     Tpose-offset   : 64

-------------------------------------------------------------------------------
RIB Out Entries
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Routes : 1
===============================================================================
```

3.10.7.1 VPLS and EVPN SRv6 seamless integration

EVPN and VPLS seamless integration (RFC 8560) is supported for SRv6 VPLS services, as in the case of VPLS services with EVPN-MPLS.

This seamless integration comprises the following aspects:

- SDP binds signaled by TLDP (manual, BGP-AD) or BGP can coexist in VPLS services where EVPN SRv6 is enabled.
  If an EVPN destination and an SDP binding to the same far end exist, the router brings down the SDP binding. The EVPN SRv6 destination far end is specified by the route's next hop (shown as **TEP Address** in the **show service id** *id* **segment-routing-v6 destinations** command), and not the locator. The router allows the creation of the EVPN destination and the SDP binding to the same far end but the SDP binding is kept operationally down (with a flag indicating there is an EVPN route conflict):

  ```
  show service id 10 sdp 32765:4294967291 detail | match Flag
  Flags              : EvpnRouteConflict
  ```

  In case of an SDP binding and EVPN endpoint to different far-end IPs on the same remote PE, both links are operationally up. This can happen if the SDP binding is terminated in an IPv6 address or IPv4 address different from the system address where the EVPN endpoint is terminated.
- The user can configure spoke SDPs and all the EVPN endpoints in the same split horizon group (SHG). The **split-horizon-group** command is added under the **bgp-evpn>segment-routing-v6** context so that the EVPN SRv6 endpoints are added to a split-horizon-group.

  Note: the BGP EVPN SRv6 split horizon group must reference user-configured SHGs and not auto-created SHGs.

  If the **split-horizon-group** command for **bgp-evpn>segment-routing-v6>** context is not used, the default split horizon group (in which all the EVPN endpoints are) is still used, but it is not possible to reference it on SAP or spoke SDPs. User-configured split horizon groups can be configured within the service context. The same *group-name* can be associated to SAPs, spoke SDPs, pw-templates, pw-template-bindings and EVPN-SRv6 destinations. The **bgp-evpn segment-routing-v6 split-horizon-group** configuration is only allowed if the **bgp-evpn> segment-routing-v6** is disabled. No changes are allowed when **bgp-evpn> segment-routing-v6** is enabled.
- The advertisement of MACs learned on spoke SDPs or SAPs that are part of an EVPN split horizon group are disabled.
  When the SAPs, spoke SDPs, or both (manual or BGP-AD/VPLS discovered) are configured within the same **split-horizon-group** as the EVPN destinations, MAC addresses are still learned on them, but they are not advertised in EVPN.

  The previously mentioned is also true if proxy-ARP or ND is enabled and an IP-MAC pair is learned on a SAP or SDP binding that belongs to the EVPN **split-horizon-group**.

3.10.7.2 EVPN SRv6 multihoming

SR OS supports EVPN multihoming in VPLS services where SRv6 is enabled. In EVPN-MPLS VPLS services, all-active multihoming (and single-active multihoming by default) uses the ESI label to identity the source ES from which the packet was sent. At the egress PE, a lookup on the ESI label determines if the router can send the packet to a specific local ES. In EVPN-SRv6 VPLS services, the identification of the source ES is based on the addition of an argument (of type arg.FE2 as defined in RFC9252) that is appended immediately after the function when forwarding end.dt2m or u.dt2m packets.

Before configuring ESs that require arguments for split-horizon filtering, use the following command to enable the use of 16-bit arguments in base SRv6 locators:

- **MD-CLI**

  ```
  configure router segment-routing segment-routing-v6 locator argument-length 16
  ```

- **classic CLI**

```
configure router segment-routing segment-routing-v6 locator argument-length 16
```

For micro-segments the use of arguments is enabled as follows:

- **MD-CLI**

```
configure router segment-routing segment-routing-v6 micro-segment argument-length 16
```

- **classic CLI**

```
configure router segment-routing segment-routing-v6 micro-segment argument-length 16
```

The following considerations apply to EVPN multihoming in VPLS SRv6 services:

- You can associate Ethernet Segments used in SRv6 VPLS services with ports, lags, or SDPs, and can configure them as type virtual or regular.
- The VPLS SRv6 instances using ESs are configured with the **mh-mode network** command.
- No argument is allocated by the router when the ES is configured as:
  - **MD-CLI**

```
configure service system bgp evpn ethernet-segment multi-homing-mode single-active-no-esi-label
```

  - **classic CLI**

```
configure service system bgp-evpn ethernet-segment multi-homing single-active no-esi-label
```

  In any other mode, an argument in the range 1 to 4095 is dynamically allocated by the router.
- When the **argument-length** command is set to 16 (from the default 0), the SRv6 prefix plus function lengths must be a multiple of 8, as enforced by the CLI. The length of each individual field, locator, and function, may be different from a multiple of 8, if the sum is a multiple of 8, so that the argument is pushed or read at the octet boundary. For micro-segments, this octet boundary is guaranteed.

3.10.7.2.1 Advertisement and processing of arguments

Use the following command with the **all-active** or **single-active** option to configure the multihoming mode. When the ES is configured as follows and the use of arguments is enabled (as described in EVPN SRv6 multihoming), an argument is dynamically allocated and advertised with the EVPN AD per-ES routes for the ES:

- **MD-CLI**

```
configure service system bgp evpn ethernet-segment multi-homing-mode
```

- **classic CLI**

```
configure service system bgp-evpn ethernet-segment multi-homing
```

Use the following command to display an EVPN AD per-ES route used to advertise an argument.

```
show service system bgp-evpn ethernet-segment name "vES6-L2"
```

Output example

```
===============================================================================
Service Ethernet Segment
===============================================================================
Name : vES6-L2
Eth Seg Type : Virtual
Admin State : Enabled                          Oper State : Up
ESI : 01:66:00:00:00:00:00:00:00:00
Oper ESI : 01:66:00:00:00:00:00:00:00:00
Auto-ESI Type : None
AC DF Capability : Include
Multi-homing : allActive                       Oper Multi-homing : allActive
ES Split Horizon Label : 524283
ES Split Horizon Arg : 6
Source BMAC LSB : None
Lag Id : 1
ES Activation Timer : 0 secs
Oper Group : (Not Specified)
Svc Carving : manual                           Oper Svc Carving : manual
Cfg Range Type : lowest-pref

-------------------------------------------------------------------------------
DF Pref Election Information
-------------------------------------------------------------------------------
Preference    Preference Last Admin Change      Oper Pref      Do No
Mode          Value                             Value          Preempt
-------------------------------------------------------------------------------
non-revertive 20         09/07/2023 12:41:59    20             Enabled
-------------------------------------------------------------------------------
EVI Ranges: <none>
ISID Ranges: <none>
Vprn NextHop EVI Ranges : <none>
===============================================================================


235 2023/09/09 23:14:24.701 UTC MINOR: DEBUG #2001 Base Peer 1: 2001:db8::5
"Peer 1: 2001:db8::5: UPDATE
Peer 1: 2001:db8::5 - Send BGP UPDATE:
  Withdrawn Length = 0
  Total Path Attr Length = 125
  Flag: 0x90 Type: 14 Len: 48 Multiprotocol Reachable NLRI:
    Address Family EVPN
    NextHop len 16 Global NextHop 2001:db8::2
    Type: EVPN-AD Len: 25 RD: 192.0.2.2:70 ESI: 01:66:00:00:00:00:00:00:00:0
0, tag: MAX-ET Label: 0 (Raw Label: 0x0) PathId:
  Flag: 0x40 Type: 1 Len: 1 Origin: 0
  Flag: 0x40 Type: 2 Len: 0 AS Path:
  Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
  Flag: 0xc0 Type: 16 Len: 16 Extended Community:
    target:64500:70
    esi-label:7/All-Active
  Flag: 0xc0 Type: 40 Len: 37 Prefix-SID-attr:
    SRv6 Services TLV (37 bytes):-
      Type: SRV6 L2 Service TLV (6)
      Length: 34 bytes, Reserved: 0x0
     SRv6 Service Information Sub-TLV (33 bytes)
       Type: 1 Len: 30 Rsvd1: 0x0
       SRv6 SID: ::
```

```
       SID Flags: 0x0 Endpoint Behavior: 0x18 Rsvd2: 0x0
       SRv6 SID Sub-Sub-TLV
         Type: 1 Len: 6
         BL:32 NL:32 FL:16 AL:16 TL:16 TO:64
"
```

In the preceding example, the arg.FE2 argument is 16-bits long and is always transposed into the label field of the ESI label extended community. The Transposition Offset (TO) and Transposition Length (TL) now refer to the argument contained in the AD per-ES route. As previously described, the dynamically allocated argument is indicated using the following command.

```
show service system bgp-evpn ethernet-segment name
```

When the use of arguments is enabled, the EVPN Inclusive Multicast Ethernet Tag (IMET) routes are also advertised with an argument length of 16 to indicate that the service supports 16-bit arguments.

EVPN validates the routes, when received, as follows:

- EVPN accepts IMET routes that contain an argument length (AL) of 0 or 16. The argument length of 16 is accepted as long as the sum of LBL+LNL+FL (locator plus function length) is a multiple of 8. In any other case treat-as-withdraw is applied on the route at the EVPN level (the route is shown as not used in the RIB-IN **show** commands.
- EVPN accepts AD per-ES routes only if they contain an argument length (AL) of 0 or 16. The argument length of 16 is accepted as long as the sum of LBL+LNL+FL (locator plus function length) is a multiple of 8. In any other case, the route is still accepted by the EVPN module for the purpose of DF election, however the received argument is considered non-usable and is not pushed when sending end.dt2m or u.dt2m packets to the peer.

If the received IMET and AD per-ES routes from a PE follow the preceding rules, the local PE starts building ES destinations for aliasing and backup forwarding, as it would for multihoming in EVPN-MPLS services. Use the following command to check the created ES destinations.

```
show service id 70 segment-routing-v6 destinations
```

Output example

```
===============================================================================
TEP, SID (Instance 1)
===============================================================================
TEP Address       Segment Id       Oper       Mcast     Num
State MACs
-------------------------------------------------------------------------------
2001:db8::2       cafe:2:0:2:f::   Up         BUM       0
2001:db8::3       cafe:2:0:3:f::   Up         BUM       0
-------------------------------------------------------------------------------
Number of TEP, SID: 2
-------------------------------------------------------------------------------
===============================================================================


===============================================================================
Segment Routing v6 Ethernet Segment Dest (Instance 1)
===============================================================================
Eth SegId                         Num. Macs              Last Update
-------------------------------------------------------------------------------
01:66:00:00:00:00:00:00:00:00     1                      09/09/2023 23:21:44
-------------------------------------------------------------------------------
Number of entries: 1
```

```
-------------------------------------------------------------------------------
===============================================================================
```

Use the following command to check the created Ethernet Segment destinations.

```
show service id 70 segment-routing-v6 esi 01:66:00:00:00:00:00:00:00:00
```

Output example

```
===============================================================================
Segment Routing v6 Ethernet Segment Dest (Instance 1)
===============================================================================
Eth SegId                              Num. Macs           Last Update
-------------------------------------------------------------------------------
01:66:00:00:00:00:00:00:00:00          1                   09/09/2023 23:21:44
-------------------------------------------------------------------------------
Number of entries: 1
-------------------------------------------------------------------------------
===============================================================================

============================================================
Instance 1 - Segment Routing v6 Dest TEP Info
============================================================
TEP Address          Segment Id       OperState
------------------------------------------------------------
2001:db8::2          cafe:2:0:2:e::   Up
2001:db8::3          cafe:2:0:3:e::   Up
------------------------------------------------------------
Number of entries: 2
------------------------------------------------------------
```

3.11 Segment routing policies with an IPv6 data plane

Segment routing policies with an IPv6 data plane (SRv6 policies) can be used as a part of traffic engineering in SRv6. SRv6 policies build on the concepts introduced with SR policies with an MPLS data plane:

- The SRv6 policies use SRv6 SIDs instead of MPLS labels as SIDs in the segment list and as binding SID. The SIDs can be encapsulated in a segment routing header (SRH).
- SRv6 polices can be statically configured on the router or programmed through BGP. In both cases, the user should use the following command to configure a source IPv6 address for use by SRv6 Polices:

```
configure router segment-routing segment-routing-v6 source-address
```

See Segment routing policies for more information about SR policies.

The router supports uncompressed 128-bit SRv6 SIDs as the binding SID and as SIDs for the segments in the segment lists. The binding SID and all segment list SIDs must be of the same type, that is, MPLS labels or SRv6 SIDs. The SRv6 SID can correspond to a 128-bit SID of any function type (for example, node SID, adjacency SID, binding SID).

Note: An SRv6 SID can also represent a SID in an IGP flexible algorithm.

At the head end of an SRv6 policy, service packets are encapsulated in the SRv6 packet using a head end behavior type. The following table lists the head end behavior types specified in *RFC 8986*.

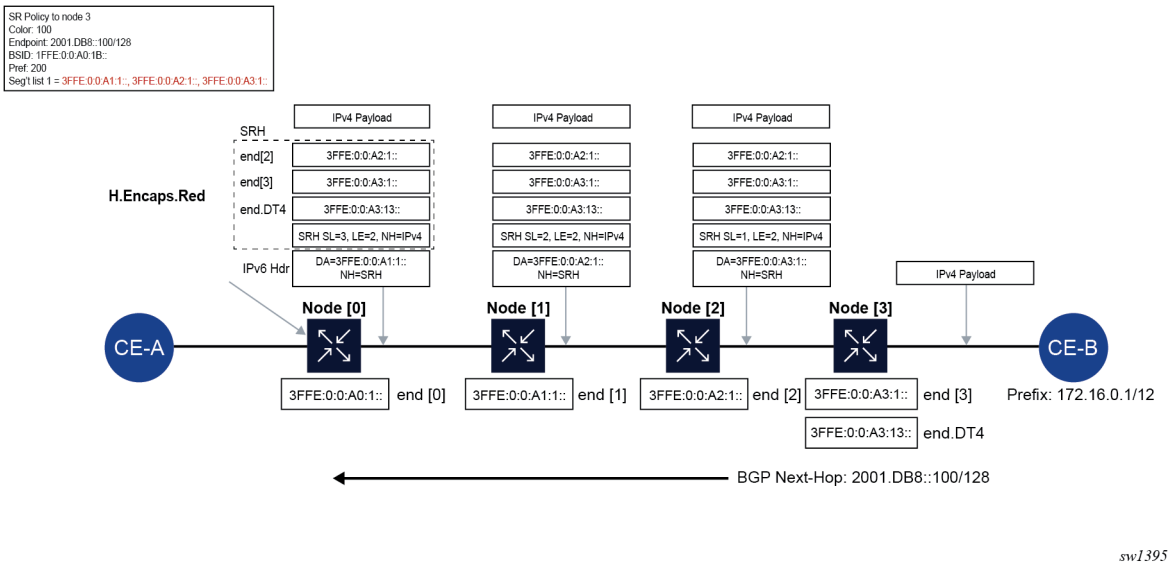
H.Encaps	SR head end with encapsulation in an SR policy
H.Encaps.Red	H.Encaps with reduced encapsulation
H.Encaps.L2	H.Encaps applied to received L2 frames
H.Encaps.L2.Red	H.Encaps.Red applied to received L2 frames

SR OS supports the following head end behavior types:

- H.Encaps.Red
- H.Encaps.L2.Red

The supported types are an optimization of the corresponding H.Encaps behavior types, that is, they exclude the first SID in the SRH of the pushed IPv6 header to reduce the length of the SRH. The Destination Address field of the pushed IPv6 header contains the first SID, while the SRH contains the other SIDs and the service SID.

Figure 51: Datapath example for a VPRNv4 service over SRv6



3.11.1 Static SRv6 policies

Use the following commands to configure a static SRv6 policy:

- `configure router segment-routing sr-policies static-policy type`

Set the type to **srv6** and only commands relevant to SRv6 policies can be configured. The default value is **sr-mpls**. If the type is set to **sr-mpls**, only MPLS labels can be configured for the binding SID and the segments in the segment lists.

- ```
  configure router segment-routing sr-policies static-policy segment-list segment srv6-sid
  ```

  Configure SRv6 SIDS for each segment in the segment lists of the SRv6 policy. SRv6 SIDs are 128-bit IPv6 addresses.

  Note: The static SRv6 policy can only be administratively enabled if its type is SRv6 and all its segments have SRv6 SIDs.

- ```
  configure router segment-routing sr-policies static-policy segment-routing-v6 binding-sid
  ```

  Configure the binding SID. The binding SID is mandatory. The system supports only one binding SID per SRv6 policy. The **locator** or the **ip-address** command in the **binding-sid** context configure the value of the binding SID. The **locator** and the **ip-address** commands are mutually exclusive. The command to use depends on the configuration of the **configure router segment-routing sr-policies static-policy head-end** command:

  – If the head end is a non-local IP address, use the **ip-address** command to configure an SRv6 binding SID for a remote SRv6 policy. The address is a 128-bit IPv6 address. The router does not perform any local checks on the configured address.

    Note: The non-local head end can only be an IPv4-formatted address to match the head end BGP router ID, which can only be IPv4.

  – If the head end is configured as **local**, configure the locator name, function, and function value for the SRv6 binding SID in the following context:

    ```
    configure router segment-routing sr-policies static-policy segment-routing-v6 binding-sid
     locator
    ```

    The router checks that a corresponding locator exists in the **configure router segment-routing segment-routing-v6** context.

    The function is a binding SID function for the static SRv6 policy and can only be **end-b6-encaps-red**. The datapath implements the End.B6.Encaps.Red as the End.B6 function for the binding SID.

    The function value is optional. If configured, the router checks the availability of the value for the configured locator. If a function value is not configured (default), the router dynamically allocates one for the function.

3.11.2 BGP SRv6 policies

The router supports BGP extensions for SRv6 policies as defined in *draft-ietf-idr-segment-routing-te-policy-11 Advertising Segment Routing Policies in BGP*. BGP uses the **sr-policy-ipv6** address family for SR policies with SAFI SR Policy (73).

SRv6 segment lists are signaled using a series of SRv6 SID sub-TLVs, as follows:

- Segment sub-TLV Type B – SID only, in the form of an IPv6 address

- SRv6 binding SID sub-TLV

The SRv6 endpoint behavior sub-TLV is not included.

3.11.3 SRv6 binding SID procedures

The router supports one binding SID per SRv6 policy. The application of binding SIDs for SRv6 policies is similar to those for SR policies with an MPLS data plane. The binding SID is programmed using the function type End.B6.Encaps.Red (as defined in *RFC 8986*).

Binding SIDs for local static SRv6 policies

If the head end is local, the router attempts to program the SR policy. The binding SID is mandatory and the configuration consists of:

- a locator name
- a function type
- a function value (optional)

Only one binding SID can be configured per SRv6 policy. The router performs the following checks and actions:

- The locator name must be configured.
- The function type must be configured. Only **end-b6-encaps-red** is supported.

  Note: The procedures defined in *RFC 8986* for the End.B6.Encaps.Red function mean that a maximum of two SRv6 policies can be concatenated to provide an end-to-end path using binding SIDs with this function.

- A function value can optionally be configured. If a value is configured, it is considered static. If no value is configured, the router tries to allocate a dynamic value for the function within the named locator.
- The binding SID function value uses the rules for the named locator:
  - **The named locator has a static and a dynamic range.**

    If a static value is configured, the router checks and allocates within the static range for the locator. If no static value is configured, the router allocates a value within the dynamic range.
  - **The named locator uses a reserved global function block from which statically and dynamically allocated function values of all locators can be drawn.**

    The router checks and allocates the value within the reserved global function block.
  - **All above checks fail.**

    The SRv6 policy is not programmed. The router retries the checks at a later time. The allocation scheme for the binding SID function values is first-come-first-served.

Binding SIDs for non-local static SRv6 policies

If the head end is not local, the router advertises the SRv6 policy in BGP toward the head end router.

The binding SID is a 128-bit IPv6 address. The router performs no local checks on the binding SID value. The router advertises the binding SID in BGP using the SRv6 binding SID sub-TLV, see BGP SRv6 policies. The SRv6 endpoint behavior sub-TLV is not included.

Receiving and programing BGP SRv6 policy routes

In general, the behavior for received SRv6 policy routes is similar to IPv4 SR-MPLS policies. For an SRv6 policy imported through BGP, the binding SID type is the indicator of the type of the SR policy. If the binding SID is an SRv6 binding SID, the router performs a longest prefix match against the route.

Note: The longest prefix match fails for SIDs with non-zero argument bits. SR OS requires that the argument bits are set to zero in the incoming SRv6 policy binding SID.

If there is no match, the SRv6 policy (candidate path) is invalid.

If there is a match, the router performs the following checks on the locator and the function value:

- If the function value collides with an already allocated function value within the static range for the locator matching the binding SID (or within the static part of a reserved global function block for the locator, if this is configured), the programming fails. If there is no collision with an already allocated static function value (regardless of the function type), the router allocates the function value to the End.B6.Encaps.Red function for the binding SID. The allocation is on a first-come-first-served basis.
- If a received SRv6 policy includes a SID structure TLV associated with the binding SID sub-TLV or the SRv6 segment sub-TLV, the router ignores it.
- If a received SRv6 policy route contains more than one SRv6 binding SID TLV, the router treats the path as invalid.

If the binding SID is valid, the router programs the SRv6 policy. The binding SID is programmed as a full /128 address.

3.11.4 Tunnel table support for SRv6 policies

Each tunnel in the tunnel table has a protocol owner. The router programs SRv6 polices in TTMv6 with **srv6-pol** as protocol owner.

If two SR policies have the same color and endpoint but a different technology (MPLS and SRv6), the router programs a separate tunnel for each.

The router treats the TTM preference for SRv6 policies the same as for MPLS-based SR policies. The router populates the TTMv6 tunnel table with the SRv6 tunnel for the SRv6 policy. IGP provides the most accurate MTU value and programs the SR tunnel MTU value in the TTM. The MTU value includes the lowest MTU among the ECMP next hops or the primary/LFA next hops for each remote locator tunnel and local adjacency SID tunnel.

Next the MTU of the SRv6 policy is derived as follows:

SRv6Policy_ MTU= SR_Tunnel_MTU - numSIDs * 16

3.11.5 SRv6 policy support for Layer 2 and Layer 3 services

The following services can resolve their next hop over an SRv6 policy in TTMv6:

- VPRNv4
- VPRNv6
- EVPN VPLS and VPWS

- BGP Shortcuts
- EVPN IFL

As with MPLS SR policies, the next hop resolution is based on the color (if present) and on the comparison of the next hop with the SR policy endpoint. The color, the endpoint, and the head end define a matching policy, but a matching policy can be an SR MPLS policy or an SRv6 policy. Therefore, the service also uses the data plane technology of the policy to select the tunnel type to resolve over. The following restrictions apply:

- MPLS services cannot resolve over SRv6 tunnels
- SRv6 services can only resolve over SRv6 tunnels

An SRv6 service cannot fall back to an MPLS tunnel type. Suppose the configuration of the resolution allows an SRv6 policy, but an SRv6 policy with a matching color and endpoint does not exist in TTMv6. In that case, the SRv6 service can fall back to any SRv6 shortest path tunnel in TTMv6 or RTM, if the locator of the tunnel matches the locator of the SRv6 service.

Use the **resolution** command to control the automatic binding of SRv6 services to SRv6 tunnels. The resolution options are:

- **tunnel-table**

  This command option resolves the route directly to a tunnel in TTMv6. The system tries to find an SRv6 policy with a matching color and endpoint for BGP routes received with an SRv6 TLV and containing an SRv6 service SID in the IPv6 tunnel table. If no such SRv6 policy is found, the resolution fails.

- **route-table**

  This command option resolves the route to a shortest-path SRv6 tunnel. If none is found, the resolution fails. This is the default behavior.

- **fallback-tunnel-to-route-table**

  This command options first tries resolving the route directly to a tunnel in the IPv6 tunnel table. If none is found, fall back to the shortest-path SRv6 resolution. If no such SRv6 policy is found, the resolution fails.

Use the **resolution** command in the following contexts to configure the resolution options:

- **For BGP shortcuts**

```
configure router bgp segment-routing-v6 family
```

- **For VPRN services**

```
configure service vprn bgp-ipvpn segment-routing-v6
```

- **For EVPN services**

```
configure service epipe bgp-evpn segment-routing-v6
```

```
configure service vpls bgp-evpn segment-routing-v6
```

- **For EVPN-IFL services**

```
configure service vprn bgp-evpn segment-routing-v6
```

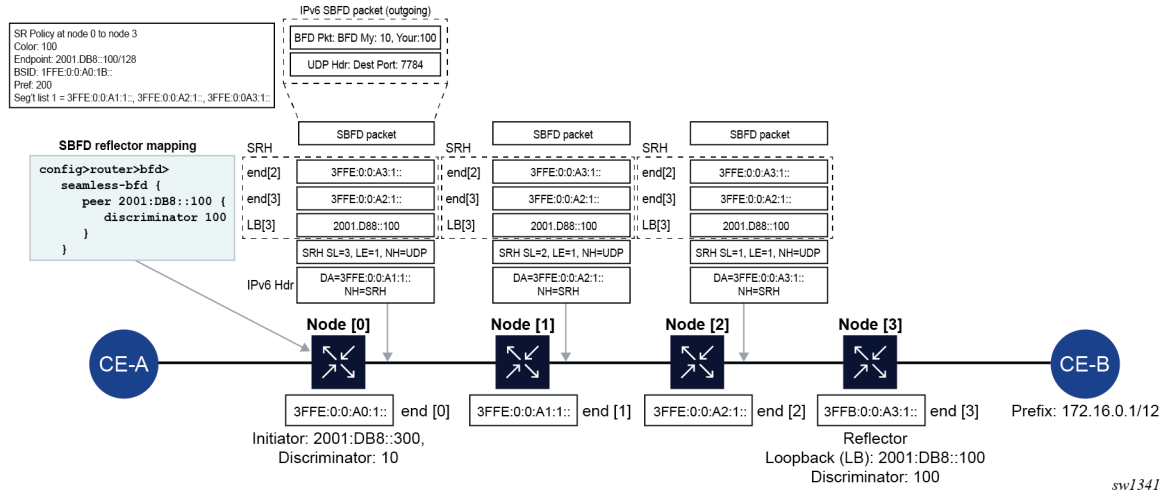
3.11.6 Seamless BFD and end-to-end protection for SRv6 policies

S-BFD and end-to-end protection, using Linear or ECMP protection modes, are supported for static and BGP SRv6 policies. To configure S-BFD and, optionally, a protection mode, apply a maintenance policy to a static or BGP SRv6 policy. The router attempts to establish an S-BFD session on each segment list of the SRv6 policy.

See Seamless BFD and end-to-end protection for SR policies for information about how to configure S-BFD and protection. S-BFD with a routed return path and S-BFD with a controlled return path are supported. In routed return path, the S-BFD reply packet is sent using a routed path from the reflector. In controlled return path, the S-BFD packet is returned to the initiator via a specified traffic engineered path such as an SRv6 Policy.

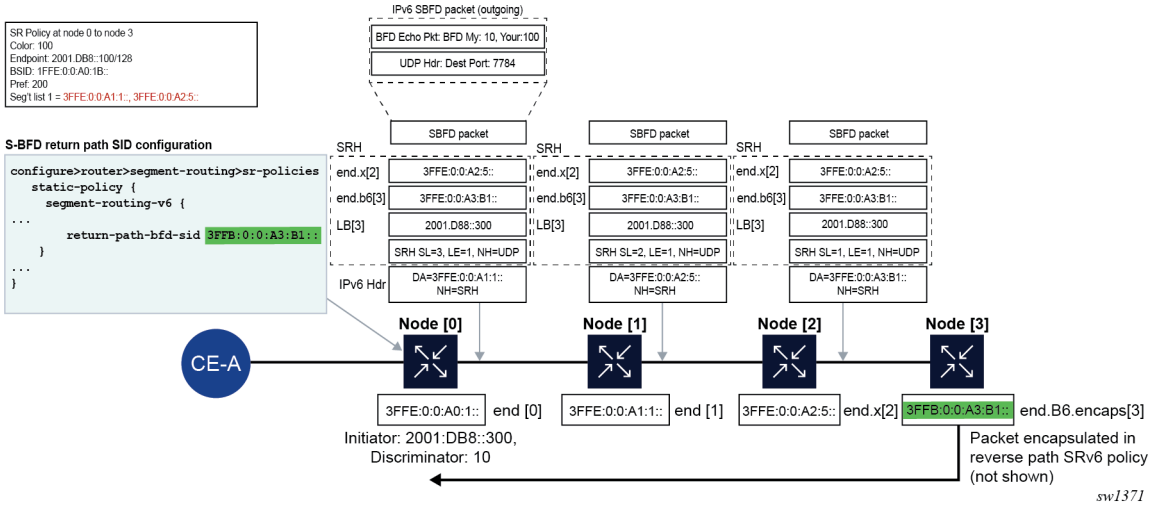
In routed return path, S-BFD packets are encapsulated in the SRv6 policy packet using a segment routing header which includes a SID containing the endpoint address of the SRv6 policy as the last SID of the header. The following figure shows S-BFD encapsulation over an SRv6 policy with routed return path.

Figure 52: S-BFD encapsulation over an SRv6 policy with routed return path



In controlled return path, a SID specified at the initiator router is also inserted in the SRH for S-BFD packets. This SID refers to a binding SID on an SRv6 policy configured at the far-end of the S-BFD session (usually the endpoint of the SRv6 policy). Echo mode is also used for the S-BFD session. The following figure shows S-BFD encapsulation over an SRv6 policy with controlled return path.

Figure 53: S-BFD encapsulation over an SRv6 policy with controlled return path



3.11.7 Traffic statistics

SRv6 policies support egress statistics similar to SR MPLS policies. See Traffic statistics, Segment routing with MPLS data plane (SR-MPLS), and Segment routing policies.

SRv6 policies do not support ingress statistics.

Traffic statistics related commands that are defined in the following context affect both SRv6 and segment routing MPLS policies. It is not possible to enable or disable statistics for only one type of policy.

```
configure router segment-routing sr-policies
```

3.11.8 Micro-segment support for SRv6 policies

SRv6 policies support micro-segment SRv6. With the seamless introduction of micro-segment SRv6 , the term SRv6 policy now refers to policies using regular (128-bit) SIDs or uSIDs (16-bit), or a combination of both.

To support uSIDs, SRv6 policies are enhanced to include the following additional functionalities:

- compression
- configuring a micro-binding SID

All services supported on SRv6 policies before the introduction of micro-segment support continue to be supported. The same applies to applications interacting with SRv6 policies: maintenance policies, sBFD, OAM, PBR, and traffic statistics.

3.11.8.1 Compression

Compression is supported for both static and BGP-signaled policies. The following compression algorithm is critical in understanding the system behavior, particularly for configuring or signaling SRv6 policies.

The compression algorithm treats each segment list the same, whether it was signaled from a peer (other node or controller) or locally configured. The algorithm parses the ordered list of segments and only compresses SIDs if they meet all the conditions of any of the three following sets of conditions:

- Set 1: B ⊂ [43, 50] AND (BL, NL, FL, AL) == ({8, 16, 24, 32, 40, 48, 56, 64}, 16, 0, 128-BL-16) AND ipv6[0, BL-1] ≠ 0 AND ipv6[BL, BL+15] ≠ 0 AND ipv6[BL+16, 127] == 0
- Set 2: (B ⊂ [52, 59] OR B ⊂ [85, 92] OR B ⊂ [93, 94]) AND (BL, NL, FL, AL) == ({8, 16, 24, 32, 40, 48, 56, 64}, 16, 16, 128-BL-32) AND ipv6[0, BL-1] ≠ 0 AND ipv6[BL, BL+15] ≠ 0 AND ipv6[BL+16, BL+31] ≠ 0 AND ipv6[BL+32, 127] == 0
- Set 3: (B ⊂ [52, 59] OR B ⊂ [85, 92] OR B ⊂ [93, 94]) AND (BL, NL, FL, AL) == ({8, 16, 24, 32, 40, 48, 56, 64}, 0, 16, 128-BL-16) AND ipv6[0, BL-1] ≠ 0 AND ipv6[BL, BL+15] ≠ 0 AND ipv6[BL+16, BL+127] == 0

where

- B = Behavior
- BL = Block Length
- NL = Node Length
- FL = Function Length
- AL = Argument Length
- ipv6[X, Y] = the bits number X up to number Y of the IPv6 address (the SID), with bit number 0 being the MSB

B, BL, NL, FL, and AL are either obtained from the static policy configuration or the "Endpoint Behavior and Structure" sub-TLV signaled with the SID in BGP. The availability of these values to the compression algorithm is a necessary (but not sufficient) condition for compression to be applied to the associated SID.

The range of values between square brackets corresponds to the IANA-registered values of END with NEXT-CSID, END.X with NEXT-CSID, END.T with NEXT-CSID, and END.B6.Encaps with NEXT-CSID (and their respective flavors). The compression algorithm only considers these behaviors for possible compression.

A detailed list of behaviors supported by the compression algorithm is provided in the following table.

Table 27: Behaviors supported by the compression algorithm

Value	Description
43	End with NEXT-CSID
44	End with NEXT-CSID & PSP
45	End with NEXT-CSID & USP
46	End with NEXT-CSID, PSP & USP
47	End with NEXT-CSID & USD
48	End with NEXT-CSID, PSP & USD
49	End with NEXT-CSID, USP & USD
50	End with NEXT-CSID, PSP, USP & USD

Value	Description
52	End.X with NEXT-CSID
53	End.X with NEXT-CSID & PSP
54	End.X with NEXT-CSID & USP
55	End.X with NEXT-CSID, PSP & USP
56	End.X with NEXT-CSID & USD
57	End.X with NEXT-CSID, PSP & USD
58	End.X with NEXT-CSID, USP & USD
59	End.X with NEXT-CSID, PSP, USP & USD
85	End.T with NEXT-CSID
86	End.T with NEXT-CSID & PSP
87	End.T with NEXT-CSID & USP
88	End.T with NEXT-CSID, PSP & USP
89	End.T with NEXT-CSID & USD
90	End.T with NEXT-CSID, PSP & USD
91	End.T with NEXT-CSID, USP & USD
92	End.T with NEXT-CSID, PSP, USP & USD
93	End.B6.Encaps with NEXT-CSID
94	End.B6.Encaps.Red with NEXT-CSID

The values between curly brackets correspond to a specific condition set on SR OS that the Block Length corresponds to one of the block length-configurable values.

To ensure backward compatibility with the regular SIDs based implementation, SIDs are not compressed in the following cases:

- any SID, including uSIDs, that do not satisfy all of the conditions of any set
- SIDs without an associated behavior and structure
- SIDs with a behavior corresponding to a regular SID

The compression algorithm does not conduct additional verifications on the received SIDs. However, if a uN segment followed by a uA segment (encoded without a uN) is received, the algorithm ensures that they are part of the same container, instead of the uN being the last SID of a container and the uA the first SID of the next container.

The system treats the segment list as programmable if, after compression (if any compression was possible), it fits in the datapath capabilities in terms of SRH size. However, other, non-micro-segment specific conditions, exist for effectively programming a segment list.

3.11.8.2 Policy segment configuration

Knowing how the compression algorithm works is critical for the configuration phase. The user should configure a policy with the understanding of whether the segment list will fit in the SRH. This applies to policies configured on SR OS, and those originated by peers or controllers and destined for an SR OS device.

On SR OS, to express the intent for compression, use the commands in the following context.

```
configure router segment-routing sr-policies static-policy segment-list segment behavior-and-
structure
```

These commands configure behavior and structure parameters of the SID. Either all command options must be configured or none may be configured; there is no default value for any command option.
The values for the behavior are limited to the base (flavorless) behaviors, as flavors play no role in compression; only micro-segment behaviors are supported. However, as indicated above, the compression algorithm can process any flavor value. The three length values correspond to the structure of the SID. The Argument Length is not configurable and always set to 128-BL-NL-FL.

These command options are associated with the IPv6 address whether the policy is local or BGP-signaled. In the latter case, the system adds these values in the Endpoint Behavior and Structure sub-TLV.

The user must configure the IPv6 address and its behavior and structure consistently.

The introduction of micro-segment support comes with the ability to specify up to 24 segments. More precisely, the SR OS allows a user to configure 24 segments if these are micro-segments (for example, an IPv6 address with a behavior and structure) but limits the number of segments to 7 in case of regular segments (for example, an IPv6 address without a behavior and structure). No compressibility verification is made at configuration or validation. The user must ensure that whatever is configured fits into the datapath capabilities.

Note:
The SR OS BGP implementation does not issue a BGP Update Message if it exceeds 4096 bytes. This limitation sets a restriction on the number of segment lists and the number of segments (especially segments with behavior and structure information) per-segment list that the user can configure. Considering segment lists only made of uSIDs (for example, an IPv6 address with behavior and structure information), the message size limit is exceeded by 13 segment lists of 8 uSIDs.

Note:
SR OS systems do not support End.T (any flavor) as a locally-instantiated SID. The CLI for configuring SRv6 policy segments supports that behavior only to allow interoperation with systems which support locally-instantiated End.T SIDs.

Note:
SR OS allows a user to configure segments of segment-lists which are in fact precompressed uSIDs. These segments must be configured without a behavior and structure. Considering a system with the ability to create an SRH with 7 segments and a typical block length of 32 bits and a micro SID length of 16, SR OS allows paths containing as many as 42 uSIDs to form.

3.11.8.3 Microbinding SID

Micro-segment support includes the ability to configure a microbinding SID, which can either be remote (for a BGP-signaled policy) or local (for a local policy). For a remote policy, the IPv6 address of the SID can be configured, but behavior and structure are not configurable. The SR OS head end ignores this information if received with a binding SID. Use the CLI commands in the following context to configure a local microbinding SID in the context of a micro-segment locator.

```
configure router segment-routing sr-policies static-policy segment-routing-v6 binding-sid
 micro-segment-locator
```

Use the CLI commands in the following context to configure remote binding SIDs.

```
configure router segment-routing sr-policies static-policy segment-routing-v6 binding-sid ip-
address
```

The function value, if configured, must be within the static range of local SID values. If it is not configured, the system selects a function value within the dynamic range of local SID values. Only a single behavior is supported: End.B6.Encaps.Red with NEXT-CSID. A single binding SID is supported per-SRv6 policy.

If a policy is received from BGP, the SR OS performs specific micro-segment checks. Having determined that a binding SID IPv6 address is covered by a locally configured micro-segment SID block, the system verifies that:

- the IPv6 address has the following format <block><uN><uB6> (the <block><uB6> format is not supported)
- the uN value corresponds to a locally configured uN
- the uB6 value is available under the block

Ultimately, the system programs two FIB entries:

- <block><uN><uB6>::/(block-length+32)
- <block><uB6>::/(block-length+16)

3.12 Assignment of loopback interface addresses from an SRv6 locator subnet

SR OS supports assigning IPv6 addresses to a system or a loopback interface drawn from a classic or micro-segment SRv6 locator prefix subnet.

To enable this feature, the user must configure an IPv6 address for the system or loopback interface from a classic or a micro-segment locator by setting bits in the least significant octet of the locator, as described in Assigning an IPv6 address from a classic SRv6 locator and Assigning an IPv6 address from a micro-segment SRv6 locator. Only a /128 subnet mask value is allowed for the IPv6 address. The locator must be assigned to algorithm 0. Addresses from a flex-algo locator are not allowed.

A number of checks are performed in CPM to enforce the correct setting of the prefix bits and mask, and to verify the overlap with a locator subnet. See CPM support with classic SRv6 locator and CPM support with micro-segment SRv6 locator for more information.

A system or a loopback interface that is configured to use an IPv6 address from an algorithm 0 locator address space behaves like any other system or loopback interface. It can be injected into IGP or BGP

routing protocols and into a VRF. Its address can be used as the source or destination address of control plane protocol and OAM packets.

Note:
This type of system or loopback interface cannot be added into an IES service.

3.12.1 Assigning an IPv6 address from a classic SRv6 locator

When assigning an IPv6 address from a classic SRv6 locator, the feature reserves the least significant octet of the locator subnet for IPv6 address of system and loopback interfaces. This is the main subnet of the locator with bits to the right of the node ID set to zero, except for the interface bits.

The following is an example of the address assignment.

- locator subnet: fc01:1:0:105::/64
- locator subnet identifier: fc01:1:0:105:0:0:0:**0**
- system/loopback interface addresses allowed are: fc01:1:0:105:0:0:0:00[**01-FF**]/**128**

This feature supports a mask value of /128 only. A maximum of 255 addresses can be allocated from the last octet of the locator. A value of zero for that octet represents the identifier of the entire subnet and is not allocatable.

3.12.1.1 CPM support with classic SRv6 locator

The following checks are performed in the CPM when assigning an IPv6 address from an SRv6 locator subnet. If any of the checks fails, the configuration fails.

- An address of the system interface or a loopback interface can only be drawn from the subnet of a classic SRv6 locator prefix assigned to algorithm 0.
- An address of the system interface or a loopback interface can only be drawn from a classic SRv6 locator prefix subnet that satisfies the following condition:

  {Prefix-length + function-length} < 120 bits
- An address of the system interface or a loopback interface must have all bits to the right of the classic SRv6 locator's node ID field set to zero, except for the bits of the least significant octet.
- Creating a new locator or changing the prefix, function-length, or prefix-length of a configured locator are not allowed if an address from the locator prefix subnet is already assigned to the system or a loopback interface. Users must first delete the corresponding addresses from the system interface or loopback interfaces.

3.12.1.2 7750 SR and 7950 XRS datapath support

The following procedures are supported in datapath:

- When the user configures an IPv6 address of the system interface or of a loopback interface and that address is drawn from the subnet of a locally configured locator, the CPM adds the /128 prefix into the route table and into the FIB.
  - A packet matching the more specific route of the interface in the FIB is extracted to the CPM for local interface processing.

  - A packet matching the less specific locator route of the locator in the FIB undergoes SRv6 tunnel termination processing, as usual.
- Ingress PE and transit P routers forward packets destined for this system or loopback interface using either the more specific interface route, if advertised in IGP or BGP by the owner router, or the less specific locator route the interface address is drawn from.
- A packet destined for the system or a loopback interface which IPv6 address is drawn from a local locator prefix subnet can be received with an SRH in its encapsulation. An example use case of a packet received from a transit router that forwarded the packet over the remote LFA or TI-LFA repair tunnel of the corresponding less specific locator route when the SIDs of the repair tunnel are of USP SRH-mode.

3.12.2 Assigning an IPv6 address from a micro-segment SRv6 locator

When assigning an IPv6 address from a micro-segment SRv6 locator, the feature reserves the least significant octet of the micro-segment locator subnet for IPv6 address of system and loopback interfaces. This is the main subnet of the locator with bits to the right of the uN set to zero, except the interface bits.

The following is an example of the address assignment.

- locator subnet: fc00:0:105::/**48**
- locator subnet identifier: fc00:0:105:0:0:0:0:**0**
- system/loopback interface addresses allowed are: fc00:0:105:0:0:0:0:00[**01-FF**]/**128**

This feature supports a mask value of /128 only. A maximum of 255 addresses can be allocated from the last octet of the locator. A value of zero for that octet represents the identifier of the entire subnet and is not allocatable.

3.12.2.1 CPM support with micro-segment SRv6 locator

The following checks are performed in CPM when assigning an IPv6 address from a micro-segment SRv6 locator subnet. If any of the checks fail, the configuration fails.

- An address of the system interface or a loopback interface can only be drawn from the subnet of a micro-segment SRv6 locator prefix assigned to algorithm 0.
- An address of the system interface or a loopback interface can only be drawn from a micro-segment SRv6 locator prefix subnet that satisfies the following condition:

  {Block-length + sid-length(uN) + sid-length(uA or uDT)} < 120 bits
- An address of the system interface or a loopback interface must have all bits to the right of the micro-segment SRv6 locator’s uN field set to zero except bits of the least significant octet.
- Creating a new locator, changing the uN value of a configured locator, or changing the block length of a configured locator are not allowed if an address from the locator prefix subnet is already assigned to the system or a loopback interface. The user must first delete the corresponding addresses from the system interface or loopback interfaces.

3.12.2.2 7750 SR and 7950 XRS datapath support

The datapath behavior for a micro-segment SRv6 locator is the same as that of a classic SRv6 locator. See 7750 SR and 7950 XRS datapath support for more information.

4 MPLS forwarding policy

The MPLS forwarding policy provides an interface for adding user-defined label entries into the label FIB of the router and user-defined tunnel entries into the tunnel table.

The endpoint policy allows the user to forward unlabeled packets over a set of user-defined direct or indirect next hops with the option to push a label stack on each next hop. Routes are bound to an endpoint policy when their next hop matches the endpoint address of the policy.

The user defines an endpoint policy by configuring a set of next-hop groups, each consisting of a primary and a backup next hops, and binding an endpoint to it.

The label-binding policy provides the same capability for labeled packets. In this case, labeled packets matching the ILM of the policy binding label are forwarded over the set of next hops of the policy.

The user defines a label-binding policy by configuring a set of next-hop groups, each consisting of a primary and a backup next hops, and binding a label to it.

This feature is targeted for router programmability in SDN environments.

4.1 Introduction to MPLS forward policy

This section provides information about configuring and operating a MPLS forwarding policy using CLI.

There are two types of MPLS forwarding policy:

- endpoint policy
- label-binding policy

The endpoint policy allows the user to forward unlabeled packets over a set of user-defined direct or indirect next hops, with the option to push a label stack on each next hop. Routes are bound to an endpoint policy when their next hop matches the endpoint address of the policy.

The label-binding policy provides the same capability for labeled packets. In this case, labeled packets matching the ILM of the policy binding label are forwarded over the set of next hops of the policy.

The data model of a forwarding policy represents each pair of {primary next hop, backup next hop} as a group and models the ECMP set as the set of Next-Hop Groups (NHGs). Flows of prefixes can be switched on a per NHG basis from the primary next hop, when it fails, to the backup next hop without disturbing the flows forwarded over the other NHGs of the policy. The same can be performed when reverting back from a backup next hop to the restored primary next hop of the same NHG.

4.2 Feature validation and operation procedures

The MPLS forwarding policy follows a number of configuration and operation rules which are enforced for the lifetime of the policy.

There are two levels of validation:

- The first level validation is performed at provisioning time. The user can bring up a policy (**no shutdown** command) after these validation rules are met. Afterwards, the policy is stored in the forwarding policy database.
- The second level validation is performed when the database resolves the policy.

4.2.1 Policy parameters and validation procedure rules

The following policy parameters and validation rules apply to the MPLS forwarding policy and are enforced at configuration time:

- A policy must have either the **endpoint** or the **binding-label** command to be valid or the **no shutdown** is not allowed. These commands are mutually exclusive per policy.
- The **endpoint** command specifies that this policy is used for resolving the next hop of IPv4 or IPv6 packets, of BGP prefixes in GRT, of static routes in GRT, of VPRN IPv4 or IPv6 prefixes, or of service packets of EVPN prefixes. It is also used to resolve the next hop of BGP-LU routes.

  The resolution of prefixes in these contexts matches the IPv4 or IPv6 next-hop address of the prefix against the address of the endpoint. The family of the primary and backup next hops of the NHGs within the policy are not relevant to the resolution of prefixes using the policy.

  See Tunnel table handling of MPLS forwarding policy for information about CLI commands for binding these contexts to an endpoint policy.
- The **binding-label** command allows the user to specify the label for binding to the policy such that labeled packets matching the ILM of the binding label can be forwarded over the NHG of the policy.

  The ILM entry is created only when a label is configured. Only a provisioned binding label from a reserved label block is supported. The name of the reserved label block using the **reserved-label-block** command must be configured.

  The payload of the packet forwarded using the ILM (payload underneath the swapped label) can be IPv4, IPv6, or MPLS. The family of the primary and backup next hops of the NHG within the policy are not relevant to the type of payload of the forwarded packets.
- Changes to the values of the **endpoint** and **binding-label** parameters require a **shutdown** of the specific forwarding policy context.
- A change to the name of the **reserved-label-block** requires a **shutdown** of the **forwarding-policies** context. The **shutdown** is not required if the user extends or shrinks the range of the **reserved-label-block**.
- The **preference** parameter allows the user to configure multiple endpoint forwarding policies with the same endpoint address value or multiple label-binding policies with the same binding label; providing the capability to achieve a 1:N backup strategy for the forwarding policy. Only the most preferred, lowest numerical preference value, policy is activated in datapath as described in Policy resolution and operational procedures.
- Changes to the value of parameter **preference** requires a shutdown of the specific **forwarding-policy** context.
- A maximum of eight label-binding policies, with different preference values, are allowed for each unique value of the binding label.

  Label-binding policies with exactly the same value of the tuple {**binding label** | **preference**} are duplicate and their configuration is not allowed.

  The user cannot perform **no shutdown** on the duplicate policy.

- A maximum eight endpoint policies, with different preference values, are allowed for each unique value of the tuple {**endpoint**}.

  Endpoint policies with exactly the same value of the tuple {**endpoint**, **reference**} are duplicate and their configuration is not allowed.

  The user cannot perform **no shutdown** on the duplicate policy.
- The **metric** parameter is supported with the endpoint policy only and is inherited by the routes which resolve their next hop to this policy.
- The **revert-timer** command configures the time to wait before switching back the resolution from the backup next hop to the restored primary next hop within an NHG. By default, this timer is disabled meaning that the NHG immediately reverts to the primary next hop when it is restored.

  The revert timer is restarted each time the primary next hop flaps and comes back up again while the previous timer is still running. If the revert timer value is changed while the timer is running, it is restarted with the new value.
- The MPLS forwarding policy feature allows for a maximum of 32 NHGs consisting of, at most, one primary next hop and one backup next hop.
- The **next-hop** command allows the user to specify a direct next-hop address or an indirect next-hop address.
- A maximum of ten labels can be specified for a primary or backup direct next hop using the **pushed-labels** command. The label stack is programmed using a super-NHLFE directly on the outgoing interface of the direct primary or backup next hop.

  Note: This policy differs from the SR-TE LSP or SR policy implementation which can push a total of 11 labels because of the fact it uses a hierarchical NHLFE (super-NHLFE with maximum 10 labels pointing to the top SID NHLFE).

- The **resolution-type** {**direct**| **indirect**} command allows a limited validation at configuration time of the NHGs within a policy. The **no shutdown** command fails if any of these rules are not satisfied. The following are the rules of this validation:
  - NHGs within the same policy must be of the same resolution type.
  - A forwarding policy can have a single NHG of resolution type **indirect** with a primary next hop only or with both primary and backup next hops. An NHG with backup a next hop only is not allowed.
  - A forwarding policy has one or more NHGs of resolution type **direct** with a primary next hop only or with both primary and backup next hops. An NHG with a backup next hop only is not allowed.
  - A check is performed to ensure the address value of the primary and backup next hop, within the same NHG, are not duplicates. No check is performed for duplicate primary or backup next-hop addresses across NHGs.
  - A maximum of 64,000 forwarding policies of any combination of label binding and endpoint types can be configured on the system.
- The IP address family of an endpoint policy is determined by the family of the **endpoint** parameter. It is populated in the TTMv4 or TTMv6 table accordingly. A label-binding policy does not have an IP address family associated with it and is programmed into the label (ILM) table.

  The following are the IP type combinations for the primary and backup next hops of the NHGs of a policy:
  - A primary or a backup indirect next hop with no pushed labels (label-binding policy) can be IPv4 or IPv6. A mix of both IP types is allowed within the same NHG.

– A primary or backup direct next hop with no pushed labels (label-binding policy) can be IP types IPv4 or IPv6. A mix of both families is allowed within the same NHG.

– A primary or a backup direct next hop with pushed labels (both endpoint and label binding policies) can be IP types IPv4 or IPv6. A mix of both families is allowed within the same NHG.

4.2.2 Policy resolution and operational procedures

This section describes the validation of parameters performed at resolution time, as well as the details of the resolution and operational procedures.

- The following parameter validation is performed by the forwarding policy database at resolution time; meaning each time the policy is re-evaluated:
  – If the NHG primary or backup next hop resolves to a route whose type does not match the configured value in **resolution-type**, that next hop is made operationally "down".

    A DOWN reason code shows in the state of the next hop.

  – The primary and backup next hops of an NHG are looked up in the routing table. The lookups can match a direct next hop in the case of the direct resolution type and therefore the next hop can be part of the outgoing interface primary or secondary subnet. They can also match a static, IGP, or BGP route for an indirect resolution type, but only the set of IP next hops of the route are selected. Tunnel next hops are not selected and if they are the only next hops for the route, the NHG is put in operationally "down" state.

  – The first 32, out of a maximum of 64, resolved IP next hops are selected for resolving the primary or backup next hop of a NHG of **resolution-type indirect**.

  – If the primary next hop is operationally "down", the NHG uses the backup next hop if it is UP. If both are operationally DOWN, the NHG is DOWN. See Datapath support for details of the active path determination and the failover behavior.

  – If the binding label is not available, meaning it is either outside the range of the configured **reserved-label-block**, or is used by another MPLS forwarding policy or by another application, the label-binding policy is put operationally "down" and a retry mechanism checks the label availability in the background.

    A policy level DOWN reason code is added to alert users who may then choose to modify the binding label value.

  – No validation is performed for the pushed label stack of or a primary or backup next hop within a NHG or across NHGs. Users are responsible for validating their configuration.
- The forwarding policy database activates the best endpoint policy, among the named policies sharing the same value of the tuple {**endpoint**}, by selecting the lowest preference value policy. This policy is then programmed into the TTM and into the tunnel table in the datapath.

  If this policy goes DOWN, the forwarding policy database performs a re-evaluation and activates the named policy with the next lowest preference value for the same tuple {**endpoint**}.

  If a more preferred policy comes back up, the forwarding policy database reverts to the more preferred policy and activates it.
- The forwarding policy database similarly activates the best label-binding policy, among the named policies sharing the same binding label, by selecting the lowest preference value policy. This policy is then programmed into the label FIB table in the datapath as detailed in Datapath support.

If this policy goes DOWN, the forwarding policy database performs a re-evaluation and activates the named policy with the next lowest preference value for the same binding label value.

If a more preferred policy comes back up, the forwarding policy database reverts to the more preferred policy and activates it.

- The active policy performs ECMP, weighted ECMP, or CBF over the active (primary or backup) next hops of the NHG entries.
- When used in the PCEP application, each LSP in a label-binding policy is reported separately by PCEP using the same binding label. The forwarding behavior on the node is the same whether the binding label of the policy is advertised in PCEP or not.
- A policy is considered UP when it is the best policy activated by the forwarding policy database and when at least one of its NHGs is operationally UP. A NHG of an active policy is considered UP when at least one of the primary or backup next hops is operationally UP.
- When the **config>router>mpls** or **config>router>mpls>forwarding-policies** context is set to **shutdown**, all forwarding policies are set to DOWN in the forwarding policy database and deprogrammed from IOM and datapath.

  Prefixes which were being forwarded using the endpoint policies revert to the next preferred resolution type configured in the specific context (GRT, VPRN, or EVPN).
- When an NHG is set to **shutdown**, it is deprogrammed from the IOM and datapath. Flows of prefixes which were being forwarded to this NHG are re-allocated to other NHGs based on the ECMP, Weighted ECMP, or CBF rules.
- When a policy is set to **shutdown**, it is deleted in the forwarding policy database and deprogrammed from the IOM and datapath. Prefixes which were being forwarded using this policy are reverted to the next preferred resolution type configured in the specific context (GRT, VPRN, or EVPN).
- The **no forwarding-policies** command deletes all policies from the forwarding policy database provided none of them are bound to any forwarding context (GRT, VPRN, or EVPN). Otherwise, the command fails.

4.3 Tunnel table handling of MPLS forwarding policy

An endpoint forwarding policy validated as the most preferred policy for an endpoint address is added to the TTMv4 or TTMv6 according to the address family of the address of the **endpoint** parameter. A new owner of **mpls-fwd-policy** is used. A tunnel ID is allocated to each policy and is added into the TTM entry for the policy. For more information about the **mpls-fwd-policy** command, used to enable MPLS forwarding policy in different services, see the following guides:

- *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 2 Services and EVPN Guide*
- *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 3 Services Guide: IES and VPRN*
- *7450 ESS, 7750 SR, 7950 XRS, and VSR Router Configuration Guide*
- *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide*

The TTM preference value of a forwarding policy is configurable using the parameter **tunnel-table-pref**. The default value of this parameter is 4.

Each individual endpoint forwarding policy can also be assigned a preference value using the **preference** command with a default value of 255. When the forwarding policy database compares multiple forwarding policies with the same endpoint address, the policy with the lowest numerical preference value is activated

and programmed into TTM. The TTM preference assigned to the policy is its own configured value in the **tunnel-table-pref** parameter.

If an active forwarding policy preference has the same value as another tunnel type for the same destination in TTM, then routes and services which are bound to both types of tunnels use the default TTM preference for the two tunnel types to select the tunnel to bind to as shown in Table 28: Route preferences .

Table 28: Route preferences

Route preference	Value	Release introduced
ROUTE_PREF_RIB_API	3	new in 16.0.R4 for RIB API IPv4 and IPv6 tunnel table entry
ROUTE_PREF_MPLS_FWD_POLICY	4	new in 16.0.R4 for MPLS forwarding policy of endpoint type
ROUTE_PREF_RSVP	7	—
ROUTE_PREF_SR_TE	8	new in 14.0
ROUTE_PREF_LDP	9	—
ROUTE_PREF_OSPF_TTM	10	new in 13.0.R1
ROUTE_PREF_ISIS_TTM	11	new in 13.0.R1
ROUTE_PREF_BGP_TTM	12	modified in 13.0.R1 (pref was 10 in R12)
ROUTE_PREF_UDP	254	introduced with 15.0 MPLS-over-UDP tunnels
ROUTE_PREF_GRE	255	—

An active endpoint forwarding policy populates the highest pushed label stack size among all its NHGs in the TTM. Each service and shortcut application on the router uses that value and performs a check of the resulting net label stack by counting all the additional labels required for forwarding the packet in that context.

This check is similar to the one performed for SR-TE LSP and SR policy features. If the check succeeds, the service is bound or the prefix is resolved to the forwarding policy. If the check fails, the service does not bind to this forwarding policy. Instead, it binds to a tunnel of a different type if the user configured the use of other tunnel types. Otherwise, the service goes down. Similarly, the prefix does not get resolved to the forwarding policy and is either resolved to another tunnel type or becomes unresolved.

For more information about the **resolution-filter** CLI commands for resolving the next hop of prefixes in GRT, VPRN, and EVPN MPLS into an endpoint forwarding policy, see the following guides:

- *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 2 Services and EVPN Guide*
- *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 3 Services Guide: IES and VPRN*
- *7450 ESS, 7750 SR, 7950 XRS, and VSR Router Configuration Guide*
- *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide*

BGP-LU routes can also have their next hop resolved to an endpoint forwarding policy.

4.4 Datapath support

Note: The datapath model for both the MPLS forwarding policy and the RIB API is the same. Unless explicitly stated, the selection of the active next hop within each NHG and the failover behavior within the same NHG or across NHGs is the same.

4.4.1 NHG of resolution type indirect

Each NHG is modeled as a single NHLFE. The following are the specifics of the datapath operation:

- Forwarding over the primary or backup next hop is modeled as a swap operation from the binding label to an implicit-null label over multiple outgoing interfaces (multiple NHLFEs) corresponding to the resolved next hops of the indirect route.
- Packets of flows are sprayed over the resolved next hops of an NHG with resolution of type indirect as a one-level ECMP spraying. See Spraying of packets in a MPLS forwarding policy .
- An NHG of resolution type **indirect** uses a single NHLFE and does not support uniform failover. It has CPM program only the active, the primary or backup, and the indirect next hop at any point in time.
- Within an NHG, the primary next hop is the preferred active path in the absence of any failure of the NHG of resolution type **indirect**.
- The forwarding database tracks the primary or backup next hop in the routing table. A **route delete** of the primary indirect next hop causes CPM to program the backup indirect next hop in the datapath.

  A **route modify** of the indirect primary or backup next hop causes CPM to update its resolved next hops and to update the datapath if it is the active indirect next hop.
- When the primary indirect next hop is restored and is added back into the routing table, CPM waits for an amount of time equal to the user programmed revert-timer before updating the datapath. However, if the backup indirect next hop fails while the timer is running, CPM updates the datapath immediately.

4.4.2 NHG of resolution type direct

The following rules are used for a NHG with a resolution type of **direct**:

- Each NHG is modeled as a pair of {primary, backup} NHLFEs. The following are the specifics of the label operation:
  - For a label-binding policy, forwarding over the primary or backup next hop is modeled as a swap operation from the binding label to the configured label stack or to an implicit-null label (if the **pushed-labels** command is not configured) over a single outgoing interface to the next hop.
  - For an endpoint policy, forwarding over the primary or backup next hop is modeled as a push operation from the binding label to the configured label stack or to an implicit-null label (if the **pushed-labels** command is not configured) over a single outgoing interface to the next hop.
  - The labels, configured by the **pushed-labels** command, are not validated.
- By default, packets of flows are sprayed over the set of NHGs with resolution of type **direct** as a one-level ECMP spraying. See Spraying of packets in a MPLS forwarding policy .
- The user can enable weighted ECMP forwarding over the NHGs by configuring weight against all the NHGs of the policy. See Spraying of packets in a MPLS forwarding policy .

- Within an NHG, the primary next hop is the preferred active path in the absence of any failure of the NHG of resolution type direct.

  Note: The RIB API feature can change the active path away from the default. The gRPC client can issue a next-hop switch instruction to activate any of the primary or backup path at any time.

- The NHG supports uniform failover. The forwarding policy database assigns a Protect-Group ID (PG-ID) to each of the primary next hop and the backup next hop and programs both of them in the datapath. A failure of the active path switches traffic to the other path following the uniform failover procedures as described in Active path determination and failover in a NHG of resolution type direct.
- The forwarding database tracks the primary or backup next hop in the routing table. A **route delete** of the primary or backup direct next hop causes CPM to send the corresponding PG-ID switch to the datapath.

  A **route modify** of the direct primary or backup next hop causes CPM to update the MPLS forwarding database and to update the datapath because both next hops are programmed.
- When the primary direct next hop is restored and is added back into the routing table, CPM waits for an amount of time equal to the user programmed **revert-timer** before activating it and updating the data path. However, if the backup direct next hop fails while the timer is running, CPM activates it and updates the datapath immediately. The latter failover to the restored primary next hop is performed using the uniform failover procedures as described in Active path determination and failover in a NHG of resolution type direct.

  Note: RIB API does not support the revert timer. The gRPC client can issue a next-hop switch instruction to activate the restored primary next hop.

- CPM keeps track and updates the IOM for each NHG with the state of active or inactive of its primary and backup next hops following a failure event, a reversion to the primary next hop, or a successful next-hop switch request instruction (RIB API only).

4.4.2.1 Active path determination and failover in a NHG of resolution type direct

An NHG of resolution type **direct** supports uniform failover either within an NHG or across NHGs of the same policy. These uniform failover behaviors are mutually exclusive on a per-NHG basis depending on whether it has a single primary next hop or it has both a primary and backup next hops.

When an NHG has both a primary and a backup next hop, the forwarding policy database assigns a Protect-Group ID (PG-ID) to each and programs both in datapath. The primary next hop is the preferred active path in the absence of any failure of the NHG.

During a failure affecting the active next hop, or the primary or backup next hop, CPM signals the corresponding PG-ID switch to the datapath which then immediately begins using the NHLFE of the other next hop for flow packets mapped to NHGs of all forwarding polices which share the failed next hop.

An interface down event sent by CPM to the datapath causes the datapath to switch the PG-ID of all next hops associated with this interface and perform the uniform failover procedure for NHGs of all policies which share these PG-IDs.

Any subsequent network event causing a failure of the newly active next hop while the originally active next hop is still down, blackholes traffic of this NHG until CPM updates the policy to redirect the affected flows to the remaining NHGs of the forwarding policy.

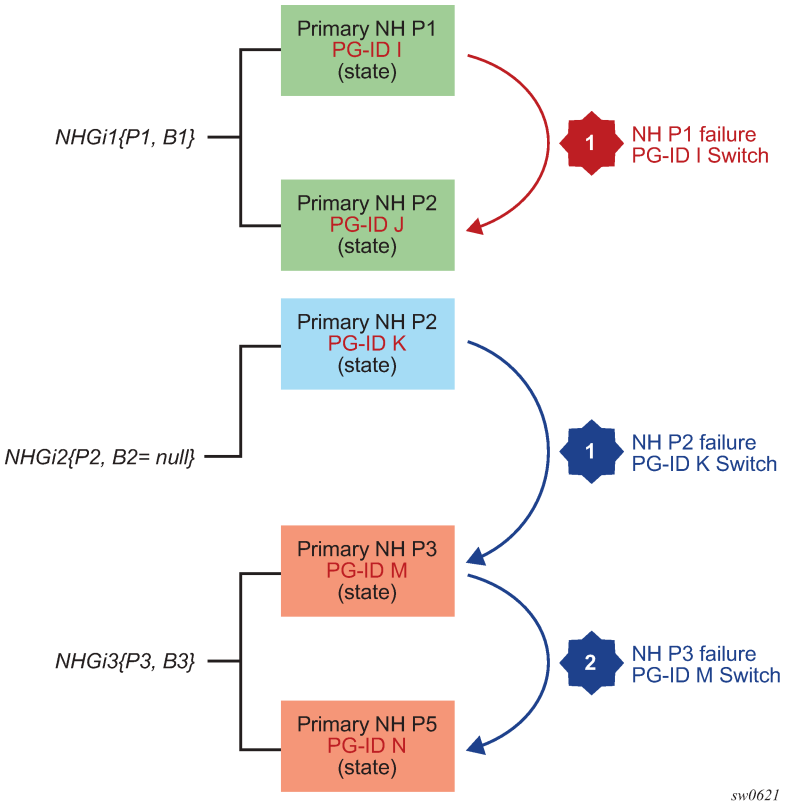
When the NHG has only a primary next hop and it fails, CPM signals the corresponding PG-ID switch to the datapath which then uses the uniform failover procedure to immediately re-assign the affected flows to the other NHGs of the policy.

A subsequent failure of the active next hop of a NHG the affected flow was re-assigned to in the first failure event, causes the datapath to use the uniform failover procedure to immediately switch the flow to the other next hop within the same NHG.

Figure 54: NHG failover based on PG-ID switch illustrates the failover behavior for the flow packets assigned to an NHG with both a primary and backup next hop and to an NHG with a single primary next hop.

The notation NHGi{Pi,Bi} refers to NHG "i" which consists of a primary next hop (Pi) and a backup next hop (Bi). When an NHG does not have a backup next hop, it is referred to as NHGi{Pi,Bi=null}.

Figure 54: NHG failover based on PG-ID switch



4.4.3 Spraying of packets in a MPLS forwarding policy

When the node operates as an LER and forwards unlabeled packets over an endpoint policy, the spraying of packets over the multiple NHGs of type **direct** or over the resolved next hops of a single NHG of type **indirect** follows prior implementation. See the *7450 ESS, 7750 SR, 7950 XRS, and VSR Interface Configuration Guide*.

When the node operates as an LSR, it forwards labeled packets matching the ILM of the binding label over the label-binding policy. An MPLS packet, including a MPLS-over-GRE packet, received over any network

IP interface with a binding label in the label stack, is forwarded over the primary or backup next hop of either the single NHG of type **indirect** or of a selected NHG among multiple NHGs of type **direct**.

The router performs the following procedures when spraying labeled packets over the resolved next hops of a NHG of resolution type **indirect** or over multiple NHGs of type **direct**.

1. The router performs the GRE header processing as described in *MPLS-over-GRE termination* if the packet is MPLS-over-GRE encapsulated. See the *7450 ESS, 7750 SR, 7950 XRS, and VSR Router Configuration Guide*.
2. The router then pops one or more labels and if there is a match with the ILM of a binding label, the router swaps the label to implicit-null label and forwards the packet to the outgoing interface. The outgoing interface is selected from the set of primary or backup next hops of the active policy based on the LSR hash on the headers of the received MPLS packet.
   - The hash calculation follows the method in the user configuration of the command **lsr-load-balancing** {**lbl-only** | **lbl-ip** | **ip-only**} if the packet is MPLS-only encapsulated.
   - The hash calculation follows the method described in *LSR Hashing of MPLS-over-GRE Encapsulated Packet* if the packet is MPLS-over-GRE encapsulated. See the *7450 ESS, 7750 SR, 7950 XRS, and VSR Interface Configuration Guide*.

4.4.4 Outgoing packet Ethertype setting and TTL handling in label binding policy

The following rules determine how the router sets the Ethertype field value of the outgoing packet:

- If the swapped label is not the Bottom-of-Stack label, the Ethertype is set to the MPLS value.
- If the swapped label is the Bottom-of-Stack label and the outgoing label is not implicit-null, the Ethertype is set to the MPLS value.
- If the swapped label is the Bottom-of-Stack label and the outgoing label is implicit-null, the Ethertype is set to the IPv4 or IPv6 value when the first nibble of the exposed IP packet is 4 or 6 respectively.

The router sets the TTL of the outgoing packet as follows:

- The TTL of a forwarded IP packet is set to MIN(MPLS_TTL-1, IP_TTL), where MPLS_TTL refers to the TTL in the outermost label in the popped stack and IP_TTL refers to the TTL in the exposed IP header.
- The TTL of a forwarded MPLS packet is set to MIN(MPLS_TTL-1, INNER_MPLS_TTL), where MPLS_TTL refers to the TTL in the outermost label in the popped stack and INNER_MPLS_TTL refers to the TTL in the exposed label.

4.4.5 Ethertype setting and TTL handling in endpoint policy

The router sets the Ethertype field value of the outgoing packet to the MPLS value.

The router checks and decrements the TTL field of the received IPv4 or IPv6 header and sets the TTL of all labels of the label stack specified in the **pushed-labels** command according to the following rules:

1. The router propagates the decremented TTL of the received IPv4 or IPv6 packet into all labels of the pushed label stack for a prefix in GRT.
2. The router then follows the configuration of the TTL propagation in the case of a IPv4 or IPv6 prefix forwarded in a VPRN context:

```
– config>router>ttl-propagate>vprn-local {none | vc-only | all}
– config>router>ttl-propagate>vprn-transit {none | vc-only | all}
```

```
    – config>service>vprn>ttl-propagate>local {inherit | none | vc-only | all}
    – config>service>vprn>ttl-propagate>transit {inherit | none | vc-only | all}
```

When a IPv6 packet in GRT is forwarded using an endpoint policy with an IPv4 endpoint, the IPv6 explicit null label is pushed first before the label stack specified in the **pushed-labels** command.

4.5 Weighted ECMP enabling and validation rules

Weighted ECMP is supported within an endpoint or a label-binding policy when the NHGs are of resolution type **direct**. Weighted ECMP is not supported with an NHG of type **indirect**.

Weighted ECMP is performed on labeled or unlabeled packets forwarded over the set of NHGs in a forwarding policy when all NHG entries have a **load-balancing-weight** configured. If one or more NHGs have **no load-balancing-weight** configured, the spraying of packets over the set of NHGs reverts to plain ECMP.

Also, the **weighted-ecmp** command in GRT (**config>router>weighted-ecmp**) or in a VPRN instance (**config>service>vprn>weighted-ecmp**) are not required to enable the weighted ECMP forwarding in an MPLS forwarding policy. These commands are used when forwarding over multiple tunnels or LSPs. Weighted ECMP forwarding over the NHGs of a forwarding policy is strictly governed by the explicit configuration of a weight against each NHG.

The weighted ECMP normalized weight calculated for a NHG index causes the datapath to program this index as many times as the normalized weight dictates for the purpose of spraying the packets.

4.6 Statistics

4.6.1 Ingress statistics

The ingress statistics feature is associated with the binding label, that is the ILM of the forwarding policy, and provides aggregate packet and octet counters for packets matching the binding label.

The per-ILM statistic index for the MPLS forwarding policy features is assigned at the time the first instance of the policy is programmed in the datapath. All instances of the same policy, for example, policies with the same binding-label, regardless of the **preference** parameter value, share the same statistic index.

The statistic index remains assigned as long as the policy exists and the **ingress-statistics** context is not shutdown. If the last instance of the policy is removed from the forwarding policy database, the CPM frees the statistic index and returns it to the pool.

If ingress statistics are not configured or are shutdown in a specific instance of the forwarding policy, identified by a unique value of pair {**binding-label**, **preference**} of the forwarding policy, an assigned statistic index is not incremented if that instance of the policy is activated

If a statistic index is not available at allocation time, the allocation fails and a retry mechanism checks the statistic index availability in the background.

4.6.2 Egress statistics

Egress statistics are supported for both binding-label and endpoint MPLS forwarding policies; however, egress statistics are only supported in case where the next-hops configured within these policies are of resolution type **direct**. The counters are attached to the NHLFE of each next hop. Counters are effectively allocated by the system at the time the instance is programmed in the data-path. Counters are maintained even if an instance is deprogrammed and values are not reset. If an instance is reprogrammed, traffic counting resumes at the point where it last stopped. Traffic counters are released and therefore traffic statistics are lost when the instance is removed from the database when the egress statistic context is deleted, or when egress statistics are disabled (**egress-statistics shutdown**).

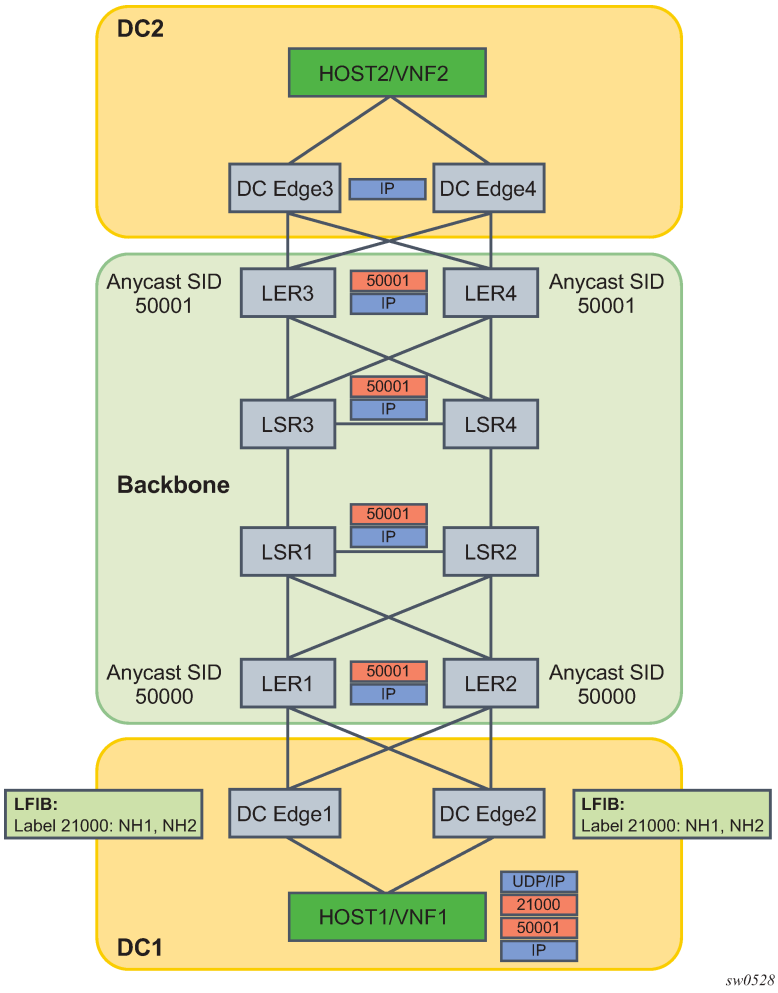
No retry mechanism is available for egress statistics. The system maintains a state per next hop and per-instance about whether the allocation of statistic indexes is successful. If the system is not able to allocate all the needed indexes on a specified instance because of a lack of resources, the user should disable egress statistics on that instance, free the required number of statistics indexes, and re-enable egress statistics on the needed entry. The selection of which other construct to release statistic indexes from is beyond the scope of this document.

4.7 Configuring static labeled routes using MPLS forwarding policy

4.7.1 Steering flows to an indirect next hop

Figure 55: Traffic steering to an indirect next hop using a static labeled route illustrates the traffic forwarding from a Virtual Network Function (VNF1) residing in a host in a Data Center (DC1) to VNF2 residing in a host in DC2 over the segment routing capable backbone network. DC1 and DC2 do not support segment routing and MPLS while the DC Edge routers do not support segment routing. Hence, MPLS packets of VNF1 flows are tunneled over a UDP/IP or GRE/IP tunnel and a static labeled route is configured on DC Edge1/2 to steer the decapsulated packets to the remote DC Edge3/4.

Figure 55: Traffic steering to an indirect next hop using a static labeled route



The following are the datapath manipulations of a packet across this network:

1. Host in DC1 pushes MPLS-over-UDP (or MPLS-over-GRE) header with outer IP destination address matching its local DC Edge1/2. It also pushes a static label 21000 which corresponds to the binding label of the MPLS forwarding policy configured in DC Edge1/2 to reach remote DC Edge3/4 (anycast address). The bottom of the label stack is the anycast SID for the remote LER3/4.
2. The label 21000 is configured on both DC Edge1 and DC Edge2 using a label-binding policy with an indirect next hop pointing to the static route to the destination prefix of DC Edge3/4. The backup next-hop points to the static route to reach some DC Edge5/6 in another remote DC (not shown).
3. There is EBGP peering between DC Edge1/2 and LER1/2 and between DC Edge3/4 and LER3/4.
4. DC Edge1/2 removes the UDP/IP header (or GRE/IP header) and swaps label 21000 to implicit-null and forwards (ECMP spraying) to all resolved next hops of the static route of the primary or backup next hop of the label-binding policy.
5. LER1/2 forwards based on the anycast SID to remote LER3/4.
6. LER3/4 removes the anycast SID label and forwards the inner IP packet to DC Edge3/4 which then forwards to Host2 in DC2.

The following CLI commands configure the static labeled route to achieve this use case. It creates a label-binding policy with a single NHG that is pointing to the first route as its primary indirect next hop and the second route as its backup indirect next hop. The primary static route corresponds to a prefix of remote DC Edge3/4 router and the backup static route to the prefix of a pair of edge routers in a different remote DC. The policy is applied to routers DC Edge1/2 in DC1.

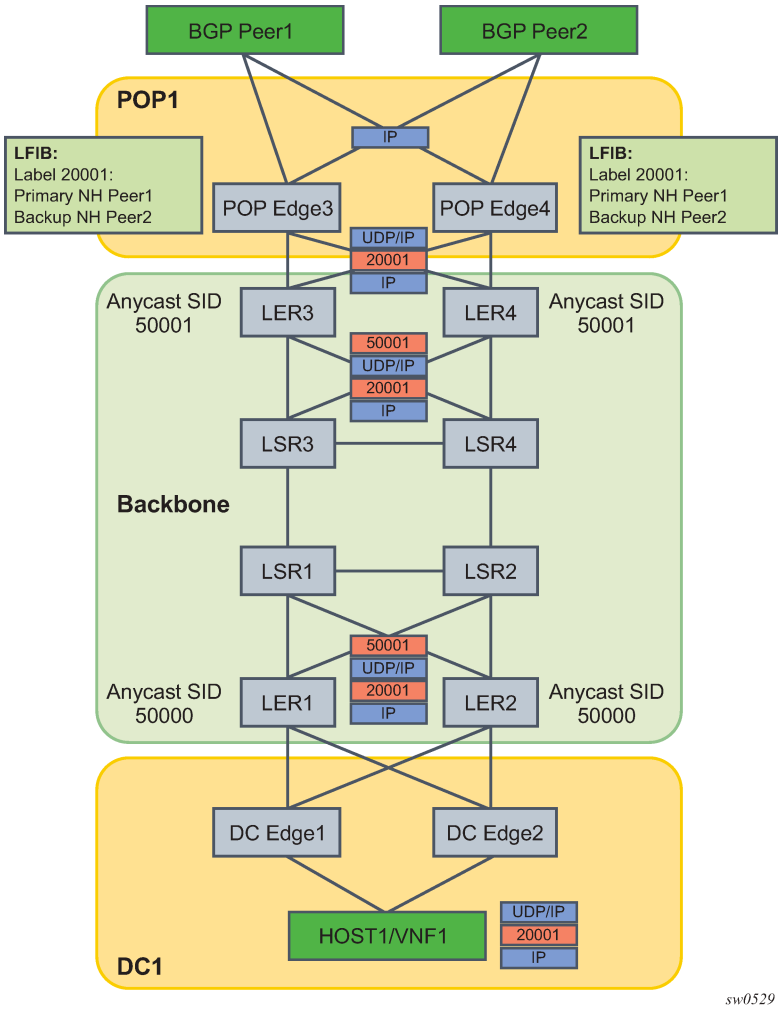
```
— config>router
    — static-route-entry fd84:a32e:1761:1888::1/128
        — next-hop 3ffe::e0e:e05
            — no shutdown
        — next-hop 3ffe::f0f:f01
            — no shutdown
    — static-route-entry fd22:9501:806c:2387::2/128
        — next-hop 3ffe::1010:1002
            — no shutdown
        — next-hop 3ffe::1010:1005
            — no shutdown
—
— config>router>mpls-labels
    — reserved-label-block static-label-route-lbl-block
        — start-label 20000 end-label 25000
—
— config>router>mpls
    — forwarding-policies
        — reserved-label-block static-label-route-lbl-block
        — forwarding-policy static-label-route-indirect
            — binding-label 21000
            — revert-timer 5
            — next-hop-group 1 resolution-type indirect
                — primary-next-hop
                    — next-hop fd84:a32e:1761:1888::1
                — backup-next-hop
                    — next-hop fd22:9501:806c:2387::2
```

4.7.2 Steering flows to a direct next hop

Figure 56: Traffic steering to a direct next hop using a static labeled route illustrates the traffic forwarding from a Virtual Network Function (VNF1) residing in a host in a Data Centre (DC1) to outside of the customer network via the remote peering Point Of Presence (POP1).

The traffic is forwarded over a segment routing capable backbone. DC1 and POP1 do not support segment routing and MPLS while the DC Edge routers do not support segment routing. Hence, MPLS packets of VNF1 flows are tunneled over a UDP/IP or GRE/IP tunnel and a static labeled route is configured on POP Edge3/4 to steer the decapsulated packets to the needed external BGP peer.

Figure 56: Traffic steering to a direct next hop using a static labeled route



The intent is to override the BGP routing table at the peering routers (POP Edge3 and Edge4) and force packets of a flow originated in VNF1 to exit the network using a primary external BGP peer Peer1 and a backup external BGP peer Peer2, if Peer1 is down. This application is also referred to as Egress Peer Engineering (EPE).

The following are the datapath manipulations of a packet across this network:

1. DC Edge1/2 receives a MPLS-over-UDP (or a MPLS-over-GRE) encapsulated packet from the host in the DC with the outer IP destination address set to the remote POP Edge3/4 routers in peering POP1 (anycast address). The host also pushes the static label 20001 for the remote external BGP Peer1 it wants to send to.
2. This label 20001 is configured on POP Edge3/4 using the MPLS forwarding policy feature with primary next hop of Peer1 and backup next hop of Peer2.
3. There is EBGP peering between DC Edge1/2 and LER1/2, and between POP Edge3/4 and LER3/4, and between POP Edge3/4 and Peer1/2.
4. LER1/LER2 pushes the anycast SID of remote LER3/4 as part of the BGP route resolution to a SR-ISIS tunnel or SR-TE policy.

5. LER3/4 removes the anycast SID and forwards the GRE packet to POP Edge3/4.
6. POP Edge3/4 removes UDP/IP (or GRE/IP) header and swaps the static label 20001 to implicit null and forwards to Peer1 (primary next hop) or to Peer2 (backup next hop).

The following CLI commands configure the static labeled route to achieve this use case. It creates a label-binding policy with a single NHG containing a primary and backup direct next-hops and is applied to peering routers POP Edge3/4.

```
—
— config>router>mpls-labels
    — reserved-label-block static-label-route-lbl-block
        — start-label 20000 end-label 25000
—
— config>router>mpls
    — forwarding-policies
        — forwarding-policy static-label-route-direct
            — binding-label 20001
            — revert-timer 10
            — next-hop-group 1 resolution-type direct
                — primary-next-hop
                    — next-hop fd84:a32e:1761:1888::1
                — backup-next-hop
                    — next-hop fd22:9501:806c:2387::2
```

5 gRPC-based RIB API

5.1 RIB/FIB API overview

Each router stores information about how to forward IP and MPLS packets in a set of RIB (routing information base) and FIB (forwarding information base) tables. These tables are conventionally populated by the management plane of the router (static entries) and by control plane protocols such as BGP, OSPF, ISIS, RSVP, LDP or segment routing.

In some SDN (Software Defined Networking) use cases, it may be useful to augment the RIB/FIB state held by a router to also include forwarding entries programmed by an external controller. SR OS supports a powerful and flexible gRPC-based RIB API service for this purpose. Using gRPC between a controller (gRPC client) and a router (gRPC server) has many benefits:

- gRPC is open source, with broad industry support and a rich ecosystem of tools and applications.
- gRPC is fast and efficient. The combination of HTTP/2 and protocol buffers ensures that a minimum number of bytes are sent across the wire as part of an RPC.
- gRPC is supported by many languages and platforms, including C, C++, C#, Go, Java, Node.js, Python, and Ruby.

To build a gRPC client that implements the RIB API service you must obtain, from Nokia, the protobuf definition file for the Nokia SR OS RIB API service. This protobuf file defines a RIB API service and its supported RPCs. The RIB API service supports one bidirectional streaming RPC called *Modify* and one unary RPC called GetVersion. The *Modify* RPC is used to add, delete or replace entries in any of the following RIB/FIB tables:

- IPv4 route table of the base router
- IPv6 route table of the base router
- IPv4 tunnel table
- IPv6 tunnel table
- MPLS label forwarding table

The *GetVersion* RPC allows the client to request the overall RIB API version and the individual RIB table versions supported by the router.

For maximum programming flexibility and speed, the entries added by the RIB-API service are not processed or stored as configuration data; they are provided directly to the control plane and modeled as though learned from a pseudo-protocol. RIB-API entries have the same persistence characteristics as protocol routes: if a router (gRPC server) detects that a gRPC client has disconnected or terminated its RPC, or if the router reboots, dependent RIB API entries are removed and must be re-programmed if persistence is required.

A gRPC client cannot delete entries it does not own, including routes from other protocols, but it can supersede routes from other sources through appropriate programming of preference values.

A gRPC client can read RIB/FIB entries programmed using the RIB API service (by any client) and obtain other state information that it needs using the gNMI management interface. gNMI is another gRPC-based service supported by the router and it supports RPCs for configuration, one-time state retrieval and

telemetry state subscriptions. The same client can have active gNMI and RIB API RPCs with the same target router and at the same time using the same TCP connection.

5.2 RIB/FIB API fundamentals

The *Modify* RPC allows a gRPC client to add, modify or delete RIB/FIB entries. To accomplish this, the client sends a stream of *ModifyRequest* messages to the server (router) and receives, in return, a stream of *ModifyResponse* messages. Each *ModifyRequest* message can include multiple *Request* messages. Each *Request* message has a 64-bit ID (used to pair it with a *Response* message) and conveys one of the following instructions:

- A request to **add** an entry to one of the five supported RIB/FIB tables. The add operation requires the client to specify values for all parameters of the route, tunnel or label entry being programmed. If the add operation is successful, the RIB/FIB entry is considered owned by the client that carried out the transaction.
- A request to **replace** an entry in one of the five supported RIB/FIB tables. This operation completely replaces a RIB/FIB entry that was previously programmed by the same client. All the parameters of the new route, tunnel or label entry must be specified, even values that did not change from the previous entry.
- A request to **delete** an entry in one of the five supported RIB/FIB tables. This operation deletes a RIB/FIB entry that was previously programmed by the same client. The delete operation requires only the key values of the entry that should be deleted.
- An **end-of-rib** marker for one of the five supported RIB/FIB tables. This operation is used to accelerate the removal of stale entries associated with a RIB/FIB table, instead of waiting for purge timers to expire. Additional details are discussed in this chapter.
- A **next-hop-switch** instruction. The client sends this request to manually activate the primary or backup next hop associated with a specific next-hop-group of a specific tunnel or label entry. This may be done to facilitate a maintenance action or to manually revert traffic back to a primary next hop after it recovers from a failure that diverted traffic to a backup next hop.

The following general points should be noted:

- The router does not support multiple RIB API RPC sessions with the same client IP. If a client machine has multiple independent controllers, they need to interact with the router using different IP addresses.
- It is up to the client to choose a unique 64-bit identifier for each *Request* transaction, but Request IDs must increase throughout the lifetime of the RPC session.
- If a gRPC client omits any parameter that is considered mandatory by the server side, the router assumes that the intended value for the parameter is zero (0). This may cause an error if the zero value is invalid or unavailable.
- A status code of OK sent by the router to a client (in a *ModifyResponse* message) only indicates that the request was valid. In the case of an add/delete/modify operation, it does not mean that the FIB was actually modified and in the case of a next-hop-switchover it does not mean that the switchover has actually occurred.
- A RIB/FIB entry programmed by a gRPC client may be unusable because none of its next hops are resolvable or the requested label resources are not available. The entry is still accepted by the router and acknowledged with an OK status code response to the client. If at a later time the entry becomes usable, it is activated by the router automatically.

5.2.1 RIB/FIB API entry persistence

All states created by the RIB API service are ephemeral. In other words, when the router reboots, none of the API-programmed entries are preserved. The necessary entries must be reprogrammed by a gRPC client in the same way that BGP routes must be relearned from BGP peers after a reboot.

The persistence of a programmed RIB/FIB entry also depends on the liveness of the RPC session with the client that owns the entry, and this in turn depends on the liveness of the underlying TCP connection. If the TCP connection with a client goes down (because of link or router failures, or both, client failure, or CPM switchover by the router) the router starts a purge timer for all affected clients and marks their owned RIB/FIB entries as stale. When a client's purge timer expires all of its stale entries are removed. While a purge timer is running, the associated stale entries remain valid and usable for forwarding but are less preferred than any non-stale entry. The purge timer gives an opportunity for the disconnected client or some other client to re-program the necessary RIB/FIB tables so that forwarding continues uninterrupted.

Detection of TCP connection failures by the router (gRPC server) can be assisted by enabling TCP **keepalive** on the gRPC TCP connections. When it is enabled, TCP keepalive messages are sent to all gRPC clients, regardless of the RPCs they support (gNMI or RIB API).

On the router, TCP **keepalive** is configured by specifying 3 parameters: **idle-time**, **interval**, and **retries**. These parameters are configured in the **config>system>grpc>tcp-keepalive** context. The sending of TCP keepalives starts when the connection has been idle (no TCP segments sent or received) for more than **idle-time** seconds. At that point the router sends a probe (TCP ACK with a sequence number = current sequence number - 1) and expects a TCP ACK. It repeats this probe every **interval** seconds for the configured number of **retries** and if no response is received to any of them the connection is immediately closed, starting the purge timer if the TCP connection is supporting a Modify RPC.

When a client is done programming all entries in a particular RIB/FIB table it can optionally send an **end-of-rib** request for that table to immediately remove all stale RIB entries associated with that table, regardless of the owner client IP.

5.3 RIB/FIB API configuration overview

About this task

Configuration related to the RIB/FIB API service on the router is spread across two general areas:

- system-level GRPC configuration (**config>system>grpc** or **config>system>security>profile>grpc**)
- routing instance configuration (**config>router** or **config>service>vprn**)

To enable the router to receive and process RIB API requests from a client perform the following steps.

Procedure

Step 1. The RIB API service must be enabled at the gRPC system level: **config>system>grpc>rib-api>no shutdown**.

Step 2. Optionally, a non-zero purge-timeout can be configured: **config>system>grpc>rib-api>purge-timeout**.

The purge-timeout applies to all gRPC clients participating in the RIB API service.

Step 3. Optionally, the sending of TCP keepalives can be enabled toward all gRPC clients by configuring values under the **config>system>grpc>tcp-keepalive** context.

Step 4. One or more gRPC user accounts should be created, and these user accounts should be attached to a profile that authorizes the *GetVersion* and *Modify* RPCs associated with the RIB API service. Clients need to send a valid username and password when initiating any RPC.

Step 5. Nokia recommends using TLS-based encryption between the client and server. This involves associating a **tls-server-profile** with the gPRC server. For more information, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR System Management Guide*.

Step 6. If you want to use the RIB API service to program MPLS label entries then a **reserved-label-block** must be configured using the **config>router>rib-api>mpls>reserved-label-block** command and MPLS programming functionality must be enabled using the **config>router>rib-api>mpls>no shutdown** command.

What to do next

To enable the router to use RIB API tunnel entries for resolving specific types of static and BGP routes, additional configuration is needed. For more information, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide*.

5.4 RIB/FIB API - IPv4 route table programming

The RIB API service proto definition requires the client to provide values for all of the parameters listed in Table 29: IPv4 route table programming when performing an **add** or **replace** of an IPv4 route. When performing a **delete** operation, only the bolded parameters (the lookup keys) are required. Table 29: IPv4 route table programming describes the meaning of each parameter and its valid range.

Table 29: IPv4 route table programming

Parameter	Type	Description
prefix	string	IPv4 prefix and prefix-length in CIDR format
preferences	uint32 (0-65535)	RIB API preference, used to compare one RIB API entry to another one; the lowest preference wins
rtm_preference	uint32 (0-255)	RTM preference, used to compare RIB API entry to other routes in RTM; the lowest preference wins
metric	uint32 (0-16777215)	Route cost/metric
tunnel_next_hop	string	A remote IPv4 address that must correspond to an API-programmed IPv4 tunnel

The router's RIB API database can hold up to eight different gPRC-programmed entries per IPv4 prefix. Typically N entries would be associated with N different gRPC clients although the same client can program multiple entries for the same prefix as long as the preference values are unique.

When an IPv4 route entry is successfully added or modified in the RIB API database, the router assesses whether the entry is valid or invalid and constantly re-evaluates this status. The entry is invalid if its next hop cannot be resolved to a gRPC-programmed IPv4 tunnel that is up.

If the entry is valid, the router compares it to all other valid API-programmed entries for the same IPv4 prefix. The router chooses any non-stale entry over a stale entry, then the entry with the lowest preference value, and then if there is a tie, the lowest metric, and then if there is a still a tie, the entry from the client with the lowest 128-bit IP address (an IPv4 address is encoded in the lower 32 bits).

If the entry is valid and the best relative to other RIB API entries then it is submitted to the route table manager. This software task compares the API route to all other non-API routes it has for the same IPv4 prefix. The router chooses the entry with the lowest RTM preference value, and then if there is a tie, the lowest metric, and then if there is still a tie, the entry submitted by the protocol with the lowest default preference.

If the route table manager selects the API route as the best route it is sent to the FIB manager for programming into the datapath.

5.5 RIB/FIB API - IPv6 route table programming

The RIB API service proto definition requires the client to provide values for all of the parameters listed in Table 30: IPv6 route table programming when performing an **add** or **replace** of an IPv6 route. When performing a **delete** operation, only the bolded parameters (the lookup keys) are required. Table 30: IPv6 route table programming describes the meaning of each parameter and its valid range.

Table 30: IPv6 route table programming

Parameter	Type	Description
prefix	string	IPv6 prefix and prefix-length in CIDR format
preferences	uint32 (0-65535)	RIB API preference, used to compare one RIB API entry to another one; the lowest preference wins
rtm_preference	uint32 (0-255)	RTM preference, used to compare RIB API entry to other routes in RTM; the lowest preference wins
metric	uint32 (0-16777215)	Route cost/metric
tunnel_next_hop	string	A remote IPv6 address that must correspond to an API-programmed IPv6 tunnel

The router's RIB API database can hold up to eight different gPRC-programmed entries per IPv6 prefix. Typically N entries would be associated with N different gRPC clients although the same client can program multiple entries for the same prefix as long as the preference values are unique.

When an IPv6 route entry is successfully added or modified in the RIB API database, the router assesses whether the entry is valid or invalid and constantly re-evaluates this status. The entry is invalid if its next hop cannot be resolved to a gRPC-programmed IPv6 tunnel that is up.

If the entry is valid, the router compares it to all other valid API-programmed entries for the same IPv6 prefix. The router chooses any non-stale entry over a stale entry, then the entry with the lowest preference value, and then if there is a tie, the lowest metric, and then if there is a still a tie, the entry from the client with the lowest 128-bit IP address (an IPv4 address is encoded in the lower 32 bits).

If the entry is valid and the best relative to other RIB API entries then it is submitted to the route table manager. This software task compares the API route to all other non-API routes it has for the same IPv6 prefix. The router chooses the entry with the lowest RTM preference value, and then if there is a tie, the lowest metric, and then if there is still a tie, the entry submitted by the protocol with the lowest default preference.

If the route table manager selects the API route as the best route it is sent to the FIB manager for programming into the datapath.

5.6 RIB/FIB API - IPv4 tunnel table programming

The RIB API service proto definition requires the client to provide values for all of the parameters listed in Table 31: IPv4 tunnel table programming when performing an **add** or **replace** of an IPv4 MPLS tunnel. When performing a **delete** operation, only the bolded parameters (the lookup keys) are required. Table 31: IPv4 tunnel table programming describes the meaning of each parameter and its valid range.

Table 31: IPv4 tunnel table programming

Parameter	Type	Description
prefix	string	IPv4 host address
preferences	uint32 (0-65535)	RIB API preference, used to compare one RIB API entry to another one; the lowest preference wins
ttm_preference	uint32 (0-255)	TTM preference, used in the programming of the tunnel in TTM
metric	uint32 (0-16777215)	Route cost/metric
next-hop-group[id]	list	A list of next-hop groups
id	uint32 (1-32)	Unique identifier of the next-hop group. Selected by the client
weight	uint32	Weight assigned to the next-hop-group when weighted ECMP is needed between next-hop-groups
primary	—	Mandatory
ip_address	string	IPv4 or IPv6 address on a local subnet; can be a secondary address
pushed_label_stack	list of uint32	A list of one or more MPLS labels, up to ten MPLS labels

Parameter	Type	Description
backup	—	Optional
ip_address	string	IPv4 or IPv6 address on a local subnet; can be a secondary address
pushed_label_stack	list of uint32	A list of up to ten MPLS labels
egress-statistics	—	—
enable	boolean	Indicates whether statistics collection is enabled for this entry

The router's RIB API database can hold up to eight different gPRC-programmed entries per IPv4 tunnel endpoint. Typically N entries would be associated with N different gRPC clients although the same client can program multiple entries for the same tunnel endpoint as long as the preference values are unique

When an IPv4 tunnel endpoint entry is successfully added or modified in the RIB API database, the router assesses whether the entry is valid or invalid and constantly re-evaluates this status. The tunnel is invalid if none of its primary next hops can be resolved to an interface that is up or if MPLS programming using the RIB API is currently administratively disabled.

If the IPv4 tunnel entry is valid the router compares it to all other valid API-programmed entries for the same IPv4 endpoint address. The router chooses any non-stale entry over a stale entry, then the entry with the lowest preference value, and then if there is a tie, the lowest metric, and then if there is a still a tie, the entry from the client with the lowest 128-bit IP address (an IPv4 address is encoded in the lower 32 bits).

If the entry is valid and the best relative to other RIB API entries then it is programmed into the FIB and added to the base router IPv4 tunnel table. The tunnel entry is now active and can be used to resolve the next hops of other routes. For more information, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide*.

5.7 RIB/FIB API - IPv6 tunnel table programming

The RIB API service proto definition requires the client to provide values for all of the parameters listed in Table 32: IPv6 tunnel table programming when performing an **add** or **replace** of an IPv6 MPLS tunnel. When performing a **delete** operation, only the bolded parameters (the lookup keys) are required. Table 32: IPv6 tunnel table programming describes the meaning of each parameter and its valid range.

Table 32: IPv6 tunnel table programming

Parameter	Type	Description
prefix	string	IPv6 host address
preferences	uint32 (0-65535)	RIB API preference, used to compare one RIB API entry to another one; the lowest preference wins
ttm_preference	uint32 (0-255)	TTM preference, used in the programming of the tunnel in TTM

Parameter	Type	Description
metric	uint32 (0-16777215)	Route cost/metric
next-hop-group[id]	list	A list of next-hop groups
id	uint32 (1-32)	Unique identifier of the next-hop group; selected by the client
weight	uint32	Weight assigned to the next-hop-group when weighted ECMP is needed between next-hop-groups
primary	—	Mandatory
ip_address	string	IPv4 or IPv6 address on a local subnet; can be a secondary address.
pushed_label_stack	list of uint32	A list of one or more MPLS labels, up to ten MPLS labels
backup	—	Optional
ip_address	string	IPv4 or IPv6 address on a local subnet; can be a secondary address
pushed_label_stack	list of uint32	A list of up to ten MPLS labels
egress-statistics	—	—
enable	boolean	Indicates whether statistics collection is enabled for this entry

The router's RIB API database can hold up to eight different gPRC-programmed entries per IPv6 tunnel endpoint. Typically N entries would be associated with N different gRPC clients although the same client can program multiple entries for the same tunnel endpoint as long as the preference values are unique.

When an IPv6 tunnel endpoint entry is successfully added or modified in the RIB API database, the router assesses whether the entry is valid or invalid and constantly re-evaluates this status. The tunnel is invalid if none of its primary next hops can be resolved to an interface that is up or if MPLS programming using the RIB API is currently administratively disabled.

If the IPv6 tunnel entry is valid the router compares it to all other valid API-programmed entries for the same IPv6 endpoint address. The router chooses any non-stale entry over a stale entry, then the entry with the lowest preference value, and then if there is a tie, the lowest metric, and then if there is a still a tie, the entry from the client with the lowest 128-bit IP address (an IPv4 address is encoded in the lower 32 bits).

If the entry is valid and the best relative to other RIB API entries then it is programmed into the FIB and added to the base router IPv6 tunnel table. The tunnel entry is now active and can be used to resolve the next hops of other routes. For more information, see the 7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide.

5.8 RIB/FIB API - MPLS LFIB programming

The RIB API service proto definition requires the client to provide values for all of the parameters listed in Table 33: MPLS LFIB programming when performing an **add** or **replace** of an MPLS LFIB entry. When performing a **delete** operation, only the bolded parameters (the lookup keys) are required. Table 33: MPLS LFIB programming describes the meaning of each parameter and its valid range.

Table 33: MPLS LFIB programming

Parameter	Type	Description
prefix	string	Incoming label value
preferences	uint32 (0-65535)	RIB API preference, used to compare one RIB API entry to another one; the lowest preference wins
next-hop-group[id]	list	A list of next-hop groups; required for a SWAP operation; omitted when the operation is a POP
id	uint32 (1-32)	Unique identifier of the next-hop group; selected by the client
weight	uint32	Weight assigned to the next-hop-group when weighted ECMP is needed between next-hop-groups
primary	—	Mandatory
ip_address	string	IPv4 or IPv6 address on a local subnet; can be a secondary address
pushed_label_stack	list of uint32	A list of zero or more MPLS labels, up to ten MPLS labels
backup	—	Optional
ip_address	string	IPv4 or IPv6 address on a local subnet; can be a secondary address
pushed_label_stack	list of uint32	A list of up to ten MPLS labels
ingress-statistics	—	—
enable	bool	—
type	enum 0, 1, 2	INVALID = 0 POP = 1 SWAP = 2
egress-statistics	—	—

Parameter	Type	Description
enable	boolean	Indicates whether statistics collection is to be enabled for this entry

The router's RIB API database can hold up to eight different gPRC-programmed entries per MPLS label value. Typically N entries would be associated with N different gRPC clients although the same client can program multiple entries for the same label value as long as the preference values are unique.

When an MPLS label entry is successfully added or modified in the RIB API database, the router assesses whether the entry is valid or invalid and constantly re-evaluates this status. The label entry is invalid if it is a SWAP operation and none of the primary next hops can be resolved to an interface that is up or if MPLS programming using the RIB API is administratively disabled or if the requested incoming label has already been allocated to another owner sharing the same reserved label block or if the requested incoming label is outside the reserved label block range.

If the label entry is valid the router compares it to all other valid API-programmed entries for the same label value. The router chooses any non-stale entry over a stale entry, then the entry with the lowest preference value, and then if there is a tie, the entry from the client with the lowest 128-bit IP address (an IPv4 address is encoded in the lower 32 bits).

If the label entry is valid and best relative to other RIB API entries then it is programmed into the forwarding plane.

5.9 RIB/FIB API - using next-hop-groups, primary next hops, and backup next hops

The RIB API service proto definition allows each MPLS tunnel and each MPLS label entry to have multiple next-hop-groups, each with a primary next hop and optionally one backup next hop. When a tunnel or label entry has more than one next-hop-group, this instructs the router to spray matching traffic across the next-hop-groups based on an ECMP or weighted-ECMP algorithm.

At any time, traffic hashed to a particular next-hop-group uses only the primary or backup next hop for forwarding. The selection of the active next hop within each next-hop-group is influenced by failures and by **next-hop-switch** *Request* messages made by the owner gRPC client. The specific rules are:

- If the primary next hop is resolved to an up interface when the next-hop-group is initially activated then it immediately becomes the active next hop.
- If the primary next hop is unresolved when the next-hop-group is initially activated then no next hop is immediately activated (even if the backup next hop is up) and a fixed wait-timer is started (three seconds). If the primary next hop comes up during that timer window then it is immediately activated. If the timer runs out and the primary has not yet come up the backup next hop is activated and stays active even if the primary comes up a short while later, after the timer expired.
- If the currently active next hop fails, the system automatically activates the other next hop.
- If the system receives a **next-hop-switch** *Request* targeting this specific entry and next-hop-group then the next hop indicated in the *Request* message is immediately activated, as long as it is up. If the requested next hop is down the message is ignored.

Note: The router returns a status of OK in response to a **next-hop-switch** *Request* as long as the key values identify a next-hop-group that exists for a tunnel or label entry owned by the gRPC client, even if the needed next hop is not activated.

5.10 RIB/FIB API - state and telemetry

A gRPC client can use the gRPC gNMI service (Get RPC, Subscribe RPC) to retrieve state information from the router that can help it make better programming decisions. All states maintained by the router (and exposed to model-driven management interfaces) are available to the gRPC client.

RIB/FIB API also introduces additional YANG state models that are complementary to the programming actions. This new state is available through the following YANG paths:

- `state/router/route-fib`
- `state/router/tunnel-fib`
- `state/router/label-fib`
- `state/router/rib-api/route`
- `state/router/rib-api/tunnel`
- `state/router/rib-api/label`

The corresponding **show** commands are also provided for reference.

- **show router fib-telemetry route**
- **show router fib-telemetry tunnel**
- **show router fib-telemetry label**
- **show router rib-api route**
- **show router rib-api tunnel**
- **show router rib-api label**

The state information represented by the `state/router/route-fib` and `state/router/tunnel-fib` paths, and the **show router fib-telemetry route** and **show router fib-telemetry tunnel** show commands list are not collected by default, because it requires additional processing. In order for this state to be collected you must configure the **configure router fib-telemetry** command. If this command is not configured then these states are not collected at all, and telemetry subscriptions are not supported for any of the following paths:

- `/state/router/route-fib`
- `/state/router/tunnel-fib`
- `/state/router/label-fib`

It is not possible for a single telemetry subscription to include any of these three paths in addition to other state paths outside of this tree. This is because of the potential volume of information in the tables described in this chapter.

For gNMI telemetry subscriptions, the following restrictions should be noted:

- If a route, tunnel or MPLS label entry is modified, and it covered by an ON-CHANGE subscription to a state path enabled by **config>router>fib-telemetry**, the update replays the current values of the entire

entry (except for statistics), including values did not change from the last update. It is up to the client to compare the update to the previous one received if it needs to know the exact properties that changed.

- Subscriptions to list keys of state paths enabled by **fib-telemetry** are not supported.

5.11 Traffic statistics

A gRPC client can make a request for traffic statistics to be collected. Both ingress and egress statistics are available but not all types of entries support both.

Traffic statistics are expressed in number of packets and in octets and are provided without forwarding-class or QoS profile distinction.

The system provides capabilities to display or show, clear, or monitor statistics.

5.11.1 Ingress statistics

Only RIB-API MPLS tunnel table entries support ingress statistics. The counters are attached to the ILM entry that is formed when the RIB-API entry is programmed. When different RIB-API entries use the same ILM or label, then the traffic statistics for these RIB-API entries are identical. Traffic counters are kept until the ILM entry is removed. Because of a lack of resources, the system may not be able to allocate counters (statistic indexes) to an ILM. In this case, the system automatically retries until it succeeds.

5.11.2 Egress statistics

Egress statistics are supported for the three RIB-API tunnel tables (IPv4, IPv6, and MPLS). The counters are attached to the NHLFE of each next hop. Counters are effectively allocated by the system at the time that the instance is programmed in the data-path. Counters are maintained even if an instance is deprogrammed and values are not reset. This means that, if an instance is reprogrammed, traffic counting resumes at the point where it last stopped. Traffic counters are released and therefore traffic statistics are lost when the instance or entry is removed from the database.

No retry mechanism is available for egress statistics. The system maintains a state per next hop and per instance identifying whether allocation of statistic indexes is successful. If the system is not able to allocate all the needed indexes on a specified instance because of a lack of resources, then the user should disable egress statistics on that instance. This action frees enough statistic indexes and re-enables egress statistics on the needed entry. The selection of which other construct to release statistic indexes from is beyond the scope of this document.

6 Path Computation Element Protocol (PCEP)

This chapter provides information about the PCEP.

6.1 Introduction to the PCEP

The PCEP is one of several protocols that communicate between a Wide-Area Network (WAN) Software-Define Networking (SDN) controller and network elements.

The Nokia WAN SDN Controller is known as the Network Services Platform (NSP).

Figure 57: NSP architecture shows the architecture of the NSP.

The NSP implements a few components that provide service provisioning, automation, optimization, and element management functions for both IP and optical networks. The following is an overview of the NSP components. More details can be found in the *NSP Planning Guide*:

- **NSP cluster**

  The NSP cluster is the core component that hosts the common services (nspOS), as well as all the major NSP software applications. Among the applications hosted by the NSP cluster is the Model-driven Mediation (MDM), which provides mediation between model-driven NSP applications and Nokia or third-party network devices. The Workflow Manager (WFM) allows for the creation and execution of workflows. The NSP Baseline Analytics monitor network traffic to establish baselines and can flag anomalous traffic patterns.

- **IP resource control (IPRC)**

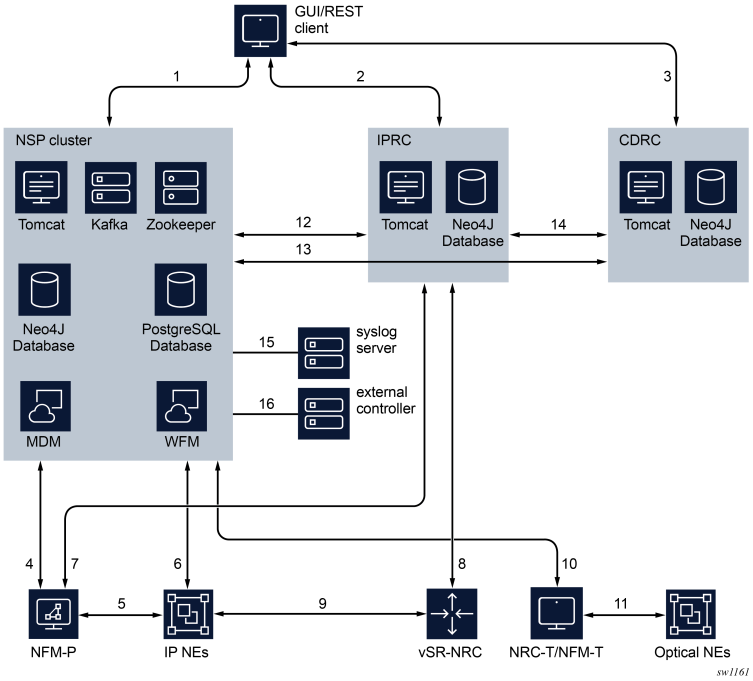
  The IPRC provides service provisioning and activation as well as the Network Resource Controller for packet networks (NRC-P). The NRC-P hosts a path computation engine and implements a stateful Path Computation Element (PCE). The PCE instantiates and manages LSPs across IP network elements (NEs), and supports RSVP and segment routing (SR) LSP technologies.

- **Cross domain resource control (CDRC)**

  This component optimizes network resources across different layers and domains of IP/MPLS, and optical networks.

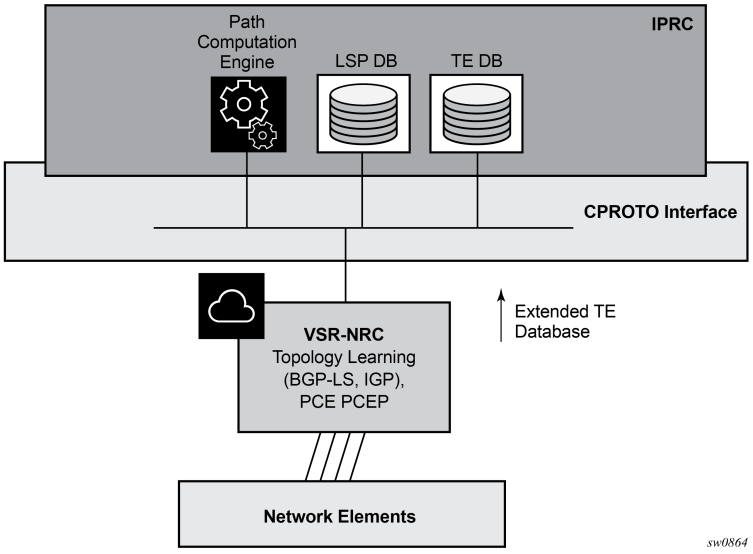
- **simulation tool**
  A traffic engineering (TE) tool that can be used by network engineers to design a new network, or optimize and simulate failures in an existing network that is imported into the tool.

Figure 57: NSP architecture



The following figure shows the NRC-P.

Figure 58: Packet network resource controller (NRC-P) architecture



The NRC-P has the following architecture:

- A single Virtual Machine (VM) handles the Java implementation of an MPLS path computation engine, a Traffic Engineering graph database (TE-DB), and an LSP database (LSP-DB). This is part of the IPRC as shown in Figure 57: NSP architecture.

- A single VM running an SR OS image handles the functions of topology discovery of multiple IGP instances and areas through IGP or BGP-LS and the PCE PCEP functions. This is referred to as the VSR Network Resource Controller (VSR-NRC).

  The VSR-NRC is a network function of the Virtual Services Router (VSR) and must be deployed as an integrated model (VSR-I). The VSR-I uses a single image for both the functions of a CPM and an IOM, which are referred to as CPMv and an IOMv respectively. See "VSR deployment model" in the *VSR VNF Installation and Setup Guide*.

  The VSR-NRC requires a special VSR license to unlock the SDN features specific to its deployment with the NRC-P.

- A plug-in adapter using the Nokia cproto interface, provides reliable, TCP-based message delivery between VSR-NRC and the IPRC. The plug-in adapter implements a compact encoding/decoding (codec) function for the message content using Google Protocol Buffers (protobuf). Google protobuf also provides for automatic C++ (VSR-NRC side) and Java (IPRC side) code generation to process the exchanged message content.

The VSR-NRC implements a PCEP PCE function, a BMP station database, a Route Origination Module (ROM), and a TE database populated using IGP and BGP-LS.

The NRC-P module of the NSP and the VSR-NRC communicate using a reliable proprietary TCP-based channel called cproto. The NRC-P module acts as the client side, it is the module which always initiates the establishment of the cproto session toward the VSR-NRC. This connection is established out-of-band using the management interface and can use the management interface IPv4 or IPv6 address as the local address.

The message data within the cproto channel is encoded and serialized using Google protobuf.

The VSR-NRC implements an NSP-Proxy module that manages all databases and channels used in the communications with the NRC-P. The NSP-Proxy opens a dedicated UDP port number 4199 for this communication and operates as the server side. This port is managed by the NSP-PROXY.

The NRC-P initiates a separate cproto session with a dedicated protobuf channel service for the information exchanged for each type of capability supported:

- PCEP – exchanges parameters and state information for RSVP-TE and SR-TE LSPs established using PCEP
- BGP_LS – passes topology discovered using BGP-LS and IGP
- BMP_STATION – passes per-BGP family route information learned using BGP or BMP peering with the network
- ROM_SRTE – ROM for BGP families **sr-policy-ipv4** and **sr-policy-ipv6**
- ROM_IPV4/V6 – ROM for BGP families IPv4 and IPv6
- ROM_LABELV4/V6 – ROM for BGP families **label-ipv4** and **label-ipv6**
- Global Health and Notification – exchanges channel health and notification messages that are not application specific

The NRC-P, via the VSR-NRC, uses PCEP to communicate with its clients, referred to as PCE Clients (PCCs). Each router acting as a PCC initiates a PCEP session to the PCE in its domain.

When the user enables PCE control for one or more segment routing or RSVP LSPs, the PCE owns the path updating and periodic reoptimization of the LSP. In this case, the PCE acts in an active stateful role. The PCE can also act in a stateful passive role for other LSPs on the router by discovering them and taking into account their resource consumption when computing the path for the LSPs it has control ownership of.

The following is a high-level description of the PCE and PCC capabilities:

- base PCEP implementation, in accordance with RFC 5440
- active and passive stateful PCE LSP update, in accordance with RFC 8231, *Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE*
- delegation of LSP control to PCE
- synchronization of the LSP database (LSP-DB) with network elements for PCE-controlled LSPs and network element-controlled LSPs
- support for the RSVP-TE P2P LSP type
- support for the SR-TE P2P LSP type, in accordance with RFC 8864
- support for PCC-initiated LSPs, in accordance with RFC 8231, *Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE*
- support for PCE-initiated LSPs, in accordance with RFC 8281, *PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model*
- support for LSP path diversity across different LERs using extensions to the PCE path profile, in accordance with *draft-alvarez-pce-path-profiles*
- support for LSP path bidirectionality constraints using extensions to the PCE path profile, in accordance with *draft-alvarez-pce-path-profiles*
- support for the PCEP ASSOCIATION object for SR-TE and RSVP-TE LSPs (RFC 8697), for signaling path diversity constraints (RFC 8800) and policy constraints (*draft-ietf-pce-association-policy-16*), and PCE-initiated SR-TE LSPs

6.1.1 PCC and PCE configuration

The following PCE parameters cannot be modified while the PCEP session is operational:

- **local-address**
- **local-address-ipv6**
- **keepalive**
- **dead-timer**

The **unknown-message-rate** PCE parameter can be modified while the PCEP session is operational.

The following PCC parameters cannot be modified while the PCEP session is operational:

- **local-address**
- **local-address-ipv6**
- **keepalive**
- **dead-timer**
- **peer** (regardless of the shutdown state in classic CLI or the administrative state in the MD-CLI)

The following PCC parameters can be modified while the PCEP session is operational:

- **report-path-constraints**
- **unknown-message-rate**

6.1.2 Base implementation of PCE

The base implementation of PCE uses the PCEP extensions defined in RFC 5440.

The main functions of the PCEP are:

- establishing, maintaining, and closing PCEP sessions
- generating path computation requests using the PCReq message
- generating path computation replies using the PCRep message
- generating notification messages (PCNtf) by which the PCEP speaker can inform its peer about events, such as path request cancellation by the PCC or path computation cancellation by the PCE
- generating error messages (PCErr) by which the PCEP speaker can inform its peer about errors related to processing requests, message objects, or TLVs

The following table lists the base PCEP messages and objects.

Table 34: Base PCEP message objects and TLVs

TLV, object, or message	Contained in object	Contained in message
OPEN object	—	OPEN, PCErr
Request Parameter (RP) object	—	PCReq, PCRep, PCErr, PCNtf
NO-PATH object	—	PCRep
END-POINTS object	—	PCReq
BANDWIDTH object	—	PCReq, PCRep, PCRpt, PCInitiate
METRIC object	—	PCReq, PCRep, PCRpt, PCInitiate
Explicit Route Object (ERO)	—	PCRep
Reported Route Object (RRO)	—	PCReq
LSPA object	—	PCReq, PCRep, PCRpt, PCInitiate
NOTIFICATION object	—	PCNtf
PCEP-ERROR object	—	PCErr
CLOSE object	—	CLOSE
ASSOCIATION object	–	PCReq, PCRpt, PCRep, PCUpd, PCInitiate

The behavior and limitations of the implementation of the objects in the preceding table are as follows:

- The PCE treats all supported objects received in a PCReq message as mandatory, regardless of whether the P-flag in the common header of the object is set (mandatory object) or not (optional object).
- The PCC implementation always sets the B-flag (B=1) in the METRIC object containing the hop metric value, which means that a bound value must be included in the PCReq message. The PCE returns the computed value in the PCRep message with flags set identically to the PCReq message.
- The PCC implementation always sets flags B=0 and C=1 in the METRIC object for the IGP or TE metric values in the PCReq message. This means that the request is to optimize (minimize) the metric without providing a bound. PCE returns the computed value in PCRep message with flags set identically to the PCReq message.
- The IRO and LOAD-BALANCING objects are not supported in the NSP PCE feature. If the PCE receives a PCReq message with one or more of these objects, it ignores them regardless of the setting of the P-flag and processes the path computations normally.
- LSP path setup and hold priorities are configurable during SR-TE LSP configuration on the router, and the PCC passes the configurations on in an LSPA object. However, the PCE does not implement LSP preemption.
- The LSPA, METRIC, and BANDWIDTH objects are also included in the PCRpt message.

The following features are not supported in SR OS:

- PCE discovery using IS-IS (as defined in RFC 5089) and OSPF (as defined in RFC 5088) along with corresponding extensions for discovering stateful PCE (as defined in *draft-sivabalan-pce-disco-stateful*)
- PCEP synchronization optimization (as defined in RFC 8232)
- jitter, latency, or packet loss link metric signaling in the PCE METRIC object (as defined in RFC 8233)

6.1.3 PCEP session establishment and maintenance

The PCEP protocol operates over TCP using destination TCP port 4189. The PCE client (PCC) always initiates the connection. After the user configures the PCEP local IPv4 or IPv6 address and the peer IPv4 or IPv6 address on the PCC, the PCC initiates a TCP connection to the PCE. If both a local IPv4 and a local IPv6 address are configured, the connection uses the local address of the same family as the peer address. When the connection is established, the PCC and PCE exchange OPEN messages, and this process initializes the PCEP session and exchanges the session parameters to be negotiated.

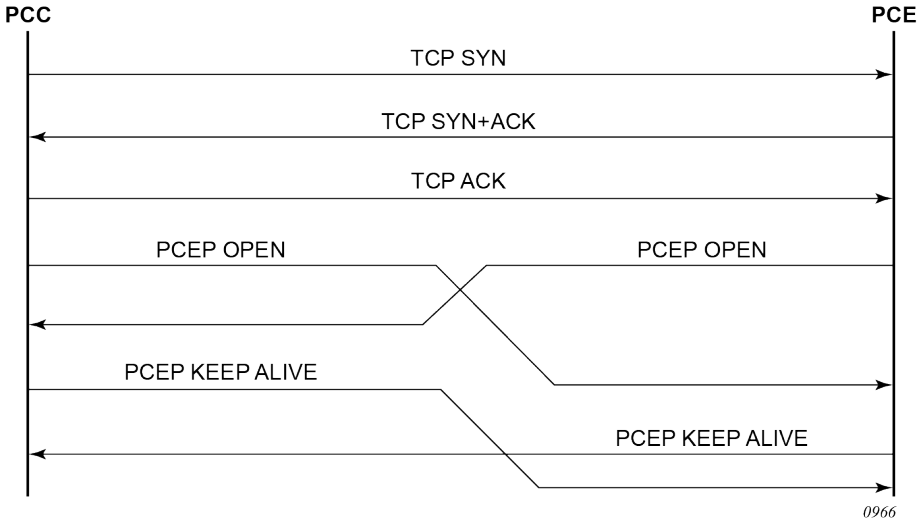
By default, the PCC attempts to reach the remote PCE address out of band using the management port. If it cannot, it attempts to reach the remote PCE address in band. The user can modify the configuration of the peer to attempt connecting in band only or out of band only. When the session comes up out of band, the management IP address is used as the local address. The local IPv4 or IPv6 address configured by the user is only used for in-band sessions and is otherwise ignored.

A keepalive mechanism is used as an acknowledgment of the acceptance of the session within the negotiated parameters. It is also used as a maintenance function to detect whether the PCEP peer is still alive.

The negotiated parameters include the keepalive timer and the dead timer, and one or more PCEP capabilities such as support of stateful PCE and the SR-TE LSP path type.

The following figure shows PCEP session initialization steps.

Figure 59: PCEP session initialization



If the session to the PCE times out, the router acting as a PCC keeps the last successfully-programmed path provided by the PCE until the session to the PCE is reestablished. Any subsequent change to the state of an LSP is synchronized at the time the session is reestablished.

When a PCEP session to a peer times out or closes, the rate at which the PCEP speaker attempts the establishment of the session is subject to an exponential back-off mechanism.

6.1.4 PCEP parameters

The following PCEP parameters are configurable on both the PCC and PCE. On the PCE, the configured parameter values are used on sessions to all PCCs.

- **keepalive timer**

  A PCEP speaker (PCC or PCE) must send a keepalive message if no other PCEP message is sent to the peer at the expiry of this timer. This timer is restarted every time a PCEP message is sent or the keepalive message is sent.

  The keepalive mechanism is asymmetric, meaning that each peer can use a different **keepalive** timer value.

  The range of this parameter is 1 to 255 seconds, and the default value is 30 seconds. The **no** version reverts to the default value.

- **dead timer**

  This timer tracks the amount of time a PCEP speaker (PCC or PCE) waits after the receipt of the last PCEP message before declaring its peer down.

  The dead timer mechanism is asymmetric, meaning that each PCEP speaker can propose a different dead timer value to its peer to use to detect session timeouts.

  The range of this parameter is 1 to 255 seconds, and the default value is 120 seconds. The **no** version reverts to the default value.

- **maximum rate of unknown messages**

When the rate of received unrecognized or unknown messages reaches this limit, the PCEP speaker closes the session to the peer.

- **session redelegation and state timer**

  If the PCEP session to the PCE goes down, all delegated PCC-initiated LSPs have their state maintained in the PCC and are not timed out. The PCC continues to attempt to reestablish the PCEP session. When the PCEP session is reestablished, the LSP database is synchronized with the PCE, and any LSP that went down after the last time the PCEP session was up has its path updated by the PCE.

6.1.5 Stateful PCE

The main function of stateful PCE, as opposed to the base PCE implementation, is the ability to synchronize the LSP state between the PCC and the PCE. This function allows the PCE to have all the required LSP information to perform reoptimization and updating of the LSP paths.

The following table describes the messages and objects supported by stateful PCE in the SR OS.

Table 35: PCEP stateful PCE extension objects and TLVs

TLV, object, or message	Contained in object	Contained in message
Path Computation State Report (PCRpt) message	—	New message
Path Computation Update Request (PCUpd) message	—	New message
Stateful PCE Capability TLV	OPEN	OPEN
Stateful Request Parameter (SRP) object	—	PCRpt, PCErr, PCInitiate
LSP object	ERO	PCRpt, PCReq, PCRep, PCInitiate
LSP Identifiers TLV	LSP	PCRpt
Symbolic Path Name TLV	LSP, SRP	PCRpt, PCInitiate
LSP Error Code TLV	LSP	PCRpt
RSVP Error Spec TLV	LSP	PCRpt
ASSOCIATION object	—	PCRpt, PCReq, PCRep, PCInitiate, PCUpd

The following behavior and limitations apply to the implementation of the objects listed in the preceding table:

- PCC and PCE support all PCEP capability TLVs defined in this document and always advertise them. If the OPEN object received from a PCEP speaker does not contain one or more of the capabilities, neither the PCE nor the PCC use them during the specific PCEP session.

- The PCC always includes the LSP object in the PCReq message to ensure that the PCE can correlate the PLSP ID for this LSP when a subsequent PCRpt message arrives with the delegation bit set. The PCE, however, still honors a PCReq message without the LSP object.
- PCE path computation only considers the bandwidth used by LSPs in its LSP-DB. As a result, in the following situations, the PCE path computation does not accurately account for the bandwidth used in the network:
  - When there are LSPs that are signaled by the routers but are not synchronized with the PCE. The user can enable the reporting of the LSP to the PCE LSP database for each LSP.
  - When the stateful PCE is peering with a third-party stateless PCC, implementing only the original RFC 5440. While the PCE is able to bring the PCEP session up, the LSP database is not updated, because stateless PCC does not support the PCRpt message. As such, PCE path computation does not accurately account for the bandwidth used by these LSPs in the network.
- The PCE ignores the reoptimize flag (R-flag) in the PCReq message when acting in stateful-passive mode for a specific LSP and always returns the new computed path, regardless of whether it is link-by-link identical or has the same metric as the current path. The PCC decides whether to initiate the new path in the network.
- The SVEC object is not supported in SR OS nor in the NSP. If the PCE receives a PCReq message with the SVEC object, it ignores the SVEC object and treats each path computation request in the PCReq message as independent, regardless of the setting of the P-flag in the SVEC object common header.
- When an LSP is delegated to the PCE, there can be no prior state in the NRC-P LSP database for the LSP. This could be because the PCE did not receive a PCReq message for the same PLSP-ID. For the PCE to become aware of the original constraints of the LSP, the following additional procedures are performed:
  - The PCC appends a duplicate of each of the LSPA, METRIC, and BANDWIDTH objects in the PCRpt message. The only difference between the two objects of the same type is that the P-flag is set in the common header of the duplicate object to indicate a mandatory object for processing by the PCE.
  - The value of the metric or bandwidth in the duplicate object contains the original constraint value, while the first object contains the operational value. This is applicable to hop metrics in the METRIC object and BANDWIDTH object only. The SR OS PCC does not support putting a bound on the IGP or TE metric in the path computation.
  - The path computation on the PCE uses the first set of objects when updating a path, if the PCRpt message contains a single set. If the PCRpt message contains a duplicate set, the PCE path computation must use the constraints in the duplicate set.
  - For interoperability, implementations compliant to PCEP standards accept the first metric object and ignore the second object without additional error handling. Because there are also BANDWIDTH and LSPA objects, the **report-path-constraints** command is provided in the PCC on a per-PCEP session basis to disable the inclusion of the duplicate objects. Duplicate objects are included by default.

The following table lists the additional messages, TLVs, and objects used by stateful PCE for PCE initiation of LSPs.

Table 36: PCEP stateful PCE extension objects and TLVs locations

TLV, object, or message	Contained in object	Contained in message
PCE LSP Initiate Message (PCInitiate)	N/A	New message
PCC LSP Create Flag (C-Flag)	LSP	PCRpt
PATH_PROFILE_ID TLV	Path Profile	N/A

6.1.6 PCEP extensions in support of SR-TE LSPs

To manage the path of an SR-TE LSP, the PCE and PCC both implement the following extensions to PCEP in support of segment routing:

- An SR-PCE-CAPABILITY in the OPEN object to indicate support of segment routing tunnels by the PCE and the PCC during PCEP session initialization.

  The PCEP speaker on the transmit side encodes the SR-PCE-CAPABILITY as both a top-level TLV in the OPEN object and as a sub-TLV of the PATH SETUP TYPE CAPABILITY TLV in the OPEN object. The PCEP speaker on the receive side processes the SR-PCE-CAPABILITY, which it implements at the TLV level and ignores the sub-TLV. The PCEP speaker performs the following actions:

  – A PCEP speaker that implements RFC 8664 and which receives both the top-level SR-PCE-CAPABILITY TLV and the top-level PATH SETUP TYPE CAPABILITY TLV processes the SR-PCE-CAPABILITY sub-TLV in the top-level PATH SETUP TYPE CAPABILITY TLV.

  – A PCEP speaker that implements RFC 8664 and which receives only the top-level SR-PCE-CAPABILITY TLV treats it as if it received the top-level PATH SETUP TYPE CAPABILITY TLV with PST={0,1}, meaning both RSVP-TE and SE-TE LSP types are supported, along with the SR-PCE-CAPABILITY sub-TLV.

  – A PCEP speaker that implements up to version *RFC 8664* and which receives both the top-level SR-PCE-CAPABILITY TLV and the top-level PATH SETUP TYPE CAPABILITY TLV processes the top-level SR-PCE-CAPABILITY and ignores the top-level PATH SETUP TYPE CAPABILITY TLV.

  This implementation is in accordance with the backward compatibility procedure defined in RFC 8664 to process the SR-PCE-CAPABILITY.

  If the OPEN object is received from a PCEP speaker without the SR-PCE-CAPABILITY in either or both of the preceding encodings, the PCE or the PCC does not send or accept PCEP messages for LSP paths of type SR-TE on that specific PCEP session.

- A Path Setup Type TLV for SR-TE LSPs to be included in the Stateful PCE Request Parameters (SRP) object during the path report (PCRpt) messages by the PCC.

  A Path Setup Type TLV with a value of 1 identifies an SR-TE LSP.

- A Segment Routing ERO and an RRO with subobjects, referred to as SR-ERO and SR-RRO subobjects, which encode the SID information in the PCRpt messages.

- The PCE implementation supports the Segment-ID (SID) Depth value in the METRIC object. This is always signaled by the PCC in the PCEP OPEN object as part of the as SR-PCE-CAPABILITY TLV. It is referred to as the Maximum Stack Depth (MSD). In addition, the per-LSP value for the **max-sr-labels** option, if configured, is signaled by the PCC to the PCE in the Segment-ID (SID) Depth value in a METRIC object for both a PCE-computed LSP and a PCE-controlled LSP. PCE computes and provides

the full explicit path with the TE-links specified. If there is no path with the number of hops lower than the MSD value, or the Segment-ID (SID) Depth value if signaled, a reply with no path is returned to the PCC. For a PCC-controlled LSP, if the label stack returned by the TE-DB's hop-to-label translation exceeds the per-LSP maximum SR label stack size, the LSP is brought down.

- If the Path Setup Type (PST) TLV is not included in the PCReq message, the PCE or PCC must assume the message is for an RSVP-TE LSP.

The following table describes the segment routing extension objects and TLVs supported in the SR OS.

Table 37: PCEP segment routing extension objects and TLVs

TLV, object, or message	Contained in object	Contained in message
SR-PCE-CAPABILITY TLV	OPEN (both as top-level TLV and sub-TLV of PATH SETUP TYPE CAPABILITY TLV)	OPEN
PATH SETUP TYPE CAPABILITY TLV	OPEN	OPEN
SR-ERO Sub-object	ERO	PCRep, PCRpt
SR-RRO Sub-object	RRO	PCReq, PCRpt
Segment-ID (SID) Depth Value in METRIC object	METRIC	PCReq, PCRpt

6.1.7 PCEP security

SR OS supports the following methods for PCEP security, as described in RFC 8253, *PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)*:

- **TCP-AO**

  With TCP-AO, the channel is only authenticated, and not encrypted. PCEP speaking parties are also authenticated.

- **TLS**

  With TLS, the channel is encrypted and authenticated, and the PCEP speaking parties are also authenticated.

In SR OS, both TCP-AO and TLS can be used individually or simultaneously for securing the PCEP channel.

6.1.7.1 PCEP over TCP-AO

In accordance with RFC 5925, *The TCP Authentication Option*, TCP-AO uses AES-CMAC and HMAC-SHA-1-96 for channel authentication. TCP-AO only provides authentication of the packets, and not encryption. SR OS supports the use of TCP-AO in the keychain mechanism.

When a keychain is specified for PCEP, it uses the keychain algorithm and key to authenticate the channel for the PCEP packets via TCP-AO.

Use the commands in the following context to configure a keychain:

- **MD-CLI**

```
configure system security keychains
```

- **classic CLI**

```
configure system security keychain
```

Use the following commands to specify a keychain for use by the PCEP:

- **MD-CLI**

```
configure router pcep pcc peer authentication-keychain
```

```
configure router pcep pce authentication-keychain
```

- **classic CLI**

```
configure router pcep pcc peer auth-keychain
```

```
configure router pcep pce auth-keychain
```

6.1.7.2 PCEP over TLS

PCEP over TLS (PCEPS) is secured using TLS on port 4189. The PCC is configured with a TLS client profile to initiate the TLS handshake. The PCE is configured with a TLS server profile to allow PCEPS. When a TLS server profile is configured on the PCE, the PCE can establish TLS and non-TLS connections, in PCE secured (PCES) and PCE modes. See PCE behavior for more information about the modes supported by SR OS.

In TLS mode, both the PCC and PCE must provide certificates for authentication. The PCE provides the server certificate to the PCC and requires the client certificate to authenticate the PCC.

6.1.7.2.1 TLS handshake

SR OS supports TLS client (PCC) and server (PCE) functionality, and TLS bidirectional authentication, where the PCE requests the client certificate to authenticate the PCC.

In a typical TLS handshake, the client starts the handshake with a ClientHello message. The server provides the server certificate for authentication to the client and sends a list of server-accepted ciphers.

The server can optionally ask the client to provide the client certificate using the server CertificateRequest option. When this option is present, the client provides the server with the client certificate and, if authenticated, the TLS symmetric key is negotiated and the TLS session is established. The symmetric key is used to encrypt the TLS datapath.

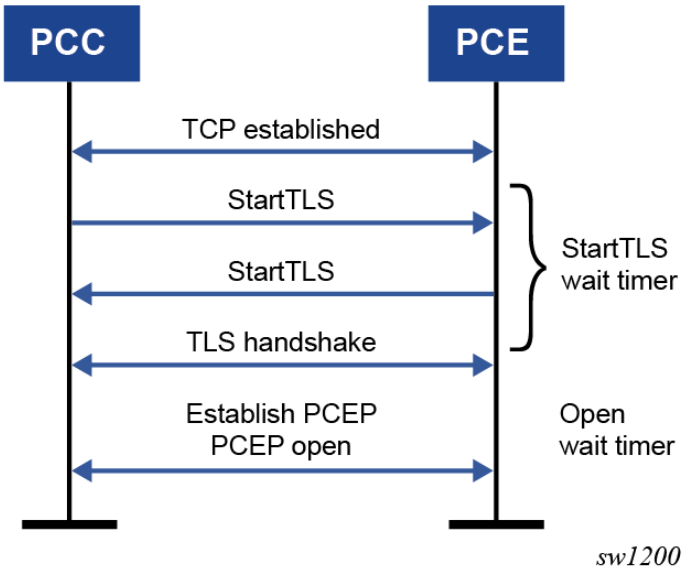
See the *7450 ESS, 7750 SR, 7950 XRS, and VSR System Management Guide* for more information about the TLS handshake steps.

6.1.7.2.2 PCEP session over TLS

To establish a PCEP session over TLS as specified in RFC 8253, the PCC sends a StartTLS message to the PCE to initiate the TLS negotiation. The PCC activates the StartTLS timer and waits for the StartTLS message from the PCE. The timer is configured using the **configure router pcep pcc peer tls-wait-timer** command; the default timer is 60 seconds.

If the PCE is TLS-capable and sends back a StartTLS message before the StartTLS timer expires, the TLS handshake is initiated. If the PCE sends an Open message or does not send back a StartTLS message, the PCC responds with an error message, closes the TCP connection, and retries to establish the connection. The PCEP Message-Type field of the PCEP common header for the StartTLS message is set to 13, as specified in RFC 8253. The following figure shows the establishment of a PCEP session over TLS.

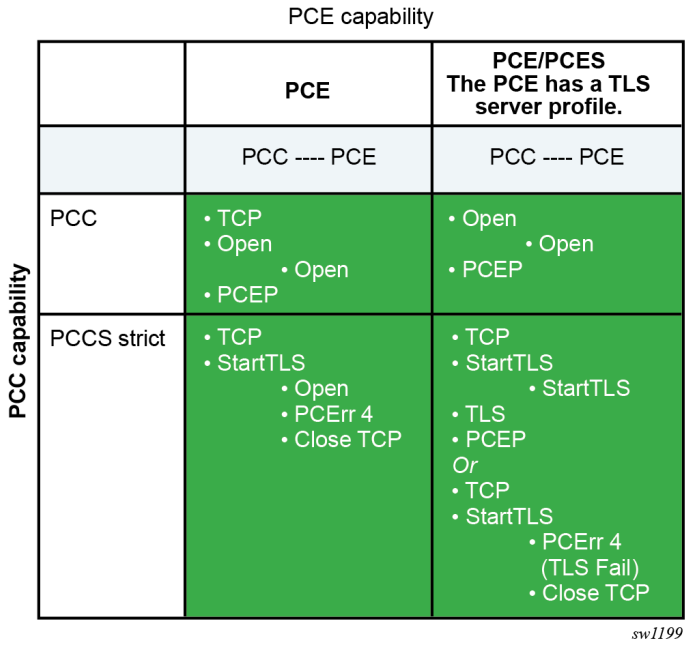
Figure 60: PCEP session over TLS



TLS supports both in-band and out-of-band PCE connections. The following figure summarizes the PCE and PCC TLS support.

Note: SR OS does not support PCES strict mode.

Figure 61: PCE and PCC TLS support



Note:
PCEP over TLS does not support CPM redundancy. After an activity switch, the PCEP over TLS connection goes down and is re-established.

6.1.7.2.3 PCC behavior

On the PCC, SR OS supports only strict TLS. That is, both PCE and PCC must support TLS and perform a successful TLS handshake before the TLS wait timer expires. Otherwise, the PCC retries the connection after 60 seconds.

Use the following commands to configure the PCC:

- **configure router pcep pcc peer tls-client-profile**
- **configure router pcep pcc peer tls-wait-timer**

6.1.7.2.4 PCE behavior

On the PCE, SR OS supports TLS or non-TLS mode. That is, when a TLS profile is configured on the PCE, the PCE accepts PCC connections that are TLS-secure or unsecured. To configure the TLS profile on the PCE, use the **configure router pcep pce tls-server-profile** command.

In the PCES and PCE mode, SR OS accepts connections with a StartTLS message or an Open message from the PCC. Depending on the PCC that sends the StartTLS message, the PCE sends back a StartTLS message also.

In the PCE-only mode, SR OS accepts only Open messages from the PCC; StartTLS messages are not accepted.

In the PCES strict mode, the PCE accepts only TLS connections from the PCC. Non-TLS connections (which open PCEP connections with Open message, not with StartTLS message) are not accepted and the TCP connection is closed. SR OS does not support PCES strict mode.

6.2 PCEP establishment and maintenance of SR-TE LSP and RSVP-TE LSP

This section describes PCEP establishment and maintenance of SR-TE LSP and RSVP-TE LSP.

6.2.1 LSP initiation

An LSP that is configured on the router is called a PCC-initiated LSP. An LSP that is not configured on the router, but is instead created by the PCE at the request of an application or a service instantiation, is called a PCE-initiated LSP.

The SR OS supports the following modes of operations for PCC-initiated LSPs that are configurable on a per-LSP basis:

- When the path of the LSP is computed and updated by the router acting as a PCE Client (PCC), the LSP is called a PCC-initiated and PCC-controlled LSP.

  A PCC-initiated and PCC-controlled LSP has the following characteristics.

  – The LSP can contain strict or loose hops, or a combination of both.

  – CSPF is supported for RSVP-TE LSPs. Local path computation takes the form of hop-to-label translation for SR-TE LSPs.

  – LSPs can be reported to synchronize the LSP database of a stateful PCE server using the **pce-report** command. In this case, the PCE acts in passive stateful mode for the LSP. The LSP path cannot be updated by the PCE, which means that control of the LSP is maintained by the PCC.

- When the path of the LSP is computed by the PCE at the request of the PCC, it is called a PCC-initiated and PCE-computed LSP.

  A PCC-initiated and PCE-computed LSP has the following characteristics.

  – The user must enable the **path-computation-method pce** option for the LSP so that the PCE can perform path computations at the request of the PCC only. The PCC retains control.

  – LSPs can be reported to synchronize the LSP database of a stateful PCE server using the **pce-report** command. In this case, the PCE acts in passive stateful mode for the LSP.

- When the path of the LSP is updated by the PCE following a delegation from the PCC, it is called PCC-initiated and PCE-controlled.

  A PCC-initiated and PCE-controlled LSP has the following characteristics.

  – The user must enable the **pce-control** command for the LSP so that the PCE can perform path updates after a network event without an explicit request from the PCC. The PCC delegates full control.

  – The user must enable the **pce-report** command for LSPs that cannot be delegated to the PCE. The PCE acts in active stateful mode for the LSP.

SR OS also supports PCE-initiated LSPs. PCE-initiated LSPs allows a WAN SDN Controller(such as the NSP) to automatically instantiate an LSP based on a service or application request. Only SR-TE PCE-initiated LSPs are supported.

The instantiated LSP does not have a configuration on the network routers and, consequently, is treated in the same way as an auto-LSP. The parameters of the LSP are provided using policy lookup in the NSP and are passed to the PCC using PCEP, in accordance with RFC 8281. Missing LSP parameters are added using a default or specified LSP template on the PCC.

PCE-initiated LSPs have the following characteristics.

- The user must enable the **pce-initiated-lsp sr-te** command to enable the PCC to accept and process PCInitiate messages from the PCE.
- The user must configure one or more LSP templates of type **pce-init-p2p-srte** for SR-TE LSPs. A default template is supported that is used for LSPs for which no ID or an ID of 0 is included in the PCInitiate message. The user must configure at least one default PCE-initiated LSP template.

PCE-initiated LSPs are a form of SR-TE auto-LSP and are available to the same forwarding contexts. See Forwarding contexts supported with SR-TE auto-LSP. Similar to other auto-LSPs, PCE-initiated LSPs are installed in the TTM and are therefore available to advanced policy-based services using auto-bind, such as VPRN and E-VPN. However, PCE-initiated LSPs cannot be used with provisioned SDPs.

6.2.1.1 Configuring PCC-initiated and PCE-computed or PCE-controlled LSPs

For the procedure described in this section, the user performs a **no shutdown** command on a PCE-controlled or PCE-computed LSP. The starting point is an LSP that is administratively down with no active path.

The router performs the following steps to configure and program a PCC-initiated SR-TE LSP when control is delegated to the PCE. The sequence assumes that an LSP configuration has been created on the PE router using the CLI or using the OSS/NSP NFM-P.

Note:
The LSP configuration dictates the needed PCE control mode: active (**pce-control** and **pce-report** commands enabled) or passive (**path-computation-method pce** option enabled and **pce-control** command disabled).

1. The PCC assigns a unique PLSP-ID to the LSP. The PLSP-ID identifies the LSP on a PCEP session and must remain constant during its lifetime. The PCC on the router must keep track of the association of the PLSP-ID to the tunnel ID and path ID, and use the latter to communicate with MPLS about a specific path of the LSP. The PCC also uses the SRP-ID to correlate PCRpt messages for each new path of the LSP.
2. The PE router does not validate the entered path; however, in SR OS, the PCE supports the computation of a path for an LSP with empty-hops in its path definition. Although the PCC includes the IRO objects in the PCReq message to PCE, the PCE ignores them and computes the path with the other constraints, except the IRO.
3. The PE router sends a PCReq message to the PCE to request a path for the LSP and includes the LSP parameters in the METRIC object, the LSPA object, and the BANDWIDTH object. The PE router also includes the LSP object with the assigned PLSP-ID. At this point, the PCC does not delegate control of the LSP to the PCE.
4. The PCE computes a new path, reserves the bandwidth, and returns the path in a PCRep message with the computed ERO in the ERO object. The PCRpt message also includes the LSP object with the unique PLSP-ID, the METRIC object with any computed metric value, and the BANDWIDTH object.

Note:
For the PCE to use the SRLG path diversity and admin-group constraints in the path computation, the user must configure the SRLG and admin-group membership against the MPLS interface and make sure that the **traffic-engineering** command is enabled in IGP. This causes IGP to flood the link SRLG and admin-group membership in its participating area, and for PCE to learn it in its TE database.

5. The PE router updates the CPM and the datapath with the new path.

   Up to this point, the PCC and PCE are using passive stateful PCE procedures. The subsequent steps synchronize the LSP database of the PCC and PCE for both PCE-computed and PCE-controlled LSPs, and also initiate the active PCE stateful procedures for the PCE-controlled LSP only.
6. The PE router sends a PCRpt message to update the PCE with an UP state, and also sends the RRO as confirmation. It now includes the LSP object with the unique PLSP-ID. For a PCE-controlled LSP, the PE router also sets the delegation control flag to delegate control to the PCE. The state of the LSP is now synchronized between the router and the PCE.
7. Following a network event or a reoptimization, the PCE computes a new path for a PCE-controlled LSP and returns it in a PCUpd message with the new ERO. The PCUpd message includes the LSP object with the same unique PLSP-ID assigned by the PCC, as well as the Stateful Request Parameter (SRP) object with a unique SRP-ID number to track error and state messages specific to this new path.
8. The PE router updates the CPM and the datapath with the new path, then sends a PCRpt message to inform the PCE that the older path is deleted. The PCRpt message includes the unique PLSP-ID value in the LSP object and the Remove (R) bit set.
9. The PE router sends a new PCRpt message to update PCE with an UP state, and also sends the RRO to confirm the new path. The state of the LSP is now synchronized between the router and the PCE.
10. If PCE owns the delegation of the LSP and is making a path update, MPLS initiates the LSP and updates the operational value of the changed parameters, while the configured administrative values do not change. Both the administrative and operational values are shown in the details of the LSP path in MPLS.
11. If a configuration change is made to the PCE-computed or PCE-controlled LSP, MPLS requests that the PCC first revoke delegation in a PCRpt message (PCE-controlled only), and the MPLS and PCC follow the preceding steps to convey the changed constraint to the PCE, which results in the programming of a new path into the datapath, the synchronization of the PCC and PCE LSP databases, and the return of delegation to PCE.

For an LSP with an active path, the following items apply.

- If the user enabled the **path-computation-method pce** option on a PCC-controlled LSP with an active path, no action is performed until the next time the router needs a path for the LSP following a network event of an LSP parameter change. At this point, the preceding procedure is followed.
- If the user enabled the **pce-control** option on a PCC-controlled or PCE-computed LSP with an active path, the PCC issues a PCRpt message to the PCE with an UP state, as well as the RRO of the active path. It sets the delegation control flag to delegate control to the PCE. The PCE keeps the active path of the LSP and makes no updates to it until the next network event or reoptimization. At this point, the preceding procedure is followed.

6.2.1.2 Configuring PCE-initiated LSPs

About this task

Perform the following steps to configure and program a PCE-initiated SR-TE LSP.

Procedure

Step 1. Enable the **configure**>**router**>**mpls**>**pce-initiated-lsp sr-te** command using the CLI or the OSS.

Optionally, configure a limit to the number of PCE-initiated LSPs that the PCE can instantiate on a node using the **max-srte-pce-init-lsps** command in the CLI or using the OSS.

Step 2. Use the **configure router mpls lsp-template** *template-name* **pce-init-p2p-srte** *template-id* [**default** | *template-id*] command to configure at least one LSP template of the **pce-init-p2p-srte** type to select the value of the LSP parameters that remain under the control of the PCC. At a minimum, a default template should be configured (**pce-init-p2p-srte default** type).

In addition, LSP templates with defined template IDs can be configured. The template ID can be included in the path profile of the PCInitiate message to indicate which non-default template to use for a specific LSP. If the PCInitiate message does not include the PCE path profile, MPLS uses the default PCE-initiated LSP template. Table 38: LSP template parameters lists the applicable LSP template parameters. These parameters fall into the following groups:

- parameters that are controlled by the PCE and that the PCC cannot change (invalid, implicit, and signaled in PCEP)
- parameters that are controlled by the PCC and are used for signaling the LSP in the control plane
- parameters that are controlled by the PCC and are related to the usability of the LSP by MPLS and other applications, such as routing protocols, services, and OAM

Table 38: LSP template parameters

Controlled by PCE			Controlled by PCC	
Invalid	**Implicit**	**Signaled in PCEP**	**LSP signaling options**	**LSP usability options**
auto-bandwidth	pce-report	bandwidth	—	—
retry-limit	—	exclude	—	bgp-shortcut
retry-timer	pce-control	from	—	bgp-transport-tunnel
shutdown	pce-report	hop-limit	default-path (mandatory, must be empty)	—
least-fill	path-computation-method pce	include	—	—
metric-type	—	—	—	entropy-label
—	—	—	—	override-tunnel-elc

Controlled by PCE			Controlled by PCC	
Invalid	**Implicit**	**Signaled in PCEP**	**LSP signaling options**	**LSP usability options**
—	—	setup-priority	—	igp-shortcut
—	—	hold-priority	—	
—	—	—	—	load-balancing-weight
—	—	—	—	max-sr-labels
—	—	—	—	additional-frr-labels
—	—	—	—	metric
—	—	—	—	vprn-auto-bind
—	—	—	—	admin-tag

All PCE-initiated LSPs that use a particular LSP template are deleted if the user deletes the template. The default template can be created or deleted if the **pce-initiated-lsp>sr-te** context does not exist. However, the **pce-init-p2p-sr-te default lsp-template** cannot be deleted if the **pce-initiated-lsp>sr-te** context exists and is not shutdown. This context must be shutdown to delete the **pce-init-p2p-sr-te default** LSP template, which brings down all PCE-initiated LSPs. The **pce-initiated-lsp>sr-te** context cannot be administratively enabled if the **pce-init-p2p-sr-te default lsp-template** is not configured.

A shutdown of an LSP template does not bring down already established LSPs. Parameters can be changed only in the shutdown state, and the changes do not take effect until a **no shutdown** is performed. This means that PCE updates use older parameters if the template is still shut down.

MPLS copies the lsp-template parameters into the lsp-entry when a PCE initiated LSP is created. MPLS handles lsp-updates based on the last copied parameters.

After the lsp-template parameter changes, when the lsp-template is **no shutdown**.

- MPLS copies the related TTM parameters (listed below) into the LSP entry, and updates TTM
- If there is a change in **max-sr-labels**, MPLS reevaluates the related LSPs, and brings paths down if applicable (for example, if current hopCount is greater than the applicable **max-sr-labels** value).

  The TTM LSP-related parameters include:

  – Metric

  – VprnAutoBind

  – LoadBalWeight

  – MaxSrLabels

  – AdditionalFrrLabels

  – MetricOffset

- IgpShortCut
- IgpShortcutLfaOnly
- IgpShortcutLfaProtect
- LspBgpShortCut
- LspBgpTransTunnel

A PCE-initiated LSP update request is accepted regardless of the LSP template administrative state, as follows.

- If the LSP template is administratively up, the system copies the LSP template parameters to the LSP/path.
- If the LSP template is administratively down, the system uses the previously copied LSP template parameters and responds to the update with an LSP down report.

Step 3. Use the **configure router pcep pcc [no] redelegation-timer** *seconds* command to configure the redelegation timer on the PCC.

Redelegation timers (known as the Redelegation Timeout Interval in RFC 8231) are started when the PCEP session goes down or the PCE signals overload. The redelegation timer applies to both PCC-initiated and PCE-initiated LSPs.

If the delegated PCE-initiated LSPs cannot be redelegated by the time the timers expire, a configurable action is performed by the PCC. The supported actions are **remove** or **none**, with a default of **remove**.

Step 4. Use the **configure router pcep pcc [no] state-timer** *seconds* **[action {remove | none}]** command to configure the state timer on the PCC.

State timers (known as the State Timeout Interval in RFC 8231) are started when the PCEP session goes down or the PCE signals overload. The state timer applies only to PCE-initiated LSPs.

If the delegated PCE-initiated LSPs cannot be redelegated by the time the timers expire, a configurable action is performed by the PCC. The supported actions are **remove** or **none**, with a default of **remove**.

Expected outcome

After configuration, the PCE can initiate and remove LSPs on the PCC. See LSP instantiation using PCEP, LSP deletion using PCEP, and Dynamic state handling for PCE-initiated LSPs for more information about these procedures.

6.2.1.2.1 LSP instantiation using PCEP

This section describes the procedures followed in the instantiation of a PCE-initiated LSP by both the NSP and SR OS router. See *RFC 8281* for additional protocol details.

6.2.1.2.1.1 NSP generation of PCInitiate

1. When the PCEP session is established from the PCC to PCE, the PCC and PCE exchange the OPEN object and both set the "I flag, LSP-INSTANTIATION CAPABILITY" flag, in the STATEFUL-PCE-CAPABILITY TLV flag field.

2. The user, using the north-bound REST interface, the NSD, or another interface, requests the NSP NFM-P to initiate an LSP, specifying the following parameters:
   - source address
   - destination address
   - LSP type (SR-TE)
   - bandwidth value
   - include/exclude admin-group constraints
   - optional PCE path profile ID for the path computation at the PCE
   - optional PCE-initiated LSP template ID for use by the PCC to complete the instantiation of the LSP
3. The NSP NFM-P crafts the PCInitiate message and sends it to the PCC using PCEP. The message contains the following:
   - LSP object with PLSP-ID=0
   - SRP object
   - ENDPOINTS object
   - computed SR-ERO (SR-TE) object
   - list of LSP attributes:
     – BANDWIDTH object
     – one or more METRIC objects
     – LSPA object

   The LSP path name is inserted into the Symbolic Path Name TLV in the LSP object.
4. The PCE-initiated LSP template ID to be used at the PCC, if any, is included in the PATH-PROFILE-ID TLV of the Path Profile object or the Association ID in an ASSOCIATION object of type Policy.

   The profile ID matches the PCE-initiated LSP template ID at the PCC and is not the same as the path profile ID used on the PCE to compute the path of this PCE-initiated LSP.

   Note:
   The range of the LSP template ID is 32-bits, but the range of the Association Group ID is only 16 bits. Therefore, the range of Association Group IDs that can be used to reference a template ID is limited to the bottom 16 bits of the 32-bit template ID range.

5. The path profile ID is used on the PCE to compute the path of this PCE-initiated LSP.

6.2.1.2.1.2 SR OS router procedure on receiving a PCInitiate message

1. If a PCInitiate message includes a name that is a duplicate of an existing LSP on the router, the system generates an error.
2. The router assigns a PLSP-ID and looks up the specified PCE-initiated LSP template ID, if any, or the default PCE-initiated LSP template, to retrieve the local parameters. The router instantiates the SR-TE LSP.

   This lookup uses the PATH-PROFILE-ID TLV if that is included in the PCInitiate message, or the Association ID of the Policy ASSOCIATION object if that is included in the PCInitiate message.

3. The instantiated LSP is added to the TTM and is used by all applications that look up a tunnel in the TTM.
4. The router crafts a PCRpt message with the tunnel ID, LSP ID, and the RRO and passes it along with the PLSP ID set to the assigned value and the delegation bit set in the LSP object to the PCE.

6.2.1.2.1.3 NSP procedure on receiving a PCRpt message for a PCE

1. The NSP confirms the bandwidth reservation and updates its LSP database. At this point, the PCC and PCE are synchronized.
2. The NSP reports the PLSP ID or tunnel ID to the application, for example NSD, or to the operator that uses it in the specific application that originated the request.
3. The PCE can perform updates to the path during the lifetime of the LSP by using the PCUpd message in the same way as with a delegated PCC-initiated LSP.

6.2.1.2.2 LSP deletion using PCEP

The procedures in this section apply to the deletion of a PCE-initiated LSP. See RFC 8281 for additional protocol level details. These procedures are applicable when the user manually deletes the PCE-initiated LSP or the NSP application, or when NSD requests the deletion of the PCE-initiated LSP. See SR OS router procedures for information about procedures that apply when a network event occurs.

The NSP NFM-P crafts a PCInitiate message for the corresponding PLSP ID and sets the R-bit in the SRP object flags to indicate to the PCC that it must delete the LSP. The NSP sends the message to the PCC using PCEP.

6.2.1.2.2.1 SR OS router procedures on receipt of a PCInitiate with the R-bit set

1. The router deletes the state of the LSP.
2. The router crafts a PCRpt message with the R-bit set in the LSP object flags.

6.2.1.2.2.2 NSP procedures upon issuance of pce-init delete command

1. The NSP deletes the LSP from its LSP database.

6.2.1.2.3 Dynamic state handling for PCE-initiated LSPs

The sequences in this section describe dynamic state handling for PCE-Initiated LSPs.

6.2.1.2.3.1 NSP procedures

1. The NRC-P controls the creation and deletion of the PCE-initiated LSP.
2. The NSP performs all LSP creation retries. If the PCC rejects an instantiation, the NSP can issue a new request for instantiation or give up and delete the LSP state locally after a configured maximum number of retries.

3. The NSP can reject an instantiation request if it does not receive a PCRpt from the PCC message within a configured time frame.
4. When the PCEP session comes up and the LSP database synchronization from the PCC to PCE is complete, the NSP reinitiates the PCE-initiated LSPs that are missing from the PCC reports.
5. If a PCEP session goes down, the NSP stops sending new or updated PCE-initiated LSP paths to that PCC; therefore, the LSP database on the NSP and PCC can go out of synchronization during that time.
6. If the PCEP session to a PCC goes down, the NSP marks all PCE-initiated and PCC-initiated LSPs for that PCC as stale but keeps their reservation for an amount of time equal to the state timer value.

   The state timer applies to both PCE-initiated and PCC-initiated LSPs on the PCE and is set to a fixed value of 10 minutes.

   Note: The state timer on the PCE must be considerably larger than the maximum state timer amount all the PCCs (configurable via the **config>router>pcep>pcc>state-timer** command) to give the PCC time to clean up PCE-initiated LSPs and prevent PCInit requests for duplicate LSPs.

   - If the PCEP session is reestablished within that timer value, the NRC-P reinitiates all PCE-initiated LSPs toward the PCC from which a PCRpt remove with the special error code LSP_ERR_SYNC_DELETE was received during the LSP database synchronization with the PCC.
   - If the state timer expires, the NRC-P releases the resources but does not delete the LSPs from the LSP database. If the PCEP session comes up subsequently, the NSP NFM-P recomputes the path of each LSP from which a PCRpt with the remove flag set in the LSP object and with the special error code LSP_ERR_SYNC_DELETE was received during the LSP database synchronization with the PCC and sends the PCC a PCInitiate message for each LSP.
7. If the VSR-NRC informs the NSP NFM-P of a PCRpt with the remove flag set in the LSP object and an SRP object set for each of them, the NSP NFM-P follows the same procedures for these LSPs as when the PCEP session goes down.

6.2.1.2.3.2 SR OS router procedures

The following table lists the impact of various PCC operational events on the status of PCE-initiated LSPs.

Table 39: Impact of PCC operational events

Event	Impact on PCE-initiated LSPs	
	Oper-down	**Deleted**
MPLS shutdown	X[22]	—
no mpls	—	X[23]
no pce-initiated-lsp	—	X (all)
no sr-te	—	X (sr-te)[23]

22 Also results in a PCRpt to the PCE with LSP error admin down.

23 Also results in a PCRpt to the PCE with LSP deleted.

Event	Impact on PCE-initiated LSPs	
	Oper-down	**Deleted**
sr-te shutdown	X (sr-te)[22]	—
pcc shutdown	—	X (all)[24]
pcc peer shutdown	—	X[24]
Delete LSP template ID	—	X (LSPs using template)[23]
Delete default LSP template	—	X (all)[23]

The following list is a detailed description of the PCC actions on PCE-initiated LSPs as a result of PCC operational events:

1. If an event causes the PCE-initiated LSPs to be deleted by the PCC, the PCC sends a PCRpt with the remove flag in both the SRP object and the LSP object set for each impacted LSP. If the event is a failure of the PCEP session to the PCE, or a shutdown of the PCC or PCC peer, the PCRpt is sent, with the special error code LSP_ERR_SYNC_DELETE, only after the PCEP session comes back up during the PCC resynchronization with the PCE.
2. If an event causes PCE-initiated LSPs to go operationally down, the PCC router sends a PCRpt with the operational bits in the LSP object set to DOWN for each impacted LSP.
3. If the user shuts down the PCC process on the router, all PCE-initiated LSPs are deleted. When the user performs a **no shutdown** of the PCC process, the PCC reports to the PCE so that the NSP is aware.
4. If a PCEP peer is shut down, the PCEP session goes down but the PCC keeps the state of all PCE-initiated LSPs, subject to the rules about redelegation and the cleanup of state, as described in section 5.7.5 of RFC 8231 and section 6 of RFC 8281. The following rules apply to all LSPs delegated to the PCE.

   Redelegation timers and state timers are started when the PCEP session goes down or the PCE signals overload. See Configuring PCE-initiated LSPs for information about configuring these timers. The system enforces that the state timer value is greater than the **redelegation-timer** value, as specified in RFC 8231.

   The objectives of redelegation are described in section 5.7.5 of RFC 8231. The existing LSP delegation state is maintained while the LSP redelegation timer is running, which gives the PCE time to recover. At the expiry of the redelegation timer, the PCC attempts to redelegate the LSPs to the PCE. The PCC performs the following redelegation process for both PCE-initiated and PCC-initiated LSPs:

   - If the PCEP session to the existing PCE is still down or the PCE is still in overload, return the delegation state to the PCC for all the delegated LSPs.
   - Wait until the PCEP session comes up, then attempt to redelegate the remaining LSPs back to the PCE. For each LSP, set a redelegation attempted flag when redelegation is attempted. If redelegation is accepted for all PCE-initiated LSPs delegated to the PCC before the state timer expires, the system is behaving as expected.

[24] A PCRpt with delete and a special error code, for example, LSP_ERR_SYNC_DELETE, is sent during the PCC rejoin synchronization that occurs when the PCC or PCC peer comes back up.

- If the state timer expires, wait until all LSPs have been processed. The PCC performs the configured action for LSPs that are not redelegated, but have the redelegation attempted flag set. If the action is **delete**, LSPs are deleted; otherwise, wait until the PCEP session comes up and then attempt to redelegate the remaining LSPs back to the PCE.

6.2.1.3 PCEP support for RSVP-TE LSPs

This section describes PCEP support of PCC-initiated RSVP-TE LSP. The PCEP support of an RSVP-TE LSP provides the following modes of operation:

- PCC-initiated and PCC-controlled
- PCC-initiated and PCE-computed
- PCC-initiated and PCE-controlled

Each primary and secondary path is assigned its own unique path LSP ID (PLSP ID). The PCC indicates to the PCE the state of each path (both up and down) and which path is currently active and carrying traffic (active state).

The PCEP support of an RSVP-TE LSP differs from that of an SR-TE LSP in that PCE-initiated RSVP-TE LSPs are not supported.

6.2.1.3.1 PCEP support for RSVP-TE LSP configuration

Use the following MPLS-level and LSP-level CLI commands to configure RSVP-TE LSPs on a router acting as a PCC.

- **config>router>mpls>pce-report rsvp-te** {**enable** | **disable**}
- **config>router>mpls>lsp>path-profile** *profile-id range* [**path-group** *group-id range*]
- **config>router>mpls>lsp>pce-report** {**enable** | **disable** | **inherit**}
- **config>router>mpls>lsp>path-computation-method pce**
- **config>router>mpls>lsp>pce-control**

Note: The PCE function implemented in the NSP and referred to as the Network Resource Controller for Packet (NRC-P), supports only Shared Explicit (SE) style bandwidth management for TE LSPs. The PCEP protocol does not have means for the PCC to convey this value to the PCE, so, regardless of whether the LSP configuration option **rsvp-resv-style** is set to **se** or **ff**, the PCE always uses the SE style in the CSPF computation of the path for a PCE-computed or PCE-controlled RSVP-TE LSP.

A **one-hop-p2p** or a **mesh-p2p** RSVP-TE **auto-lsp** only supports the **pce-report** command in the LSP template:

config>router>mpls>lsp-template>pce-report {**enable** | **disable** | **inherit**}

The user must shut down the LSP template before changing the value of the **pce-report** option.

A manual bypass LSP does not support PCE-related commands. Reporting a bypass LSP to PCE is not required because it does not book bandwidth.

All other MPLS, LSP, and path-level commands are supported, with the exception of **backup-class-type**, **class-type**, **least-fill**, **main-ct-retry-limit**, **mbb-prefer-current-hops**, and **srlg** (on secondary standby paths), which, if enabled, result in a no operation.

RSVP-TE PCC-initiated LSPs supports the same instantiation modes as SR-TE PCC-initiated LSPs. See LSP initiation for more information.

6.2.1.3.2 Behavior of the LSP path update

When the **pce-control** command is enabled, the PCC delegates the control of the RSVP-TE LSP to the PCE.

The NRC-P sends a path update using the PCUpd message in the following cases:

- a failure event that impacts a link or a node in the path of a PCE-controlled LSP

  The operation is performed by the PCC as an MBB, if the LSP remained in the UP state because of protection provided by FRR or a secondary path. If the LSP went down, the update brings it into the UP state. A PCRpt message is sent by the PCC for each change to the state of the LSP during this process.

- a topology change that impacts a link in the path of a PCE-controlled LSP

  This topology change can be a change to the IGP metric, the TE metric, the admin group, or the SRLG membership of an interface. This update is performed as an MBB by the PCC.

- the user performed a manual resignal of the PCE-controlled RSVP-TE LSP path from the NRC-P

  This update is performed as an MBB by the PCC.

- the user performed a Global Concurrent Optimization (GCO) on a set of PCE-controlled RSVP-TE LSPs from the NRC-P

  This update is performed as an MBB by the PCC.

The procedures for the path update are the same as those for an SR-TE LSP. See LSP initiation for more information. However, the PCUpd message from the PCE does not contain the label for each hop in the computed ERO. PCC then signals the path using the ERO returned by the PCE and, if successful, programs the datapath and then sends the PCRpt message with the resulting RRO and hop labels provided by RSVP-TE signaling.

If the signaling of the ERO fails, the ingress LER returns a PCErr message to PCE with the LSP Error Code field of the LSP-ERROR-CODE TLV set to a value of 8 (RSVP signaling error).

If the **no adaptive** option is set for an RSVP-TE LSP, the ingress LER cannot perform an MBB for the LSP. A PCUpd message received from the PCE is then failed by the ingress LER, which returns a PCErr message to the PCE with the LSP Error code field of the LSP-ERROR-CODE TLV set to a value of 8 (RSVP signaling error).

When the NRC-P reoptimizes the path of a PCE-controlled RSVP-TE LSP, it is possible that a path that satisfies the constraints of the LSP no longer exists. In this case, the NRC-P sends a PCUpd message with an empty ERO, which forces the PCC to bring down the path of the RSVP-TE LSP.

The NRC-P sends a PCUpd message with an empty ERO if the following cases are true:

- The requested bandwidth is the same as current bandwidth, which avoids bringing down the path on a resignal during an MBB transition.
- Local protection is not currently in use, which avoids bringing down a path that activated an FRR backup path. The LSP can remain on the FRR backup path until the NRC-P finds a new primary path.
- The links of the current path are all operationally up, which allows NRC-P to make sure that the RSVP control plane can report the path down when a link is down and not prematurely bring the path down with an empty ERO.

6.2.1.3.3 Behavior of LSP MBB

In addition to the Make-Before-Break (MBB) support when the PCC receives a path update (see Behavior of the LSP path update for more information), an RSVP-TE LSP supports the MBB procedure for any parameter configuration change, including the PCEP-related commands when they result in a change to the path of the LSP.

If the user adds or modifies the **path-profile** command for an RSVP-TE LSP, a Config Change MBB is only performed if the **path-computation-method pce**, **pce-report**, or **pce-control** options are enabled on the LSP. Otherwise, no action occurs. When the **path-computation-method pce**, **pce-report**, or **pce-control** options are enabled on the LSP, the Path Update MBB (**tools perform router mpls update-path**) fails, resulting in a no operation.

MBB is also supported for the Manual Resignal and Auto-Bandwidth MBB types.

When the LSP goes into a MBB state at the ingress LER, the behavior depends on the LSP operating mode.

6.2.1.3.3.1 PCE-controlled LSPs

The LSP MBB procedures for a PCE-controlled LSP (**pce-control** command enabled) are as follows.

Steps 1 through 5 of the following procedure apply to the Config Change, Manual Resignal, and Auto-Bandwidth MBB types. The Delayed Retry MBB type used with the SRLG on that secondary standby LSP feature is not supported with a PCE-controlled LSP. See Behavior of secondary LSP paths for information about the SRLG on secondary standby LSP feature.

1. The PCC temporarily removes the delegation by sending a PCRpt message for the corresponding PLSP-ID with the delegation D-bit clear.
2. For an LSP with the **path-computation-method pce** option disabled, MPLS submits a path request to the local CSPF including the updated path constraints.
3. For an LSP with the **path-computation-method pce** option enabled, PCC issues a PCReq for the same PLSP-ID and includes the updated constraints in the METRIC, LSPA, or BANDWIDTH objects. The bandwidth object contains the current operational bandwidth of the LSP in the case of the auto-bandwidth MBB.
   - If the PCE successfully finds a path, it replies with a PCRep message with the ERO.
   - If the PCE does not find a path, it replies with a PCRep message containing the No-Path object.
4. If the local CSPF or the PCE return a path, the PCC performs the following actions.
   - The PCC signals the LSP with the RSVP control plane and moves traffic to the new MBB path. The PCC then sends a PCRpt message with the delegation D-bit set to return delegation and containing the RRO and LSP object, with the LSP identifiers TLV containing the LSP-ID of the new MBB path. The message includes the METRIC, LSPA, and BANDWIDTH objects where the P-flag is clear, which indicates the operational values of these parameters. Unless the user disabled the **report-path-constraints** option under the **pcc** context, the PCC also includes a second set of METRIC, LSPA, or BANDWIDTH objects with the P-flag set to convey the constraints of the path to the PCE.
   - The PCC sends a PathTear message to delete the state of the older path in the network. The PCC then sends a PCRpt message to the PCE with the older path PLSP-ID and the remove R-bit set to also have PCE remove the state of the LSP from its database.

5. If the local CSPF or the PCE returns no path, or the RSVP-TE signaling of the returned path fails, the router makes no further requests. That is, there is no retry for the MBB.
   - The PCC sends a PCErr message to PCE with the LSP Error code field of the LSP-ERROR-CODE TLV set to a value of 8 (RSVP signaling error) if the MBB failed because of an RSVP-TE signaling error.
   - The PCC sends a PCRpt message with the delegation D-bit set to return delegation and containing the RRO and LSP objects with the LSP identifiers TLV containing the LSP-ID of the currently active path. The message includes the METRIC, LSPA, and BANDWIDTH objects with the P-flag set to clear to indicate the operational values of these parameters. Unless the user has disabled the **report-path-constraints** option under the **pcc** context, the PCC also includes a second set of METRIC, LSPA, and BANDWIDTH objects with the P-flag set to convey the constraints of the path to the PCE.
6. The ingress LER takes no action in the case of a network-event-triggered MBB, such as FRR Global Revertive, TE Graceful Shutdown, or Soft Pre-Emption.
   - The ingress PE retains the information as required and sets the state of the MBB to one of the FRR Global Revertive, TE Graceful Shutdown, or Soft Pre-emption MBB values but does not perform the MBB action.
   - The NRC-P computes a new path in the case of a Global Revertive MBB because of a failure event. This computation uses the PCUpd message to update the path using the MBB procedure (see Behavior of the LSP path update for information about the procedure). The activation of a bypass LSP by a PLR in the network causes the PCC to issue an updated PCRpt message with the new RRO reflecting the PLR and RRO hops. The PCE releases the bandwidth on the links that are no longer used by the LSP path.
   - The NRC-P computes a new path in the case of the TE graceful MBB if the RSVP-TE is using the TE metric, because the TE metric of the link in TE graceful shutdown is set to infinity. This computation uses the PCUpd message to update the path using the MBB procedure. See Behavior of the LSP path update for a description of the MBB procedure.
   - The NRC-P does not act on the TE graceful MBB if the RSVP-TE is using the IGP metric or is on the soft pre-emption MBB; however, the user can perform a manual resignal of the LSP path from the NRC-P to force a new path computation, which accounts for the newly available bandwidth on the link that caused the MBB event. This computation uses the PCUpd message to update the path using the MBB procedure. See Behavior of the LSP path update for information about the MBB procedure.
   - The user can perform a manual resignal of the LSP path from the ingress LER, which forces an MBB for the path as per the remove-delegation, MBB, and return-delegation procedures described in this section.
   - If the user performs **no pce-control** while the LSP still has the state for any of the network-event-triggered MBBs, the MBB is performed immediately by the PCC. See PCE-computed LSPs for a description of the procedures for a PCE-computed LSP. See PCC-controlled LSPs for a description of the procedure for a PCC-controlled LSP.
7. The timer-based resignal MBB behaves like the TE graceful or soft pre-emption MBB. The user can perform a manual resignal of the LSP path from the ingress LER or from PCE.
8. The Path Update MBB (**tools perform router mpls update-path**) is failed and results in a no operation. This is true in all cases when the RSVP-TE LSP enables the **pce-report** option.

6.2.1.3.3.2 PCE-computed LSPs

All MBB types are supported for PCE-computed LSPs. The LSP MBB procedure for a PCE-computed LSP (**path-computation-method pce** enabled and **pce-control** disabled) is as follows.

1. The PCC issues a PCReq for the same PLSP-ID and includes the updated constraints in the METRIC, LSPA, and BANDWIDTH objects.
   - If the PCE successfully finds a path, it replies with a PCRep message with the ERO.
   - If the PCE does not find a path, it replies with a PCRep message containing the No-Path object.
2. If the PCE returns a path, the PCC signals the LSP with the RSVP control plane and moves traffic to the new MBB path. If **pce-report** is enabled for this LSP, the PCC sends a PCRpt message with the delegation D-bit clear to retain control and containing the RRO and LSP object with the LSP identifiers TLVs containing the LSP-ID of the new MBB path. The message includes the METRIC, LSPA, and BANDWIDTH objects where the P-flag is clear, which indicates the operational values of these parameters. Unless the user disables the **report-path-constraints** option in the **pcc** context, the PCC also includes a second set of METRIC, LSPA, and BANDWIDTH objects with the P-flag set to convey the path constraints to the PCE.
3. If the PCE returns no path or the RSVP-TE signaling of the returned path failed, MPLS puts the LSP into retry mode and sends a request to the PCE at the frequency of the *retry-timer* value (in seconds) and up to the *retry-count* value.
4. When the **pce-report** command is enabled for the LSP and the FRR Global Revertive MBB is triggered following a bypass LSP activation by a PLR in the network, the PCC issues an updated PCRpt message with the new RRO reflecting the PLR and RRO hops. The PCE releases the bandwidth on the links that are no longer used by the LSP path.
5. If the user changes the RSVP-TE LSP configuration from **path-computation-method pce** to **no path-computation-method**, MBB procedures are not supported. In this case, the LSP path is torn down and put into retry mode to compute a new path from the local CSPF on the router to signal it.

6.2.1.3.3.3 PCC-controlled LSPs

All MBB types are supported for PCC-controlled LSPs. The LSP MBB procedure for a PCC-controlled LSP (**path-computation-method pce** and **pce-control** disabled) is as follows.

1. MPLS submits a path request, including the updated path constraints, to the local CSPF.
2. If the local CSPF returns a path, the PCC signals the LSP with RSVP control plane and moves traffic to the new MBB path. If the **pce-report** command is enabled for this LSP, the PCC sends a PCRpt message with the delegation bit clear to retain control. The PCRpt message also contains the RRO and LSP object with the LSP identifiers TLV containing the LSP-ID of the new MBB path. It includes the METRIC, LSPA, and BANDWIDTH objects where the P-flag is clear, which indicates the operational values of these parameters. Unless the user disables the **report-path-constraints** option in the **pcc** context, the PCC also includes a second set of METRIC, LSPA, and BANDWIDTH objects with the P-flag set to convey the path constraints to the PCE.
3. If the CSPF returns no path, or the RSVP-TE signaling of the returned path fails, MPLS puts the LSP into retry mode and sends a request to the local CSPF at the frequency of the configured **retry-timer** value (in seconds) and up to the **retry-count** value.
4. When **pce-report** is enabled for the LSP and the FRR Global Revertive MBB is triggered following a bypass LSP activation by a PLR in the network, PCC issues an updated PCRpt message with the new

RRO reflecting the PLR and RRO hops. PCE releases the bandwidth on the links that are no longer used by the LSP path.

6.2.1.3.4 Behavior of secondary LSP paths

Each of the primary, secondary standby, and secondary non-standby paths of the same LSP must use a separate PLSP-ID. In the PCE function of the NSP NFM-P, the NRC-P checks the LSP-IDENTIFIERS TLVs in the LSP object and can identify which PLSP-IDs are associated with the same LSP or the same RSVP session. The following parameters are checked:

- IPv4 Tunnel Sender Address
- Tunnel ID
- Extended Tunnel ID
- IPv4 Tunnel Endpoint Address

This approach allows the use of all the PCEP procedures for all types of LSP paths.

The PCC indicates the following states for the path in the LSP object to the PCE: down, up (signaled but is not carrying traffic), or active (signaled and carrying traffic).

The PCE tracks active paths and displays them in the NSP NFM-P GUI. It also provides only the tunnel ID of an active PLSP-ID to a specific destination prefix when a request is made by a service or a steering application.

The PCE recomputes the paths of all PLSP-IDs that are affected by a network event. The user can select each path separately on the NSP NFM-P GUI and trigger a manual resignal of one or more paths of the LSP.

Note:
Enabling the **srlg** option on a secondary standby path results in a **no** operation. The NRC-P supports link and SRLG disjointness using the PCE path profile, and the user can apply to the primary and secondary paths of the same LSP. See PCE path profile support for more information.

6.2.1.3.5 PCE path profile support

The PCE path profile ID and path group ID are configured at the LSP level.

The NRC-P can enforce path disjointness and bidirectionality among a pair of forward and a pair of reverse LSP paths. Both pairs of LSP paths must use a unique path group ID along with the same Path Profile ID, which is configured on the NRC-P to enforce path disjointness or path bidirectionality constraints.

When the user wants to apply path disjointness and path bidirectionality constraints to LSP paths, it is important to use the following guidelines. The user can configure the following sets of LSP paths.

- Configure a set of LSPs, consisting of a pair of forward LSPs and a pair of reverse LSPs, each with a single path, primary or secondary

  The pair of forward LSPs can originate and terminate on different routers. The pair of reverse LSPs must mirror the forward pair. In this case, the path profile ID and the path group ID configured for each LSP must match. Because each LSP has a single path, the bidirectionality constraint applies automatically to the forward and reverse LSPs, which share the same originating node and the same terminating routers.

- Configure a pair of LSPs consisting of a forward LSP and a reverse LSP, each with a primary path and a single secondary path, or each with a couple of secondary paths

  Because the two paths of each LSP inherit the same LSP level path profile ID and path group ID configuration, the NRC-P path computation algorithm cannot guarantee that the primary paths in both directions meet the bidirectionality constraint. That is, it is possible that the primary path for the forward LSP shares the same links as the secondary path of the reverse LSP, or for the opposite situation to be true.

6.2.2 LSP path diversity and bidirectionality constraints using path profiles

The PCE path profile defined in *draft-alvarez-pce-path-profiles* is used to request path diversity or a disjoint for two or more LSPs originating on the same or different PE routers. The PCE path profile is also used to request that paths of two unidirectional LSPs between the same two routers use the same TE links. This is referred to as the bidirectionality constraint. As an alternative, SR OS supports the use of association groups to signal path diversity constraints. See Policy Association Group for more information about association group support.

Path profile and association group are not interoperable. That is, LSPs with path profile and those with association group cannot be considered in the same group and cannot be compared against each other.

The user defines path profiles directly on the NRC-P Policy Manager using LSP path constraints, which are metrics with upper bounds specified, and with an objective, which are metrics optimized with no bound specified. The NRC-P Policy Manager allows the user to configure the following PCE constraints within each PCE Path Profile:

- path diversity, node-disjoint, link-disjoint
- path bidirectionality, symmetric reverse route preferred, symmetric reverse route required
- maximum path IGP metric (cost)
- maximum path TE metric
- maximum hop count

The user can also specify which PCE objective to use to optimize the path of the LSP in the PCE path profile:

- IGP metric (cost)
- TE metric
- hops (span)

The CSPF algorithm optimizes this objective. If a constraint is provided for the same metric, the CSPF algorithm ensures the selected path achieves a lower or equal value to the bound specified in the constraint.

For hop-count metrics, if a constraint is sent in a METRIC object, and is also specified in a PCE profile referenced by the LSP, the constraint in the METRIC object is used.

For IGP and TE metrics, if an objective is sent in a METRIC object, and is also specified in a PCE profile referenced by the LSP, the objective in the path profile is used.

The constraints in the BANDWIDTH object and the LSPA object that include or exclude admin-group constraints and setup and hold priorities are not supported in the PCE profile.

To indicate the path diversity and bidirectionality constraints to the PCE, the user must configure the profile ID and path group ID of the PCE path that the LSP belongs to. The CLI for this is described in the

Configuring and operating SR-TE section. The path group ID does not need to be defined in the PCE as part of the path profile configuration; it implicitly identifies the set of paths that must have the path diversity constraint applied.

The user can only associate a single path group ID with a specific PCE path profile ID for an LSP; however, the same path group ID can be associated with multiple PCE profile IDs for the same LSP.

The PCC infers the path profiles using the path ID in the path request. When the PE router acting as a PCC wants to request path diversity from a set of other LSPs belonging to a path group ID value, it adds a new path profile object into the PCReq message. The object contains the path profile ID and the path group ID as an extended ID field. That is, the diversity metric is carried in an opaque way from PCC to PCE.

The bidirectionality constraint operates the same way as the diversity constraint. The user can configure a PCE profile with both the path diversity and bidirectionality constraints. The PCE checks if there is an LSP in the reverse direction that belongs to the same path group ID as an originating LSP it is computing the path for, and enforces the constraint.

For the PCE to be aware of the path diversity and bidirectionality constraints for an LSP that is delegated but for which there is no prior state in the NRC-P LSP database, the PATH-PROFILE object is included in the PCRpt message with the P-flag set in the common header to indicate that the object must be processed.

The following table lists the new objects introduced in the PCE path profile.

Table 40: PCEP path profile extension objects and TLVs

TLV, object, or message	Contained in object	Contained in message
PATH-PROFILE-CAPABILITY TLV	OPEN	OPEN
PATH-PROFILE object	—	PCReq, PCRpt, PCInitiate

A PATH-PROFILE object can contain multiple TLVs containing each profile ID and extend ID, and should be processed properly. If multiple path profile objects are received, the first object is interpreted and the others are ignored. The PCC and the PCE support all PCEP capability TLVs defined in this chapter and always advertise them. If the OPEN object received from a PCEP speaker does not contain one or more of the capabilities, the PCE or PCC does not use them during that PCEP session.

6.2.3 PCEP Associations

SR OS supports the use of PCEP Association Groups to reference SR-TE and RSVP-TE LSP path constraints. PCEP Association Groups allow LSPs to share common information, such as common policies or common configuration parameters. Users can use association groups for PCC-initiated SR-TE and RSVP-TE LSPs, PCE-initiated SR-TE LSPs, and on-demand SR-TE auto-LSPs. Association groups are supported on both the PCC and NSP PCE.

The PCEP ASSOCIATION object defines an Association ID and an Association Type to signal any type of association between LSPs (as defined in RFC 8697). Association groups are identified by a tuple consisting of an Association ID, Association Type, and Association Source. Association groups offer a disaggregated approach to specifying association, where the Association ID is equivalent to the path group ID, and the tuple (Association ID, Association Type, Association Source) is equivalent to the path profile ID.

The tuple is the key to the association group. Both IPv4 and IPv6 PCEP control planes support signaling of the Association IDs. The NSP uses the Association ID to reference a path profile.

Path diversity constraints are signaled using the Disjoint Association Type defined in RFC 8800, while policy constraints are signaled using the Policy Association Type defined in *draft-ietf-pce-association-policy-16*. The Association ID for a policy association is also used by the PCE to signal which LSP template to bind to a PCE-initiated SR-TE LSP.

Use the commands in the following contexts to configure PCE associations on the PCC.

```
configure router pcep pcc pce-associations diversity
configure router pcep pcc pce-associations policy
```

You must configure the Association ID and Association Source address in these contexts. In most cases, the Association Source address (which may be IPv4 or IPv6) is the PCC source address. Configuring the Association Source address allows interoperability with third-party controllers that assume a specific allocation scheme, which supports cases such as the diversity between LSPs that originate on different PCC nodes but are configured with the same Association Source address for a specific Association ID and Association Type.

See Diversity Association Group and Policy Association Group for more details about the configuration of these association types.

Association Group support for PCC-initiated SR-TE and RSVP-TE LSPs

A PCC-initiated SR-TE or RSVP-TE LSP can have up to five associations. These associations must be defined under the PCC context.

Use the following commands to configure the type and the association for a PCC-initiated P2P SR-TE LSP:

- **MD-CLI**

```
configure router mpls lsp type p2p-sr-te
configure router mpls lsp pce-associations diversity
configure router mpls lsp pce-associations policy
```

- **classic CLI**

```
configure router mpls lsp sr-te
configure router mpls lsp pce-associations diversity
configure router mpls lsp pce-associations policy
```

The **configure router mpls lsp pce-associations** context is also supported for PCC-initiated RSVP-TE LSPs.

On-demand SR-TE auto-LSPs are bound to association groups using the commands in the **pce-associations** context in the LSP template, with up to five associations per template.

Use the following commands to configure on-demand SR-TE auto-LSPs and bind them to association groups:

- **MD-CLI**

```
configure router mpls lsp-template type p2p-sr-te-on-demand
configure router mpls lsp-template pce-associations diversity
configure router mpls lsp-template pce-associations policy
```

- **classic CLI**

```
configure router mpls lsp-template on-demand-p2p-srte
configure router mpls lsp-template pce-associations diversity
configure router mpls lsp-template pce-associations policy
```

The configuration of a PCE association for an LSP or LSP template and the configuration of the path profile ID and path group ID are mutually exclusive. Path profile and association group are also not interoperable. That is, LSPs with path profile and those with association group cannot be considered in the same group and cannot be compared against each other.

The PCC can signal one or more ASSOCIATION objects for each Association Type based on the configuration of the PCC-initiated LSP. This information is passed by MPLS to the PCC module. The PCC can include the ASSOCIATION object and its TLVs in any of the following PCEP messages: PCReq, PCRpt. The PCE may reflect the same ASSOCIATION objects and TLVs in any of the following PCEP messages: PCRep, PCUpd. In the specific case of the Diversity ASSOCIATION object, the PCE includes the status of the diversity as computed by PCE in the DISJOINTNESS-STATUS-TLV.

Association group support for PCE-initiated LSPs

The only ASSOCIATION object applicable to a PCE-initiated LSP is the policy type. The PCE uses the Association ID for the policy type to signal a reference to the required PCE-initiated LSP template to use on the router for the associated PCE-initiated SR-TE LSP. The ASSOCIATION object is included in the following PCEP messages: PCInitiate, PCUpd. This information is also included in subsequent PCRpt messages from the PCC to the PCE. The NSP uses this information to reference a policy association for the LSP.

Note: The range of the LSP template ID is 32-bits, but the range of the Association Group ID is only 16 bits. Therefore, the system limits the range of Association Group IDs that can be used to reference a template ID to the bottom 16 bits of the 32-bit template ID range.

ASSOCIATION object error handling

If the ASSOCIATION objects in the PCE do not match the ones sent by PCC in the PCReq or PCRpt messages, the PCC returns an error.

If the PCC rejects a request for a particular association from MPLS with a "path not found" message, the PCC updates the reason code and tears down the path if it is up, and retries until the retry limit is exceeded.

The router handles consistency between the association group configuration for a set of LSPs on a specific PCC. That is, an association group can only have one set of parameters within an association (for example, Diversity Type and Disjointness Type), which ensures that LSPs added to the same association group do not have inconsistent parameters. However, LSPs that originate on different PCCs can be added to the same association group, but those association groups have different parameters configured on the different PCCs. In that case, only the NSP can detect parameter inconsistencies.

For LSPs that are delegated and have inconsistent association parameters, the NSP sends a PCUpd down message followed by a PCErr message with the appropriate error message. This causes the affected LSPs to go operationally down. For non-delegated LSPs, the NSP sends a PCErr message.

If an LSP is added to an association group that has inconsistent parameters when compared with the same association group to which operationally up LSPs are already assigned, the NSP only registers an error on the new LSP and leaves the existing LSPs undisturbed.

6.2.3.1 Diversity Association Group

Use the following CLI syntax to configure the Diversity Association Group parameters under the PCC:

- **MD-CLI**

```
configure router pcep
     pcc
       pce-associations
          diversity [assoc-name] string
            association-id number
            association-source ipv4-address
            disjointness-reference {true|false}
            disjointness-type {strict | loose}
            diversity-type {none | link | node | srlg-link | srlg-node}
```

- **classic CLI**

```
configure router pcep
   pcc
      pce-associations
         [no] diversity association-name
            association-id [1..65535]
            no association-id
            association-source ip-address
            no association-source
            disjointness-reference
            no disjointness-reference
            disjointness-type {loose | strict}
            no disjointness-type
            diverity-type {link | node | srlg-link | srlg-node}
            no diversity-type
```

The Association ID for the diversity type supports the user-configured mode of operation [RFC 8800]. The ID value is interpreted as a global value independent of the IP address of the PCC or PCE node that uses it to associate LSPs.

The user can configure the following parameters for the path Diversity Association:

- **diversity-type** – **link**, **node**, **srlg-link**, or **srlg-node**. Configuration of this parameter is mandatory. If this parameter is not configured, the system does not validate the association configuration.
- **disjointness-type** – **strict** or **loose**. The default is **loose**.
- **disjointness-reference** – used to indicate if this LSP path is the reference path for the disjoint set of paths. When set, the PCE must first compute the path of this LSP, and then apply the requested disjointness type to compute the path of all other paths in the same diversity association ID. The default is **no disjointness reference**.

The DISJOINTNESS-CONFIGURATION TLV is used to convey the diversity parameters requested for the set of SR-TE or RSVP-TE LSPs. The PCE uses the DISJOINTNESS-STATUS TLV to convey the status of the diversity parameters after the path computation. The same flags are set to indicate whether any of the parameters were met by the returned path.

6.2.3.2 Policy Association Group

Use the following CLI syntax to configure the policy association under the PCC:

- **MD-CLI**

```
configure router pcep
     pcc
       pce-associations
         policy [assoc-name] string
            association-id number
            association-source ip-address
```

- **classic CLI**

```
configure router pcep
   pcc
      pce-associations
         [no] policy association-name
            association-id association-id
            no association-id
            association-source ip-address
            no association-source
```

This association is always operator-configured [*draft-ietf-pce-association-policy*], that is, the value of the Association ID has a global meaning for this Association Type within a specific domain, independent of the source address in the ASSOCIATION object. For example, a PCC node can associate a policy Association ID with an SR-TE or RSVP-TE LSP included in the PCRpt message by including the PCC source address in the ASSOCIATION object, but the PCE looks up a path profile ID solely based on the Association ID value for PCC nodes in that domain. If another PCC node in the same domain signals another LSP with the same policy Association ID, the PCE also looks up the same path profile.

6.2.4 Path computation fallback for PCC-initiated LSPs

For PCC-initiated RSVP-TE and SR-TE LSPs, the router supports fallback to a local path computation method in the case where the configured PCEP sessions are down or the PCE is unreachable, or when all configured PCEs are signaling overload and the redelegation timer expires while all configured LSPs are signaling overload so that the LSP cannot be redelegated. The fallback method can be configured to be the **local-cspf** or **none**. In the latter case, MPLS uses the hop-to-label translation (SR-TE LSPs) or the explicit IGP path (RSVP-TE LSPs).

This capability is supported by both active and passive stateful LSPs. Active stateful LSPs are fully delegated to the PCE by being both PCE-computed (**path-computation-method pce**) and PCE-controlled. Passive stateful LSPs are PCE-computed.

Note:
For the passive stateful case, it is important that the **retry-timer** and **retry-limit** values exceed the **redelegation-timer** value, otherwise, the LSP may go operationally down before the fallback path computation has occurred.

A fallback path computation method is configured as follows:

```
configure>router>
   mpls
      lsp lsp-name
         pce-control
         path-computation-method {pce | local-cspf}
         fallback-path-computation-method {none | local-cspf}
```

If **none** is configured, MPLS uses the default method based on the configured path, which is hop-to-labeled path computation for SR-TE LSPs and IGP-based path computation for RSVP-TE LSPs.

The **fallback-path-computation-method** command is only valid for **path-computation-method pce**, irrespective of whether **pce-control** is configured. The **fallback-path-computation-method** command cannot be configured if the **path-computation-method local-cspf** or **no path-computation-method** commands are configured.

The fallback mechanism is only triggered if PCC informs MPLS that the PCEP is down. It is not triggered while the PCC is administratively down or is not yet configured.

Note:
On the first local path computation following a fallback, MPLS is not aware of the list of SRLGs or administrative groups that are used by the original path computed by PCE. As a result, MPLS can only provide a list of hops or links to avoid on the first computation.

PCE reports are sent, where applicable, with the delegation bit cleared.

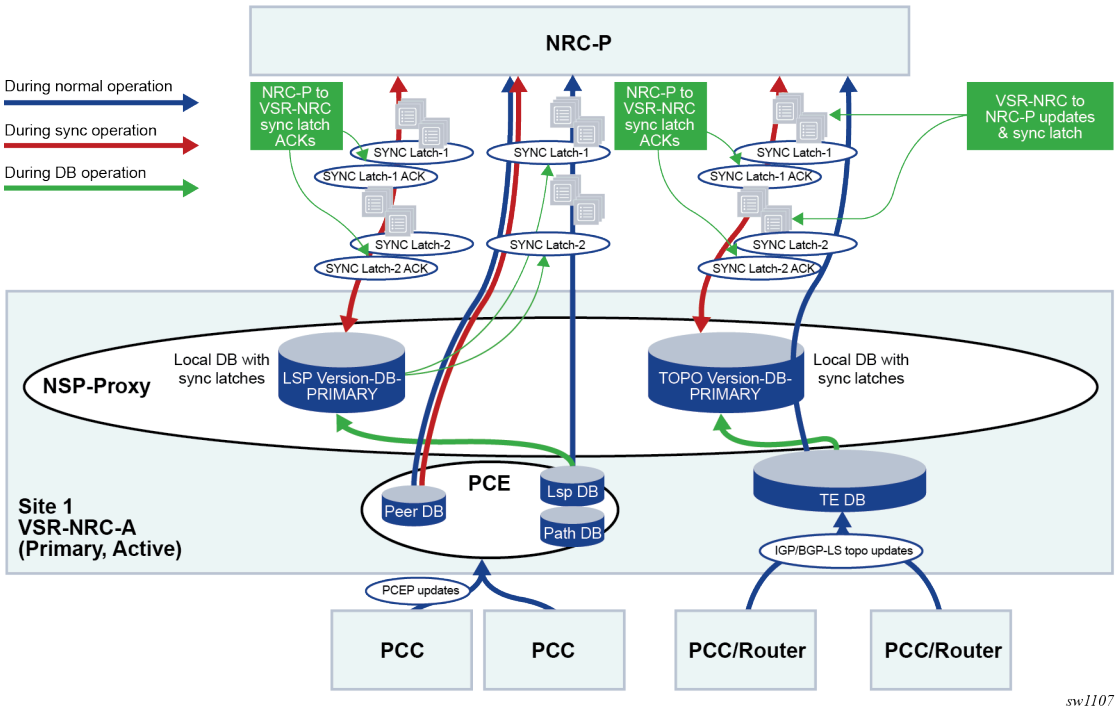
6.3 TE-DB and LSP-DB partial synchronization

VSR-NRC to NSP partial synchronization of TE database (TE-DB) and LSP database (LSP-DB) allows the VSR-NRC to send incremental TE-DB and LSP-DB records to the NRC-P when the cproto session flaps. Without this feature, a full synchronization of all records is performed each time the cproto session flaps which increases the convergence time of the NRC-P PCE.

With partial synchronization, VSR-NRC keeps track of the last record acknowledged by the NRC-P before the cproto session went down. When the cproto session is reestablished, VSR-NRC sends only the records received from the network after the last acknowledged record.

The following figure shows the behavior of the partial synchronization of the TE-DB and LSP-DB.

Figure 62: TE and LSP database partial synchronization



DB refers to the local TE-DB or LSP-DB maintained by the NSP-PROXY on the VSR-NRC.

The phrase Version-DB refers to the new copy of the same DB, TE-DB or LSP-DB, augmented with synchronization latches, and is used during the synchronization process to play back the records and latches to the NRC-P.

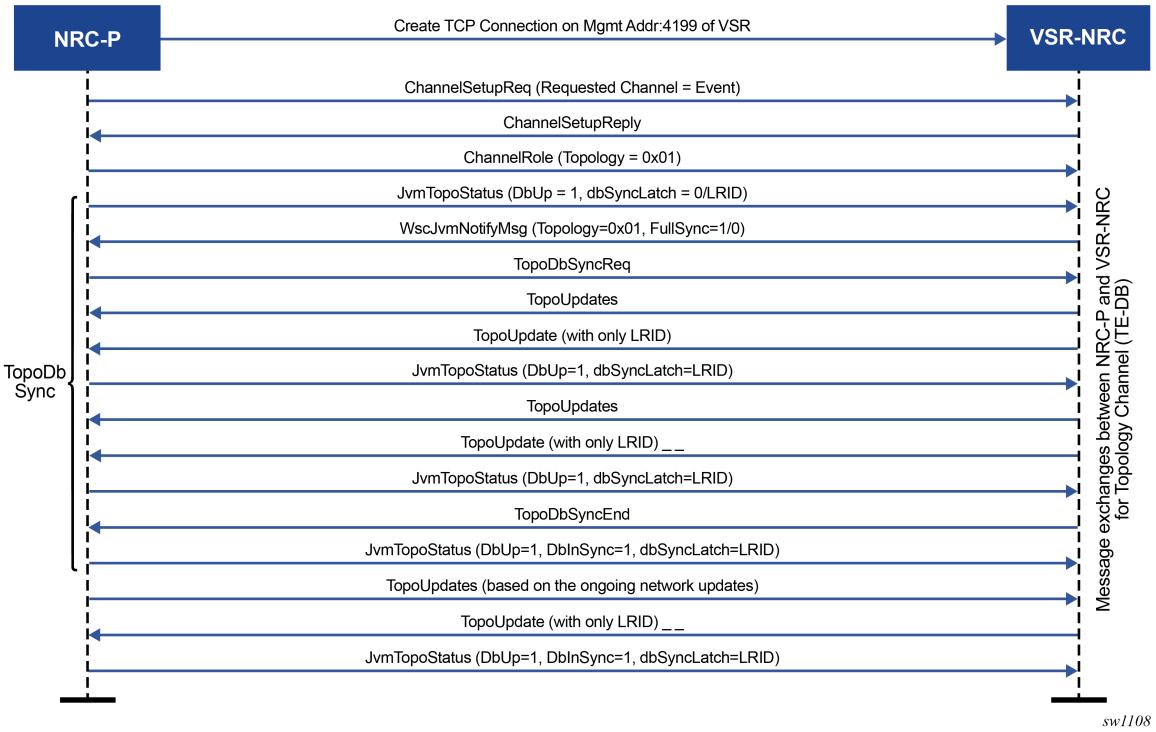
The main processes of the partial synchronization feature are as follows.

- Enhancements to DB maintenance in active VSR-NRC are follows:
  - The active VSR-NRC, VSR-NRC-A or VAR-NRC-B (see VSR-NRC 1+1 redundancy), maintains a local copy of the database, referred to as Version-DB, for each record NSP-PROXY sends to NRC-P.
  - Every 10 seconds, the NSP-PROXY also sends to NRC-P a message which contains a Latch Reference ID (LRID) only.
  - The active VSR-NRC maintains a latch in Version-DB for each LRID sent to NRC-P. This latch is used to identify the start point for the SYNC_START message processing based on the LRID received from NRC-P.
- Enhancements to active VSR-NRC and NRC-P DB synchronization are as follows:
  1. After opening a PCEP or a BGP_LS service channel, the NRC-P in the primary site sends a SYNC_START with an LRID (empty for full sync) to the active VSR-NRC (VSR-NRC-A or VSR-NRC-B).
  2. The NSP-PROXY on the active VSR-NRC begins synchronization with the NRC-P of the Version-DB records, which include saved interleaved synchronization latches, from LRID specified in SYNC_START. An empty SYNC_START means the full content of the Version-DB is played back to NRC-P.

3. After processing all the records up to the specified LRID, NRC-P sends an acknowledgement of that LRID to NSP-PROXY using SYNC_ACK.
4. After initial synchronization, the NSP-PROXY on the active VSR-NRC resumes sending records from the main DB and inserts every 10 seconds an LRID message between the records it sends to NRC-P. The same stream of records interleaved with LRID messages is saved in the local Version-DB.
5. As in the initial synchronization phase, NRC-P acknowledges back an LRID to NSP-PROXY using a SYNC_ACK after the complete processing of all the records up to that LRID.

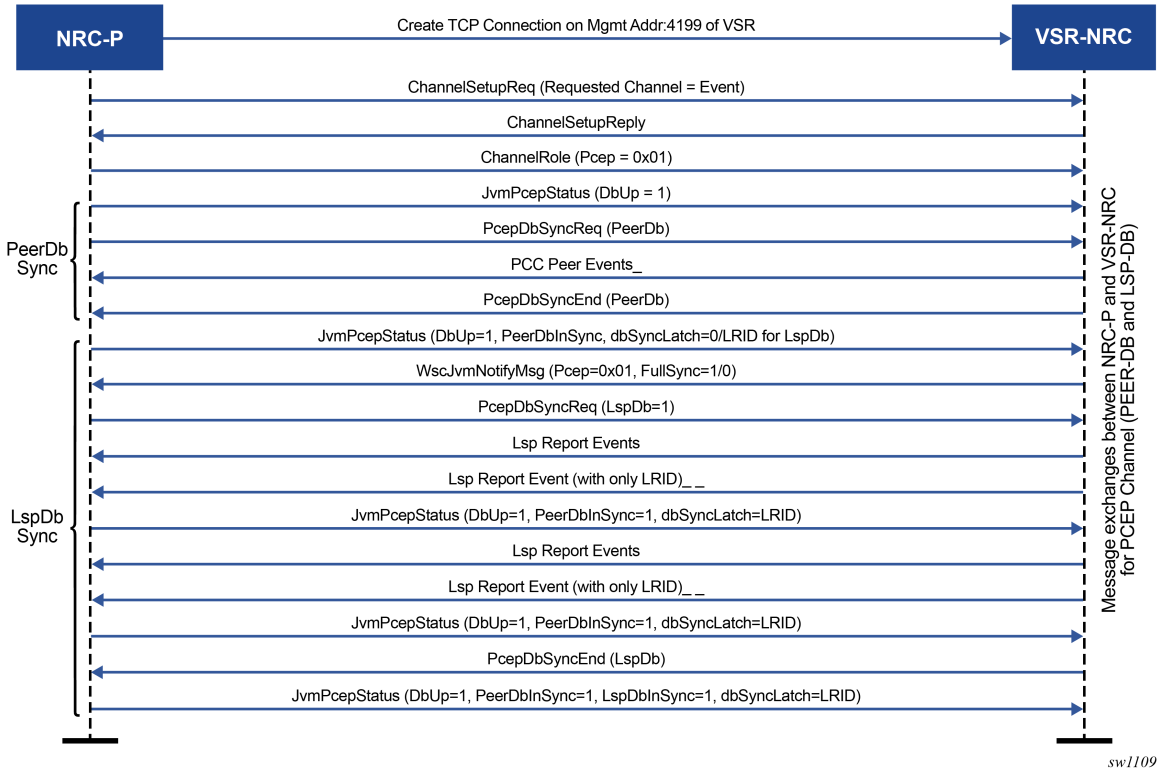
The following diagram shows the enhancements to the cproto protobuf messages exchanged between VSR-NRC and NRC-P to provide full or partial synchronization of the TE-DB.

Figure 63: TE-DB partial synchronization message sequence



The following diagram shows the enhancements to the cproto protobuf messages exchanged between VSR-NRC and NRC-P to provide full or partial synchronization of the LSP-DB.

Figure 64: LSP-DB partial synchronization message sequence



6.4 NSP and VSR-NRC PCE redundancy

This feature introduces resilience support to the PCE and PCC capabilities.

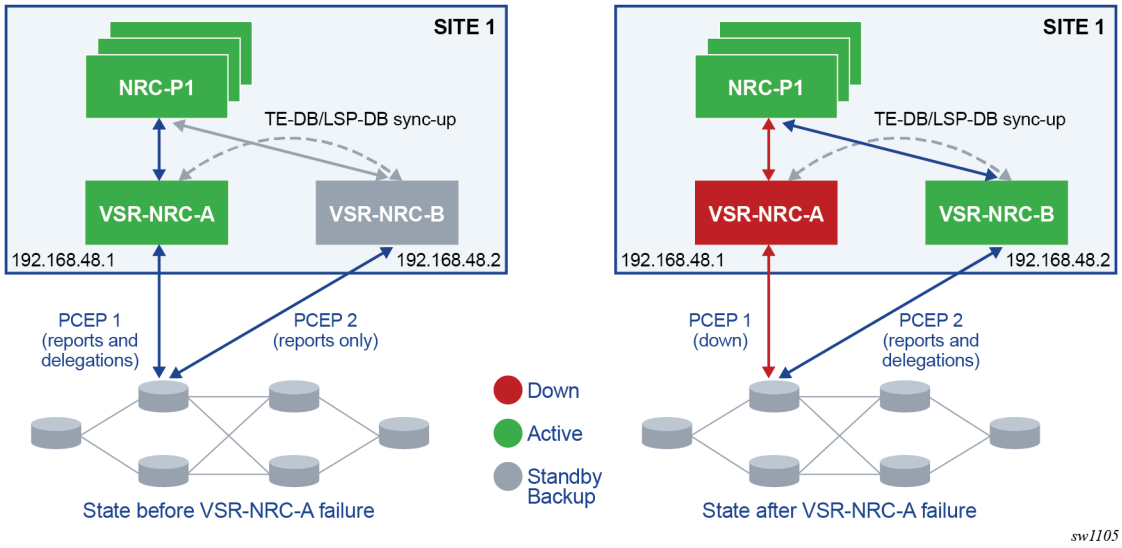
6.4.1 Overview of NSP ecosystem redundancy

The NSP ecosystem resilience consists of local, or single site, and remote, or dual site, redundancy mechanisms.

6.4.1.1 Redundancy in a single site deployment

The following diagram shows the NSP ecosystem and provisioning of redundancy within a single-site deployment.

Figure 65: NSP ecosystem redundancy in single-site deployment



The NSP, where the NRC-P component resides, is protected by a cluster of three Virtual Machines (VMs). This local redundancy scheme elects one VM as the active and the other two VMs become standby backups. NSP must always be deployed in a cluster of three VMs.

The VSR-NRC module runs the integrated VSR model (VSR-I) and can be deployed standalone.

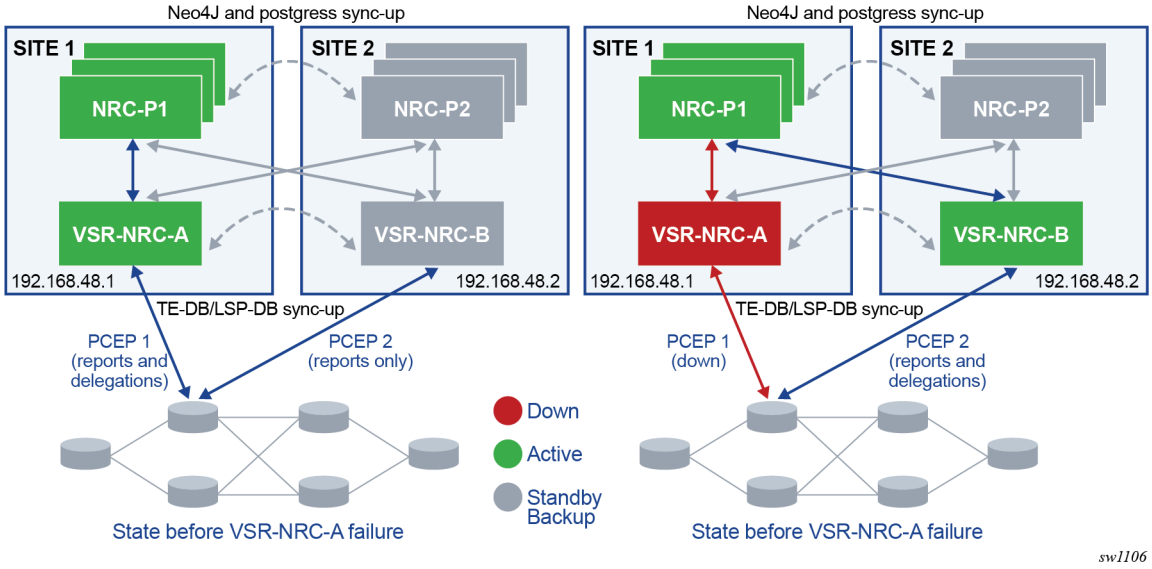
The VSR-NRC can be protected with a VM cluster implementing the 1+1 redundancy scheme. The local redundancy provides for continuous full and partial synchronization of the TE-DB and LS-DB between the active VSR-NRC and the backup standby VSR-NRC, as well as with the local NRC-P.

The details of the VSR-NRC 1+1 single-site redundancy mechanism are provided in VSR-NRC 1+1 single-site redundancy.

6.4.1.2 Redundancy in a dual site deployment

The following diagram shows the NSP ecosystem and provisioning of redundancy within a dual-site deployment.

Figure 66: NSP ecosystem redundancy in dual-site deployment



Both local and remote redundancy are deployed. The remote redundancy, sometimes referred to as Disaster Recovery (DR) or geo redundancy, consists of a primary site and a secondary backup site each with an NSP VM cluster and a single VSR-NRC VM.

A heartbeat protocol runs between the NSP clusters in the primary site and the standby backup sites.

The VSR-NRC connects to both the NRC-P within its own site and to the NRC-P in the remote site. A failover to the remote VSR-NRC occurs when the primary site fails entirely, when the primary NRC-P fails, and when the VSR-NRC only fails.

TE-DB and LSP-DB full and partial synchronization among the cluster of two VSR-NRCs improves the coverage of remote redundancy. In addition, the VSR-NRC 1+1 redundancy scheme is extended to the remote site. The details of the VSR-NRC 1+1 dual site redundancy are provided in VSR-NRC dual-site redundancy.

6.4.2 PCC and PCE redundancy configuration

The following CLI command enables the configuration on the PCC of a second PCEP session to the secondary backup PCE peer. A **preference** parameter value is used to indicate the primary or the secondary backup PCE peer role:

```
configure router pcep pcc peer ip-address [preference preference]
```

A maximum of two PCE peers are supported. The PCE peer that is not in overload is always selected by the PCC as the active PCE; however, if neither of the PCEs are signaling the overload state, the PCE with the higher numerical preference value is selected. In case of a tie, the PCE with the lower IP address is selected.

To change the value of the **preference** parameter, the peer must be deleted and recreated.

6.4.2.1 PCE in-band and out-of-band configuration management

The following CLI command configures the routing preference to reach the PCE server. Use this command to configure in-band reachability for the PCE when the BOF autoconfiguration is enabled and a wide static route, for example, 0.0.0.0/0, is assigned through DHCP.

```
configure router pcep pcc peer ip-address route-preference {both | inband | outband}
```

The command options are:

- **both**
  This option is the default value and specifies the use of the out-of-band routes in the management routing instance to reach the server before the in-band routes in the base routing instance.
- **inband**
  This option specifies the use of in-band routes only.
- **outband**
  This option specifies the use of out-of-band routes only.

The cproto channels are established through the management interface IPv4 or IPv6 address and open TCP port 4199 on both the primary and secondary VSR-NRCs.

In addition, the NRC-P always provides the active VSR-NRC acting as a cproto server with the system address of the mate VSR-NRC which to initiate the cproto channel to. The address is provided using the new Global Health and Notification cproto channel.

NRC-P provides a configuration for the primary VSR-NRC. This is the preferred active VSR-NRC. The other VSR-NRC is secondary. Nokia recommends setting the VSR-NRC co-located with the NRC-P as the primary VSR-NRC to take advantage of lowest latency and more reliable cproto channel.

In Figure 66: NSP ecosystem redundancy in dual-site deployment, the primary VSR-NRC in the local site is VSR-NRC-A and the secondary VSR-NRC in the remote site is VSR-NRC-B. The reverse configuration is performed in the remote site. With single-site VSR-NRC redundancy, both VSR-NRCs are local and either can be configured as the primary VSR-NRC.

6.4.3 NSP cluster redundancy

The following rules apply to the NSP cluster:

- At each site, a master is elected among the cluster of three VMs. In a DR deployment, the cluster in one site is designated as the primary, meaning it is the preferred active cluster. The site is referred to as the primary site. The second cluster and site are referred to as secondary and therefore act as the standby backup cluster or site.
- The application processes at the standby site are shut down, but the Neo4j and other databases are synchronized with the primary/active site.
- Switching to the standby site can be initiated manually or by using an automated approach stemming from the loss of heartbeat between the primary and standby sites.
- When the NSP cluster at the primary/active site is down (two out of three servers must be inactive, shut down, or failed), the heartbeat mechanism between the primary and standby NSP clusters fails after three timeouts. This initiates the activity at the inactive secondary/standby site.

- When the NSP cluster at the primary site is back up, the heartbeat mechanism between the primary/standby and secondary/active NSP clusters is restored. The primary site can be restored to the active site manually. Automatic reversion to the primary NSP cluster is not supported.

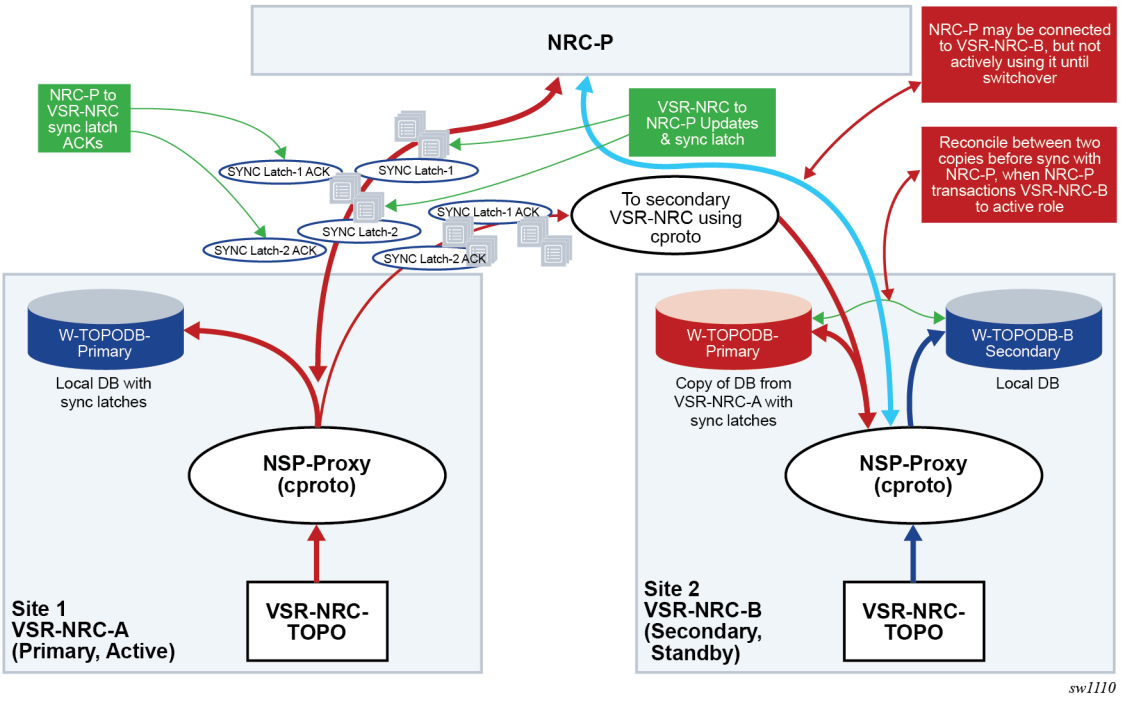
6.4.4 VSR-NRC 1+1 redundancy

This feature implements support for the single site or local 1+1 redundancy of the VSR-NRC.

6.4.4.1 VSR-NRC 1+1 single-site redundancy

Single-site or local 1+1 redundancy of the VSR-NRC relies on extending the communication of the NSP-PROXY to synchronize the contents of the TE-DB and LSP-DB between the primary VSR-NRC (VSR-NRC-A) and the secondary VSR-NRC (VSR-NRC-B). This is supported by the cproto sync channel running between the two VSR-NRCs. The following diagram shows VSR-NRC 1+1 single-site redundancy.

Figure 67: VSR-NRC 1+1 single-site redundancy



6.4.4.1.1 Initial establishment of active/standby VSR-NRC roles

The initial establishment of active/standby VSR-NRC roles is as follows:

1. NRC-P performs up to three attempts at 10-second intervals to establish the global cproto channel to the primary VSR-NRC (for example, VSR-NRC-A in the primary site). If not successful, it performs another three attempts to the secondary VSR-NRC (VSR-NRC-B in primary site). This process is continued cycling among the two VSR-NRCs until the global channel is established to either VSR-NRC-

A or VSR-NRC-B, which then becomes the target active VSR-NRC. The cproto channel establishment attempts continue to the other VSR-NRC which becomes the target standby VSR-NRC.
This cproto channel establishment process is also followed when the global cproto channel to the active VSR-NRC goes down beginning always with an attempt to the primary VSR-NRC (VSR-NRC-A) and then onto the secondary VSR-NRC (VSR-NRC-B) and continues cycling between them until a channel is established.

2. At successful global cproto channel establishment, the NRC-P sends a notify message (wscIsActive=TRUE, mateAddr, matePort) requesting a transition to active role from the VSR-NRC (VSR-NRC-A or VSR-NRC-B).
3. When NRC-P also established the TOPO and PCEP cproto channels, the active VSR-NRC begins the partial database synchronization procedures to NRC-P, as mentioned in TE-DB and LSP-DB partial synchronization.
4. When the reconcile process is complete, the active VSR-NRC acknowledges the NRC-P by sending a notify reply message (notifyReply=TRUE).
5. The active VSR-NRC attempts to establish a cproto sync channel to the mate VSR-NRC. After it is successfully established, it begins the full or partial database synchronization procedures to the mate VSR-NRC following the procedures, as mentioned in TE-DB and LSP-DB partial synchronization.
6. After the global cproto channel to the target standby VSR-NRC is established, NRC-P sends a notify message (wscIsActive=FALSE, mateAddr, matePort) requesting transition to a standby role. The target standby VSR-NRC acknowledges by sending a notify reply message (notifyReply=TRUE). No database synchronization occurs between the standby VSR-NRC and the NRC-P.
7. The standby VSR-NRC maintains a copy of the mate active VSR-NRC database and independently builds its own database using BGP-LS and PCEP peerings with the network.
8. When the active and standby roles have been assigned by the NRC-P, the primary and secondary VSR-NRCs keep that role until further notice from NRC-P. The roles are not affected by the state of the cproto sync channel.

6.4.4.1.2 Failover to backup VSR-NRC

The failover to backup VSR-NRC process is as follows:

1. If the active VSR-NRC fails, the NRC-P detects it when the global cproto channel goes down.
   The NRC-P uses a keepalive timer of 60 seconds and a multiplier of 2.2 for a total keepalive timeout of 132 seconds.

   At keepalive timeout, NRC-P determines that the channel is idle and closes it. NRC-P also closes the TOPO and PCEP cproto channels, if they are not already down.
2. Next, the NRC-P begins its cproto establishment cycle as detailed in Initial establishment of active/standby VSR-NRC roles. If the failed VSR-NRC is the primary VSR-NRC (VSR-NRC-A in the primary site), three attempts are performed to bring the global cproto channel back up. If the global cproto channel is successfully restored, the primary VSR-NRC remains the target active VSR-NRC. If not, three attempts are performed to the secondary VSR-NRC (VSR-NRC-B) and so on.

Note: It can take up to 162 seconds for the NRC-P to switch to the secondary standby VSR-NRC. This includes the keepalive idle time of 132 seconds plus up to three attempts of 10-second intervals to establish the global cproto channel, three unsuccessful attempts to primary VSR-NRC and one successful attempt to the secondary VSR-NRC. If the user reboots the primary VSR-NRC (VSR-NRC-A) while it is active, it may come back up faster, and therefore,

it remains the target active VSR-NRC and NRC-P does not switch to secondary VSR-NRC (VSR-NRC-B).

3. When the global cproto channel is up to the target active VSR-NRC, either the primary or secondary VSR-NRC, NRC-P sends a notify message requesting active role and providing the IP address for the mate VSR-NRC (wscIsActive=TRUE, mateAddr, matePort).
4. The target active VSR-NRC begins the transition to the active role by flapping the cproto sync session to the mate VSR-NRC, if not already down. Then it begins reconciling the local copy of the mate databases with its own network-learned databases. The common records only have a difference in the setting of the Delegate bit in the PCEP Report messages.

   The following reconcile process is followed:

   a. Common records of the local database are carried over with setting of the Delegate bit in the PCEP Report messages and with the LRID information from the mate database.
   b. Records in the mate database which are not reconciled with the local database are deleted.
   c. Records in the local database which are not reconciled with the mate database are always carried over.

   Common records of the local database are preferred over those of the mate copy database except for the LRID. After the reconcile process is complete, the now newly active VSR-NRC destroys the mate copy database and acknowledges the NRC-P by sending a notify reply message (notifyReply=TRUE).
5. The newly active VSR-NRC begins the partial database synchronization procedures to NRC-P as described in TE-DB and LSP-DB partial synchronization. The newly active VSR-NRC stops accepting new records from its own TE-DB and LSP-DB until after it completes the reconcile between the mate copy databases and its local databases and completes the partial synchronization with NRC-P.
6. The newly active VSR-NRC notifies the local PCE process to set the PCEP overload to OFF and to start an overload timer, hard-coded to 10 minutes, for each PCEP session. At the receipt of the first PCEP redelegation from a PCC, VSR-NRC stops the timer for that PCC and sends a PCC ready message to NRC-P, which can then begin sending update messages to that PCC. If the overload timer expires before receiving a PCEP redelegation message, the newly active VSR-NRC clears all delegations of the corresponding PCC toward NRC-P.
7. The newly active VSR-NRC attempts to establish a cproto channel to the mate VSR-NRC. After successfully establishing a channel, it begins the full or partial database synchronization procedures to the mate VSR-NRC following similar procedures as mentioned in TE-DB and LSP-DB partial synchronization.

6.4.4.1.3 Recovery of the failed VSR-NRC

The recovery of the failed VSR-NRC process is as follows:

1. The recovered VSR-NRC assumes the role of None, meaning it is neither active nor standby, until the global channel is successfully established from the NRC-P. In this state, it does not accept a cproto sync channel from its mate VSR-NRC.
2. After successfully opening a global cproto channel, NRC-P sends a notify message (wscIsActive=FALSE, mateAddr, matePort) to the recovered VSR-NRC to request transition to standby role.
3. The recovered VSR-NRC then accepts the cproto sync channel from its mate and prepares the mate copy DBs to accept the updates from its mate VSR-NRC.

4. VSR-NRC sends to NRC-P a notify reply message (notifyReply=TRUE) to accept the standby role assigned by NSP.
5. The newly standby VSR-NRC notifies the local PCE process to set the PCEP overload to ON.
6. NRC-P keeps this global cproto channel alive by sending KAs at regular intervals. No other service cproto channel is created while this VSR-NRC is in standby role.
7. NRC-P does not automatically revert the active role to the recovered VSR-NRC. A manual reversion procedure is supported. See Manual switchover to the mate VSR-NRC for the reversion procedure.

6.4.4.1.4 Manual switchover to the mate VSR-NRC

The NRC-P provides an API to perform a manual switchover to the mate VSR-NRC. This could be used, for example, to revert the active role back to a recovered primary VSR-NRC.

1. NRC-P sends a notify message (wscIsActive=FALSE, mateAddr, matePort) over the global channel to currently active VSR-NRC to request transition to standby role.
2. The currently active VSR-NRC shuts down the cproto sync channel to its mate and prepares the mate copy DBs to accept the updates from its mate VSR-NRC.
3. The currently active VSR-NRC sends to NRC-P a notify reply message (notifyReply=TRUE) to accept the standby role requested.
4. NRC-P then follows the steps in Initial establishment of active/standby VSR-NRC roles to transition the currently standby VSR-NRC to active role.

6.4.4.2 VSR-NRC dual-site redundancy

The behavior of dual-site redundancy follows the single site redundancy procedures because only the active NRC-P can establish cproto channels to the pair of primary and secondary VSR-NRCs.

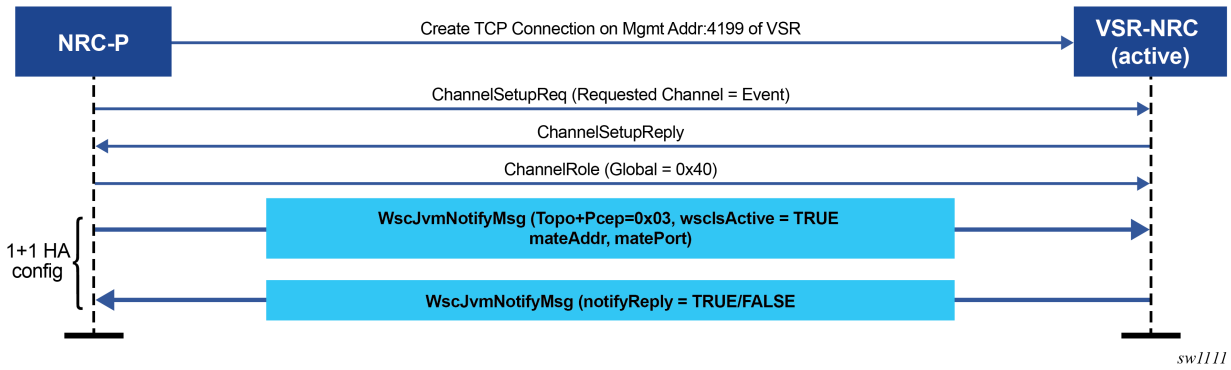
When the NRC-P in the secondary site becomes active, it attempts to establish a cproto global channel to both primary and secondary VSR-NRCs. Nokia recommends configuring the local VSR-NRC (VSR-NRC-B) as the primary VSR-NRC in the secondary site.

6.4.4.3 Global health and notification cproto channel

The VSR-NRC 1+1 redundancy features introduces a global channel between each VSR-NRC and the NRC-P for exchanging channel health and notification messages that are not application-specific.

The following diagram shows the sequence of messages to establish the global health and notification channel as well as the sequence of messages used for establishing the VSR-NRC role of active or standby.

Figure 68: Global health and notification channel message sequence



6.4.5 PCE southbound and PCC behavior

6.4.5.1 PCE southbound behavior

The following describes VSR-NRC PCE redundancy rules:

- **steady state behavior**
  1. The PCC establishes a PCEP session to each of the primary active VSR-NRC and secondary standby VSR-NRC. The secondary standby VSR-NRC is either in the primary site with single site redundancy or at the secondary site with dual site redundancy, however, the secondary standby VSR-NRC sets PCEP sessions with the PCCs in the overload state. The VSR-NRC enters this PCEP overload state when its upstream cproto session to the NSP cluster is down, or is being instructed by the NRC-P to enter the standby state as described in VSR-NRC 1+1 redundancy.
  2. The VSR-NRC acting as a PCE signals the overload state to the PCCs in a PCEP notification message. While in the overload state, the VSR-NRC PCE accepts reports (PCRpt) without delegation but rejects requests (PCReq) and reject reports (PCRpt) with delegation. The VSR-NRC PCE also does not originate initiate messages (PCInitiate) and update messages (PCUpd).
  3. The secondary standby VSR-NRC maintains its BGP and IGP peerings with the network and updates its TE database as a result of any network topology changes.
- **primary active NSP cluster failure**

  When the NSP cluster at the primary active site is down (two out of three servers must be inactive, shut down, or failed), the heartbeat mechanism between the primary active and secondary standby NSP clusters fails. This initiates the NSP cluster activity at the secondary standby site.

  The following are the procedures on the VSR-NRC:

  1. The primary VSR-NRC detects cproto global channel failure and puts all its PCEP sessions to the PCCs into the overload state.
  2. The NRC-P in the NSP cluster at the secondary site follows the procedures in VSR-NRC 1+1 redundancy to transition the secondary VSR-NRC into active state.
  3. The VSR-NRC at the primary site must also return the delegation of all LSPs back to the PCCs by sending an empty LSP Update Request that has the Delegate flag set to 0 in accordance with RFC

8231. To accommodate third party PCE implementations which may not return delegations, each PCC concurrently revokes the delegation of its LSPs from the primary VSR-NRC PCE. This allows the PCCs to delegate all eligible LSPs, including PCE-initiated LSPs, to the PCE function in the VSR-NRC at the secondary site. If the entire primary active site fails, the PCE side procedure in this step does not apply.

- **VSR-NRC complex failure at the primary site (NSP server is still up)**

  A VSR-NRC complex failure at the primary active NSP triggers the failover to backup VSR-NRC procedures in VSR-NRC 1+1 redundancy.

6.4.5.2 PCC behavior

The following rules apply to the PCC:

- PCCs can establish upstream PCEP sessions with, at most, two VSR-NRC PCEs.
- Each upstream session has a preference that takes effect when both upstream PCEP sessions are successfully established. The PCE peer that is not in overload is always selected by the PCC as the active PCE. However, if neither of the PCEs are signaling the overload state, the PCE with the higher numerical preference value is selected, and, in case of a tie, the PCE with the lower IP address is selected.
- In the steady state, because one upstream VSR-NRC PCE is in overload, only one PCEP session is active. The PCCs delegate an LSP using a report message (PCRpt) with the Delegate flag set to the active VSR-NRC PCE only. PCReq messages are not sent to the secondary/standby VSR-NRC PCE in overload. PCRpt messages are sent with the Delegate flag clear to the secondary/standby VSR-NRC PCE in overload.
- If the current active PCEP session signals overload state, the PCC selects the other PCE as the active PCE, as long as the corresponding PCEP session is not in overload. A new PCReq message or PCRpt message, with the Delegate flag set, is sent to the new PCE.

  The PCE in overload returns the delegation of all existing LSPs back to this PCC by sending an empty LSP Update Request that has the Delegate flag set, as described in RFC 8231. To accommodate third party PCE implementations which may not return delegations, each PCC concurrently revokes the delegation of its LSPs from the current PCE. This PCC then delegates these LSPs to the new active PCE by sending a path report (PCRpt) with the Delegate flag set.
- If the current active PCEP session goes operationally down, the PCC starts the redelegation timer (default 90 seconds) and state timer (default 180 seconds).
  - If the PCEP session is restored before the redelegation timer expires, no delegation change is performed and the LSP state is maintained.
  - Upon expiration of the redelegation timer, the PCC looks for the other PCEP session and, if not in overload, it immediately delegates the LSPs to the newly active PCE. If the new PCE accepts the delegation, the LSP state is maintained.
  - If the PCEP session does not recover before the state timer expires, and the PCC fails to find another active PCEP session, by default the PCC clears the LSP state of PCE-initiated LSPs after state timer expiry. The PCC deletes the PCE-initiated LSPs and releases all their resources. A configuration option of the state timer CLI command allows the user to keep the state of the PCE-initiated LSPs instead. The PCC does not clear the state of PCC-initiated LSPs; however, the user can do this by deleting the configuration.

6.5 VSR-NRC ROM

The VSR-NRC supports originating routes on behalf of the NRC-P toward route reflectors or BGP routers in the network. One of the most common applications is origination using BGP of SR policy candidate paths. The VSR-NRC ROM implements this capability. SR policy candidate paths are advertised as **sr-policy-ipv4** BGP routes.

NRC-P sends SR policy candidate path routes to both the active and secondary backup VSR-NRCs using the cproto ROM channel. NRC-P tracks the last route sent to each VSR-NRC separately. This tracking includes whether the last route was route ADD, DELETE, or MODIFY.

The VSR-NRC uses a cache to store SR policy route origination requests from the NRC-P. Upon receiving an SR policy route origination request, the VSR-NRC submits the SR policy route ADD or DELETE to the SR policy manager. ROM handles route ADD as a route MODIFY when there is already a record of the policy candidate path in its cache. The interface from ROM to the SR policy manager database uses <color, endpoint, headend> as the key in the lookup to identify the SR policy and uses <preference, rd> as a parameter to identify a specific candidate path of the SR policy.

The SR policy manager in VSR-NRC saves and organizes all submitted SR policy routes according to owner. Owner STATIC is used for routes submitted using the CLI and the management interface, owner BGP is used for routes submitted using the BGP control plane, and owner NSP is used for routes submitted by ROM.

When a cproto ROM channel goes down, ROM starts a timer with a value equal to the cproto keep alive expiry timeout value of 60 × 2.2 seconds. When this timer expires, ROM scans through all NSP owner records in its route cache and requests the SR policy manager database to flush them. As a result, the BGP module is called upon to withdraw the corresponding NLRIs from the BGP peers in the network.

Because NRC-P switches to the backup VSR-NRC when the cproto ROM channel goes down, the flush operation is necessary. However, if the originally active VSR-NRC is still up, it may remain the preferred source for the routes selected by the BGP peer routers in the network while its own SR policy manager routes are out-of-date with those of the NRC-P.

The NRC-P also supports sending an explicit flush request while the cproto ROM channel is up. This is used to clear stale route entries caused by multiple redundancy switchover operations.

When a cproto ROM channel comes back up, ROM in VSR-NRC sends a Notify message to NRC-P to indicate that the route cache and SR policy manager database were flushed. Subsequently, NRC-P checks the VSR-NRC uptime. If it is lower than the downtime of the cproto ROM channel to the VSR-NRC, NRC-P assumes the VSR-NRC rebooted and sends the full SR policy route database to ROM (full synchronization). If the VSR-NRC uptime is higher than the downtime of the cproto ROM channel to the VSR-NRC, NRC-P sends only the newer records (partial synchronization).

6.6 Configuring and operating RSVP-TE LSP with PCEP

This section provides information about configuring and operating RSVP-TE LSP with PCEP using CLI.

The following describes the detailed configuration of an inter-area RSVP-TE LSP with both a primary path and a secondary path. The network uses IS-IS with the backbone area in Level 2 and the leaf areas in Level 1. Topology discovery is learned by NRC-P using BGP-LS.

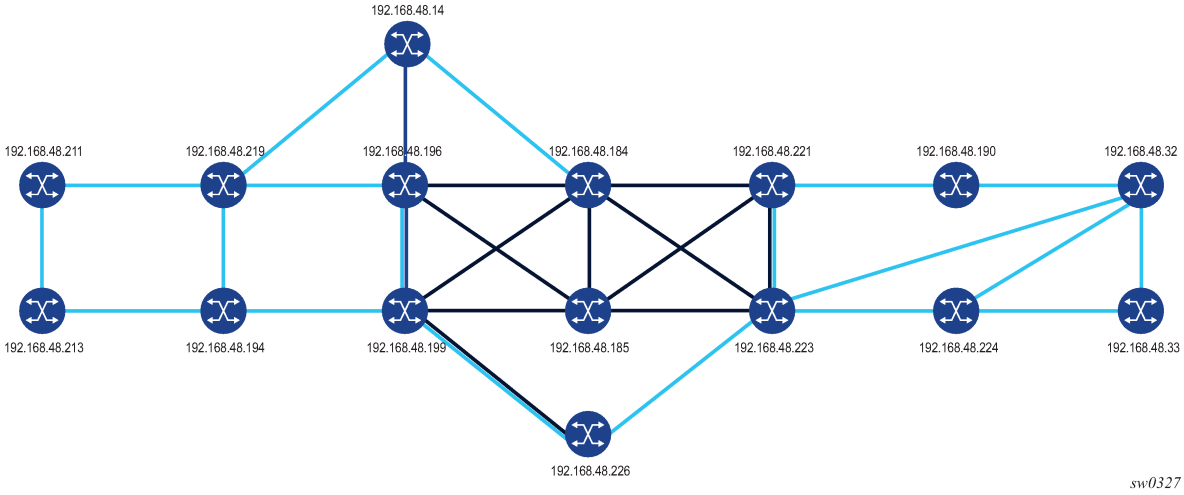
The LSP uses an admin-group constraint to keep the paths of the secondary and primary link disjoint in the backbone area. The LSP is PCE-controlled but also has **path-computation-method pce** enabled so the initial path, and any MBB path, is also computed by PCE.

The NSP and SR OS load versions used to produce this example are:

- For NSP, NSP-2.0.3-rel.108
- For PCE SR OS, TiMOS-B-0.0.W129
- For PCC, TiMOS-B-0.0.I4902

The following diagram shows a multilevel IS-IS topology in the NSP GUI.

Figure 69: Multilevel IS-IS topology in the NSP GUI



Example: Configuration and show command output of the PCEP on the PCE node and the PCC node

```
*A:PCE Server 226>config>router>pcep>pce# info
----------------------------------------------
            local-address 192.168.48.226
            no shutdown
----------------------------------------------
*A:Reno 194>config>router>pcep>pcc# info
----------------------------------------------
            peer 192.168.48.226
                no shutdown
            exit
            no shutdown
----------------------------------------------

*A:PCE Server 226>config>router>pcep>pce# show router pcep pce status
===============================================================================
Path Computation Element Protocol (PCEP) Path Computation Element (PCE) Info
===============================================================================
Admin Status           : Up              Oper Status          : Up
Unknown Msg Limit      : 10 msg/min
Keepalive Interval     : 30 seconds      DeadTimer Interval   : 120 seconds
Capabilities List      : stateful-delegate stateful-pce segment-rt-path
Local Address          : 192.168.48.226
PCE Overloaded         : false
-------------------------------------------------------------------------------
```

```
PCEP Path Computation Element (PCE) Peer Info
-------------------------------------------------------------------------------
Peer                      Sync State            Oper Keepalive/Oper DeadTimer
-------------------------------------------------------------------------------
192.168.48.190:4189       done                  30/120
192.168.48.194:4189       done                  30/120
192.168.48.198:4189       done                  30/120
192.168.48.199:4189       done                  30/120
192.168.48.219:4189       done                  30/120
192.168.48.221:4189       done                  30/120
192.168.48.224:4189       done                  30/120
-------------------------------------------------------------------------------
===============================================================================

*A:Reno 194# show router pcep pcc status
===============================================================================
Path Computation Element Protocol (PCEP) Path Computation Client (PCC) Info
===============================================================================
Admin Status           : Up              Oper Status          : Up
Unknown Msg Limit      : 10 msg/min
Keepalive Interval     : 30 seconds      DeadTimer Interval   : 120 seconds
Capabilities List      : stateful-delegate stateful-pce segment-rt-path
Address                : 192.168.48.194
Report Path Constraints: True
-------------------------------------------------------------------------------
PCEP Path Computation Client (PCC) Peer Info
-------------------------------------------------------------------------------
Peer                      Admin State/Oper State Oper Keepalive/Oper DeadTimer
-------------------------------------------------------------------------------
192.168.48.226            Up/Up                 30/120
-------------------------------------------------------------------------------
===============================================================================

*A:Reno 194# show router pcep pcc lsp-db
===============================================================================
PCEP Path Computation Client (PCC) LSP Update Info
===============================================================================
PCEP-specific LSP ID: 11
LSP ID              : 14378            LSP Type            : rsvp-p2p
Tunnel ID           : 1                Extended Tunnel Id  : 192.168.48.194
LSP Name            : From Reno to Atlanta RSVP-TE::primary_empty
Source Address      : 192.168.48.194   Destination Address : 192.168.48.224
LSP Delegated       : True             Delegate PCE Address: 192.168.48.226
Oper Status         : active
-------------------------------------------------------------------------------
PCEP-specific LSP ID: 12
LSP ID              : 14380            LSP Type            : rsvp-p2p
Tunnel ID           : 1                Extended Tunnel Id  : 192.168.48.194
LSP Name            : From Reno to Atlanta RSVP-TE::secondary_empty
Source Address      : 192.168.48.194   Destination Address : 192.168.48.224
LSP Delegated       : True             Delegate PCE Address: 192.168.48.226
Oper Status         : up
===============================================================================
```

Example: Configuration and show command output of BGP on the PCE node and the ABR node-to-learn topology using the BGP-LS NLRI family

```
*A:PCE Server 226>config>router>bgp# info
----------------------------------------------
            family bgp-ls
            min-route-advertisement 1
            link-state-export-enable
            group "IBGP_L2"
```

```
                family bgp-ls
                peer-as 65000
                neighbor 192.168.48.198
                exit
                neighbor 192.168.48.199
                exit
                neighbor 192.168.48.221
                exit
            exit
            no shutdown
-----------------------------------------------

*A:Chicago 221>config>router>bgp# info
-----------------------------------------------
            min-route-advertisement 1
            advertise-inactive
            link-state-import-enable
            group "IBGP_L2"
                family bgp-ls
                peer-as 65000
                neighbor 192.168.48.226
                exit
            exit
            no shutdown
-----------------------------------------------

*A:PCE Server 226# show router bgp summary
===============================================================================
 BGP Router ID:192.168.48.226    AS:65000       Local AS:65000
===============================================================================
BGP Admin State         : Up          BGP Oper State              : Up
Total Peer Groups       : 1           Total Peers                 : 3
Total BGP Paths         : 182         Total Path Memory           : 44896
Total IPv4 Remote Rts   : 0           Total IPv4 Rem. Active Rts  : 0
Total McIPv4 Remote Rts : 0           Total McIPv4 Rem. Active Rts: 0
Total McIPv6 Remote Rts : 0           Total McIPv6 Rem. Active Rts: 0
Total IPv6 Remote Rts   : 0           Total IPv6 Rem. Active Rts  : 0
Total IPv4 Backup Rts   : 0           Total IPv6 Backup Rts       : 0
Total Supressed Rts     : 0           Total Hist. Rts             : 0
Total Decay Rts         : 0
Total VPN Peer Groups   : 0           Total VPN Peers             : 0
Total VPN Local Rts     : 0
Total VPN-IPv4 Rem. Rts : 0           Total VPN-IPv4 Rem. Act. Rts: 0
Total VPN-IPv6 Rem. Rts : 0           Total VPN-IPv6 Rem. Act. Rts: 0
Total VPN-IPv4 Bkup Rts : 0           Total VPN-IPv6 Bkup Rts     : 0
Total VPN Supp. Rts     : 0           Total VPN Hist. Rts         : 0
Total VPN Decay Rts     : 0
Total L2-VPN Rem. Rts   : 0           Total L2VPN Rem. Act. Rts   : 0
Total MVPN-IPv4 Rem Rts : 0           Total MVPN-IPv4 Rem Act Rts : 0
Total MDT-SAFI Rem Rts  : 0           Total MDT-SAFI Rem Act Rts  : 0
Total MSPW Rem Rts      : 0           Total MSPW Rem Act Rts      : 0
Total RouteTgt Rem Rts  : 0           Total RouteTgt Rem Act Rts  : 0
Total McVpnIPv4 Rem Rts : 0           Total McVpnIPv4 Rem Act Rts : 0
Total McVpnIPv6 Rem Rts : 0           Total McVpnIPv6 Rem Act Rts : 0
Total MVPN-IPv6 Rem Rts : 0           Total MVPN-IPv6 Rem Act Rts : 0
Total EVPN Rem Rts      : 0           Total EVPN Rem Act Rts      : 0
Total FlowIpv4 Rem Rts  : 0           Total FlowIpv4 Rem Act Rts  : 0
Total FlowIpv6 Rem Rts  : 0           Total FlowIpv6 Rem Act Rts  : 0
Total LblIpv4 Rem Rts   : 0           Total LblIpv4 Rem. Act Rts  : 0
Total LblIpv6 Rem Rts   : 0           Total LblIpv6 Rem. Act Rts  : 0
Total LblIpv4 Bkp Rts   : 0           Total LblIpv6 Bkp Rts       : 0
Total Link State Rem Rts: 271         Total Link State Rem. Act Rts: 0
===============================================================================
BGP Summary
```

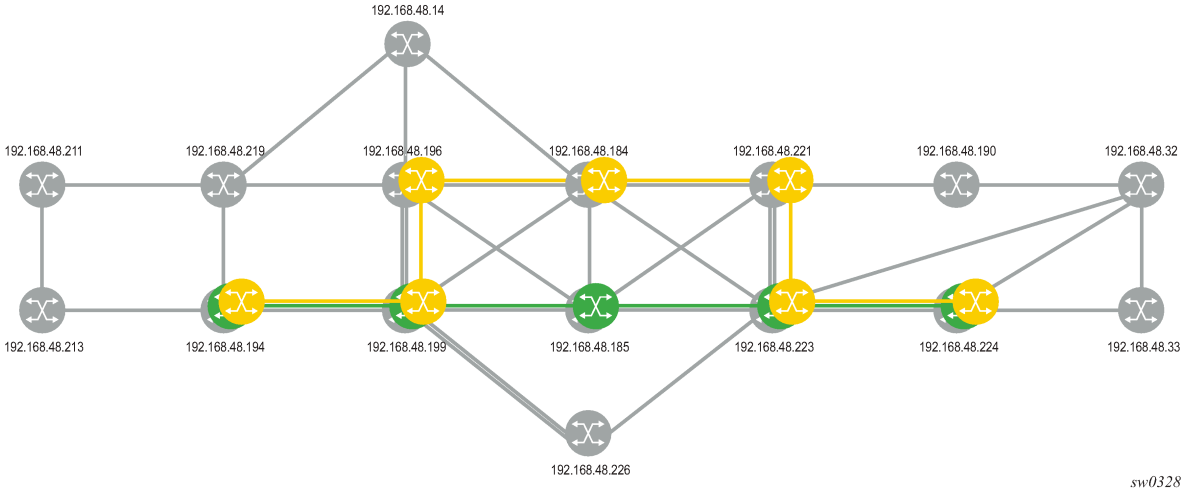
```
===============================================================================
Legend : D - Dynamic Neighbor
===============================================================================
Neighbor
Description
                  AS PktRcvd InQ  Up/Down   State|Rcv/Act/Sent (Addr Family)
                     PktSent OutQ
-------------------------------------------------------------------------------
192.168.48.198
               65000        0    0 02h42m56s Active
                            0    0
192.168.48.199
               65000      503    0 02h42m56s 76/0/0 (LinkState)
                          328    0
192.168.48.221
               65000      519    0 02h42m56s 195/0/0 (LinkState)
                          328    0
-------------------------------------------------------------------------------

*A:PCE Server 226# show router bgp routes bgp-ls hunt link
===============================================================================
 BGP Router ID:192.168.48.226    AS:65000       Local AS:65000
===============================================================================
 Legend -
 Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                 l - leaked, x - stale, > - best, b - backup, p - purge
 Origin codes  : i - IGP, e - EGP, ? - incomplete
===============================================================================
BGP-LS Link NLRIs
===============================================================================
-------------------------------------------------------------------------------
RIB In Entries
-------------------------------------------------------------------------------
Network:
 Type          : LINK-NLRI
 Protocol      : ISIS Level-2           Identifier     : 0xa
 Local Node descriptor:
  Autonomous System  : 0.0.253.232
  Link State Id      : 10
  IGP Router Id      : 0x38120048184
 Remote Node descriptor:
  Autonomous System  : 0.0.253.232
  Link State Id      : 10
  IGP Router Id      : 0x38120048223
 Link descriptor:
  IPV4 Interface Addr: 10.0.14.184
  IPV4 Neighbor Addr : 10.0.14.223
Nexthop        : 192.168.48.199
From           : 192.168.48.199
Res. Nexthop   : 0.0.0.0
Local Pref.    : 100                    Interface Name : NotAvailable
Aggregator AS  : None                   Aggregator     : None
Atomic Aggr.   : Not Atomic             MED            : None
AIGP Metric    : None
Connector      : None
Community      : No Community Members
Cluster        : No Cluster Members
Originator Id  : None                   Peer Router Id : 192.168.48.199
Flags          : Valid  Best  IGP
Route Source   : Internal
AS-Path        : No As-Path
Route Tag      : 0
Neighbor-AS    : N/A
Orig Validation: N/A
```

```
Source Class   : 0                      Dest Class     : 0
Add Paths Send : Default
Last Modified  : 02h27m50s
-------------------------------------------------------------------------------
Link State Attribute TLVs :
 Administrative group (color) : 0x0
 Maximum link bandwidth : 100000 Kbps
 Max. reservable link bandwidth : 100000 Kbps
 Unreserved bandwidth0 : 100000 Kbps
 Unreserved bandwidth1 : 100000 Kbps
 Unreserved bandwidth2 : 100000 Kbps
 Unreserved bandwidth3 : 100000 Kbps
 Unreserved bandwidth4 : 100000 Kbps
 Unreserved bandwidth5 : 100000 Kbps
 Unreserved bandwidth6 : 100000 Kbps
 Unreserved bandwidth7 : 100000 Kbps
 TE Default Metric : 100
 IGP Metric : 100
 Adjacency Segment Identifier (Adj-SID) :       flags 0x30 weight 0 sid 262136
-------------------------------------------------------------------------------
Network:
 Type          : LINK-NLRI
 Protocol      : ISIS Level-2           Identifier     : 0xa
 Local Node descriptor:
  Autonomous System  : 0.0.253.232
  Link State Id      : 10
  IGP Router Id      : 0x38120048184
 Remote Node descriptor:
  Autonomous System  : 0.0.253.232
  Link State Id      : 10
  IGP Router Id      : 0x38120048223
 Link descriptor:
  IPV4 Interface Addr: 10.0.14.184
  IPV4 Neighbor Addr : 10.0.14.223
Nexthop        : 192.168.48.221
From           : 192.168.48.221
Res. Nexthop   : 0.0.0.0
Local Pref.    : 100                    Interface Name : NotAvailable
Aggregator AS  : None                   Aggregator     : None
Atomic Aggr.   : Not Atomic             MED            : None
AIGP Metric    : None
Connector      : None
Community      : No Community Members
Cluster        : No Cluster Members
Originator Id  : None                   Peer Router Id : 192.168.48.221
Flags          : Valid  IGP
TieBreakReason : OriginatorID
Route Source   : Internal
AS-Path        : No As-Path
Route Tag      : 0
Neighbor-AS    : N/A
Orig Validation: N/A
Source Class   : 0                      Dest Class     : 0
Add Paths Send : Default
Last Modified  : 02h27m54s
-------------------------------------------------------------------------------
Link State Attribute TLVs :
 Administrative group (color) : 0x0
 Maximum link bandwidth : 100000 Kbps
 Max. reservable link bandwidth : 100000 Kbps
 Unreserved bandwidth0 : 100000 Kbps
 Unreserved bandwidth1 : 100000 Kbps
 Unreserved bandwidth2 : 100000 Kbps
 Unreserved bandwidth3 : 100000 Kbps
```

```
 Unreserved bandwidth4 : 100000 Kbps
 Unreserved bandwidth5 : 100000 Kbps
 Unreserved bandwidth6 : 100000 Kbps
 Unreserved bandwidth7 : 100000 Kbps
 TE Default Metric : 100
 IGP Metric : 100
 Adjacency Segment Identifier (Adj-SID) :       flags 0x30 weight 0 sid 262136
-------------------------------------------------------------------------------
```

The following diagram shows primary and secondary RSVP-TE LSP paths in the NSP GUI.

Figure 70: Primary and secondary RSVP-TE LSP paths in the NSP GUI



Example: Configuration and show command output of the MPLS on the PCC node

```
*A:Reno 194>config>router>mpls>lsp# info
----------------------------------------------
                to 192.168.48.224
                egress-statistics
                    shutdown
                exit
                fast-reroute facility
                    no node-protect
                exit
                path-computation-method pce
                pce-report enable
                pce-control
                revert-timer 1
                primary "primary_empty"
                    exclude "top"
                    bandwidth 10
                exit
                secondary "secondary_empty"
                    standby
                    exclude "bottom"
                    bandwidth 5
                exit
                no shutdown
----------------------------------------------

*A:Reno 194# show router mpls lsp "From Reno to Atlanta RSVP-TE" path detail
===============================================================================
```

```
MPLS LSP From Reno to Atlanta RSVP-TE Path  (Detail)
===============================================================================
Legend :
    @ - Detour Available              # - Detour In Use
    b - Bandwidth Protected           n - Node Protected
    s - Soft Preemption
    S - Strict                        L - Loose
    A - ABR
===============================================================================
-------------------------------------------------------------------------------
LSP From Reno to Atlanta RSVP-TE Path primary_empty
-------------------------------------------------------------------------------
LSP Name         : From Reno to Atlanta RSVP-TE
Path LSP ID      : 14382
From             : 192.168.48.194       To                  : 192.168.48.224
Admin State      : Up                   Oper State          : Up
Path Name        : primary_empty        Path Type           : Primary
Path Admin       : Up                   Path Oper           : Up
Out Interface    : 1/1/1                Out Label           : 262094
Path Up Time     : 0d 00:00:22          Path Down Time      : 0d 00:00:00
Retry Limit      : 0                    Retry Timer         : 30 sec
Retry Attempt    : 0                    Next Retry In       : 0 sec
BFD Template     : None                 BFD Ping Interval   : 60
BFD Enable       : False
Adspec           : Disabled             Oper Adspec         : Disabled
CSPF             : Enabled              Oper CSPF           : Enabled
Least Fill       : Disabled             Oper LeastFill      : Disabled
FRR              : Enabled              Oper FRR            : Enabled
FRR NodeProtect  : Disabled             Oper FRR NP         : Disabled
FR Hop Limit     : 16                   Oper FRHopLimit     : 16
FR Prop Admin Gr*: Disabled             Oper FRPropAdmGrp   : Disabled
Propogate Adm Grp: Disabled             Oper Prop Adm Grp   : Disabled
Inter-area       : False
PCE Updt ID      : 0
PCE Report       : Enabled              Oper PCE Report     : Enabled
PCE Control      : Enabled              Oper PCE Control    : Enabled
PCE Compute      : Enabled
Neg MTU          : 1496                 Oper MTU            : 1496
Bandwidth        : 10 Mbps              Oper Bandwidth      : 10 Mbps
Hop Limit        : 255                  Oper HopLimit       : 255
Record Route     : Record               Oper Record Route   : Record
Record Label     : Record               Oper Record Label   : Record
Setup Priority   : 7                    Oper Setup Priority : 7
Hold Priority    : 0                    Oper Hold Priority  : 0
Class Type       : 0                    Oper CT             : 0
Backup CT        : None
MainCT Retry     : n/a
    Rem          :
MainCT Retry     : 0
    Limit        :
Include Groups   :                      Oper Include Groups :
None                                          None
Exclude Groups   :                      Oper Exclude Groups :
top                                           top
Adaptive         : Enabled              Oper Metric         : 40
Preference       : n/a
Path Trans       : 7                    CSPF Queries        : 7172
Failure Code     : noError
Failure Node     : n/a
Explicit Hops    :
    No Hops Specified
Actual Hops      :
    10.202.5.194 (192.168.48.194) @              Record Label       : N/A
 -> 10.202.5.199 (192.168.48.199) @              Record Label       : 262094
```

```
 -> 192.168.48.185 (192.168.48.185)              Record Label        : 262111
 -> 10.0.5.185                                   Record Label        : 262111
 -> 192.168.48.223 (192.168.48.223)              Record Label        : 262121
 -> 10.0.7.223                                   Record Label        : 262121
 -> 192.168.48.224 (192.168.48.224)              Record Label        : 262116
 -> 10.101.4.224                                 Record Label        : 262116
Computed Hops    :
    10.202.5.199(S)
 -> 10.0.5.185(S)
 -> 10.0.7.223(S)
 -> 10.101.4.224(S)
Resignal Eligible: False
Last Resignal    : n/a                  CSPF Metric          : 40
-------------------------------------------------------------------------------
LSP From Reno to Atlanta RSVP-TE Path secondary_empty
-------------------------------------------------------------------------------
LSP Name         : From Reno to Atlanta RSVP-TE
Path LSP ID      : 14384
From             : 192.168.48.194       To                   : 192.168.48.224
Admin State      : Up                   Oper State           : Up
Path Name        : secondary_empty      Path Type            : Standby
Path Admin       : Up                   Path Oper            : Up
Out Interface    : 1/1/1                Out Label            : 262091
Path Up Time     : 0d 00:00:25          Path Down Time       : 0d 00:00:00
Retry Limit      : 0                    Retry Timer          : 30 sec
Retry Attempt    : 0                    Next Retry In        : 0 sec
BFD Template     : None                 BFD Ping Interval    : 60
BFD Enable       : False
Adspec           : Disabled             Oper Adspec          : Disabled
CSPF             : Enabled              Oper CSPF            : Enabled
Least Fill       : Disabled             Oper LeastFill       : Disabled
Propogate Adm Grp: Disabled             Oper Prop Adm Grp    : Disabled
Inter-area       : False
PCE Updt ID      : 0
PCE Report       : Enabled              Oper PCE Report      : Enabled
PCE Control      : Enabled              Oper PCE Control     : Enabled
PCE Compute      : Enabled
Neg MTU          : 1496                 Oper MTU             : 1496
Bandwidth        : 5 Mbps               Oper Bandwidth       : 5 Mbps
Hop Limit        : 255                  Oper HopLimit        : 255
Record Route     : Record               Oper Record Route    : Record
Record Label     : Record               Oper Record Label    : Record
Setup Priority   : 7                    Oper Setup Priority  : 7
Hold Priority    : 0                    Oper Hold Priority   : 0
Class Type       : 0                    Oper CT              : 0
Include Groups   :                      Oper Include Groups  :
None                                           None
Exclude Groups   :                      Oper Exclude Groups  :
bottom                                         bottom
Adaptive         : Enabled              Oper Metric          : 60
Preference       : 255
Path Trans       : 28                   CSPF Queries         : 10
Failure Code     : noError
Failure Node     : n/a
Explicit Hops    :
    No Hops Specified
Actual Hops      :
    10.202.5.194 (192.168.48.194)               Record Label        : N/A
 -> 10.202.5.199 (192.168.48.199)               Record Label        : 262091
 -> 10.0.9.198 (192.168.48.198)                 Record Label        : 262096
 -> 192.168.48.184 (192.168.48.184)             Record Label        : 262102
 -> 10.0.2.184                                  Record Label        : 262102
 -> 192.168.48.221 (192.168.48.221)             Record Label        : 262119
 -> 10.0.4.221                                  Record Label        : 262119
```

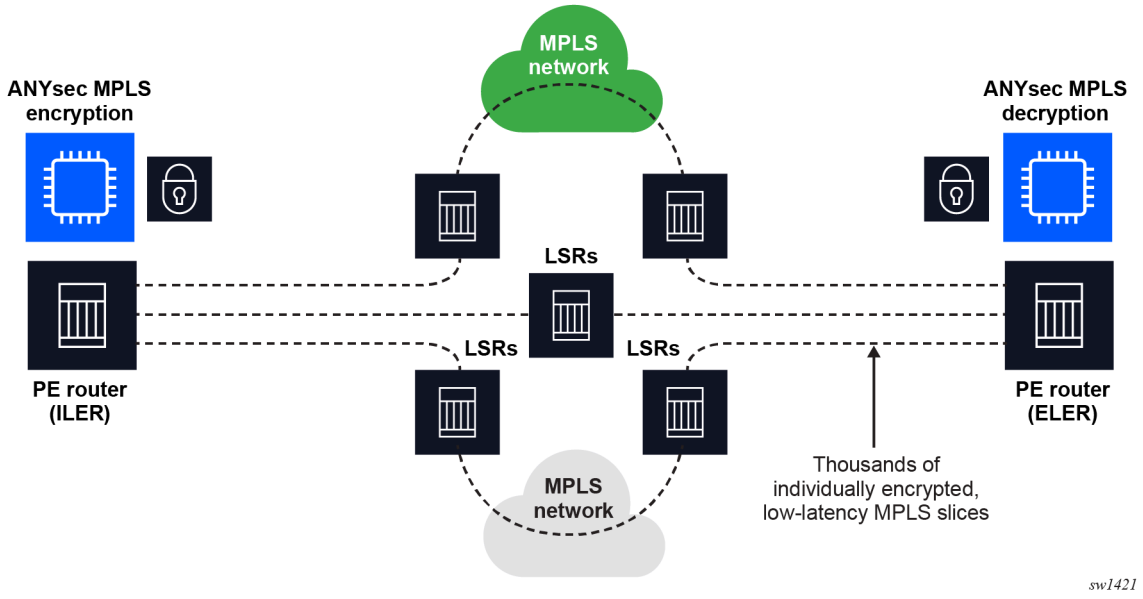
```
 -> 192.168.48.223 (192.168.48.223)              Record Label        : 262088
 -> 10.0.10.223                                  Record Label        : 262088
 -> 192.168.48.224 (192.168.48.224)              Record Label        : 262115
 -> 10.101.4.224                                 Record Label        : 262115
Computed Hops    :
    10.202.5.199(S)
 -> 10.0.9.198(S)
 -> 10.0.2.184(S)
 -> 10.0.4.221(S)
 -> 10.0.10.223(S)
 -> 10.101.4.224(S)
Srlg             : Disabled
Srlg Disjoint    : False
Resignal Eligible: False
Last Resignal    : n/a                  CSPF Metric          : 60
===============================================================================
```

7 ANYsec

Nokia ANYsec uses the IEEE802.1AE (MACsec) encryption engine in the Nokia FP5 chipset to encrypt MPLS payloads, while leaving the MPLS labels in clear and unauthenticated. After the MPLS payload is encrypted, the MPLS packet is switched through an MPLS network and eventually is decrypted using the encryption engine at the terminating FP5-enabled PE. Having the MPLS labels in clear and unauthenticated allows any LSR router to switch the ANYsec packets from the ILER ANYsec PE to the ELER ANYsec PE. Any LSR router, including third-party routers, can manipulate the MPLS header. This includes performing label actions such as label swap, pop, and push.

The following figure shows the ANYsec MPLS encryption and decryption process.

Figure 71: ANYsec MPLS encryption and decryption



7.1 ANYsec overview

Nokia ANYsec uses IEEE802.1AE (MACsec) as its datapath encryption engine and IEEE802.1x (dot1x/MKA) as its control plane signaling protocol. ANYsec uses the user-configured MACsec connectivity-association (CA) with a set of pre-shared keys (PSKs), the same way MACsec does, to negotiate datapath keys between two or more peers.

The following is the basic high-level overview of the SR OS ANYsec encryption process.

1. The user configures the ANYsec-specific CA, which supports the exclusive use of the CA with ANYsec encryption.

```
configure macsec connectivity-association anysec true
```

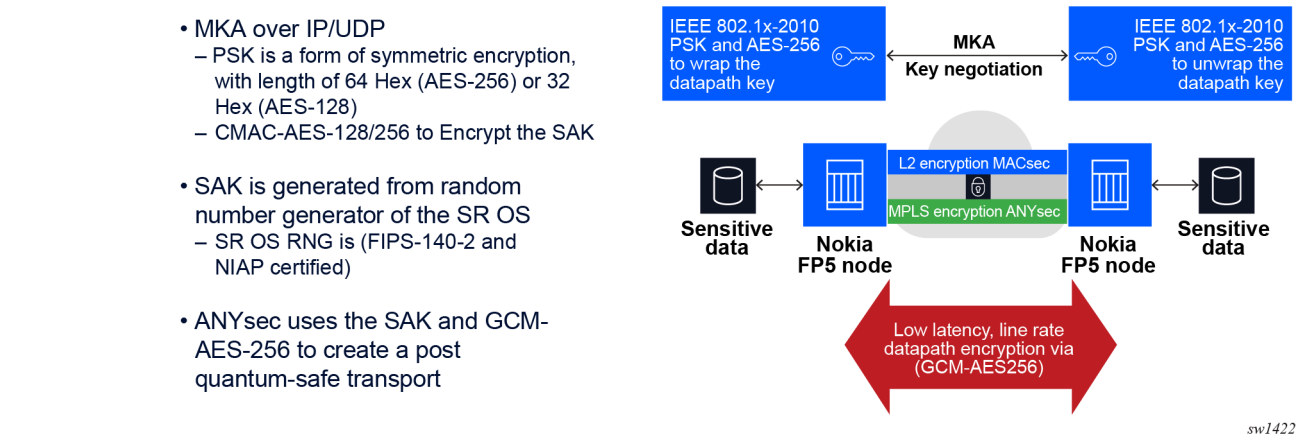
2. The MACsec key agreement (MKA) key server generates the datapath security association keys (SAKs) locally on the SR OS. The user configures the key-server selection priority using the following command.

```
configure macsec connectivity-association static-cak mka-key-server-priority
```

3. The MKA uses CMAC-AES-128/256 to encrypt the SAKs using PSK and distributes the SAKs.
4. ANYsec modifies the MKA for transport over IP/UDP (described in more detail in later topics). MKA distributes the SAKs between peers, and ANYsec uses the SAKs to encrypt the MPLS payload via the IEEE802.1AE encryption engine.

The following figure shows the ANYsec high-level implementation.

Figure 72: ANYsec high-level implementation

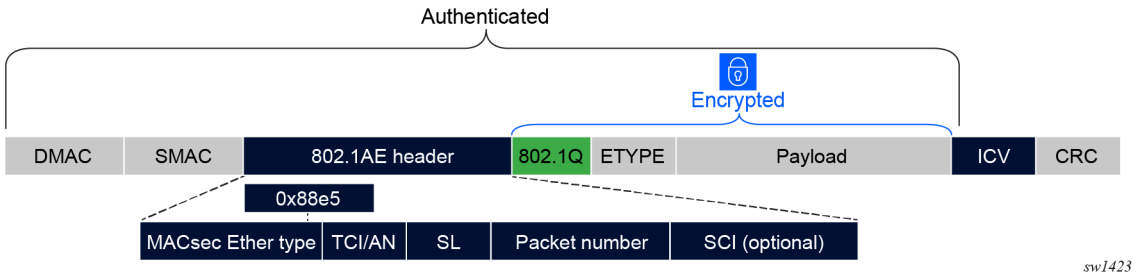


7.2 ANYsec packet format

ANYsec MPLS provides packet encryption formats that are different from the MACsec packet formats.

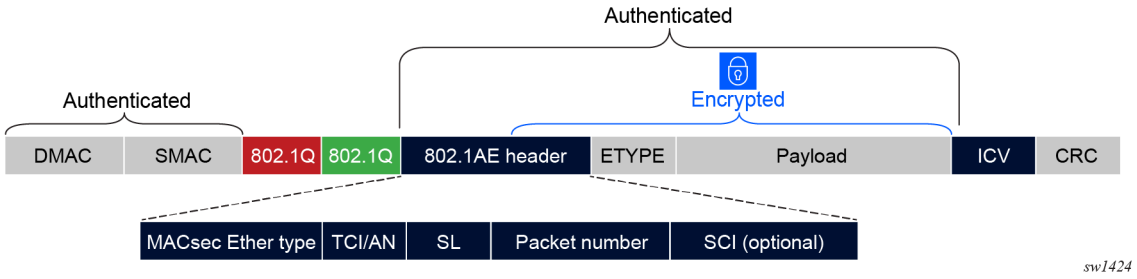
As shown in the following figure, MACsec leaves the MAC (802.1 AE) header in clear while authenticating the entire packet in the LAN.

Figure 73: MACsec packet authentication



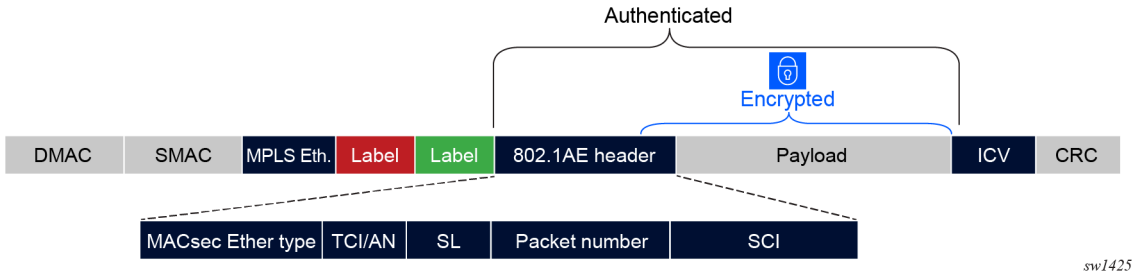
The following figure shows that the MACsec WAN mode leaves the dot1q VLANs in clear and unauthenticated, so the packets can be manipulated by the VLAN switched network.

Figure 74: MACsec packet authentication for dot1q packets



With ANYsec, the entire MPLS label stack is in clear and unauthenticated. This allows the LSR routers to manipulate the MPLS label stack while the system encrypts and authenticates the payload. The following figure shows ANYsec MPLS packet authentication.

Figure 75: ANYsec MPLS packet authentication



7.3 ANYsec encryption

Nokia FP5 ANYsec uses the IEEE802.1AE (MACsec) encryption engine to encrypt multiple layers of OSI. For example, the same encryption engine can encrypt at Layer 2, MACsec, or Layer 2.5 MPLS ANYsec.

ANYsec encryption supports MACsec encryption algorithms, MPLS protocols, and per-flow encryption.

7.3.1 Encryption algorithms

ANYsec uses the MACsec encryption engine and therefore supports all the MACsec encryption algorithms, including the following:

- CMAC-AES-128/AES-256 for pre-shared key encryption used by the MKA protocol
- GCM-AES 128/256 for datapath encryption, using extended packet number (XPN)

7.3.2 MPLS protocol support

ANYsec currently encrypts LSPs that are signaled by the following Segment Routing (SR) protocols:

- SR-ISIS
- SR-OSPF
- SR-OSPF3

Currently ANYsec encrypts the service label, entropy label, and entropy-indication label.

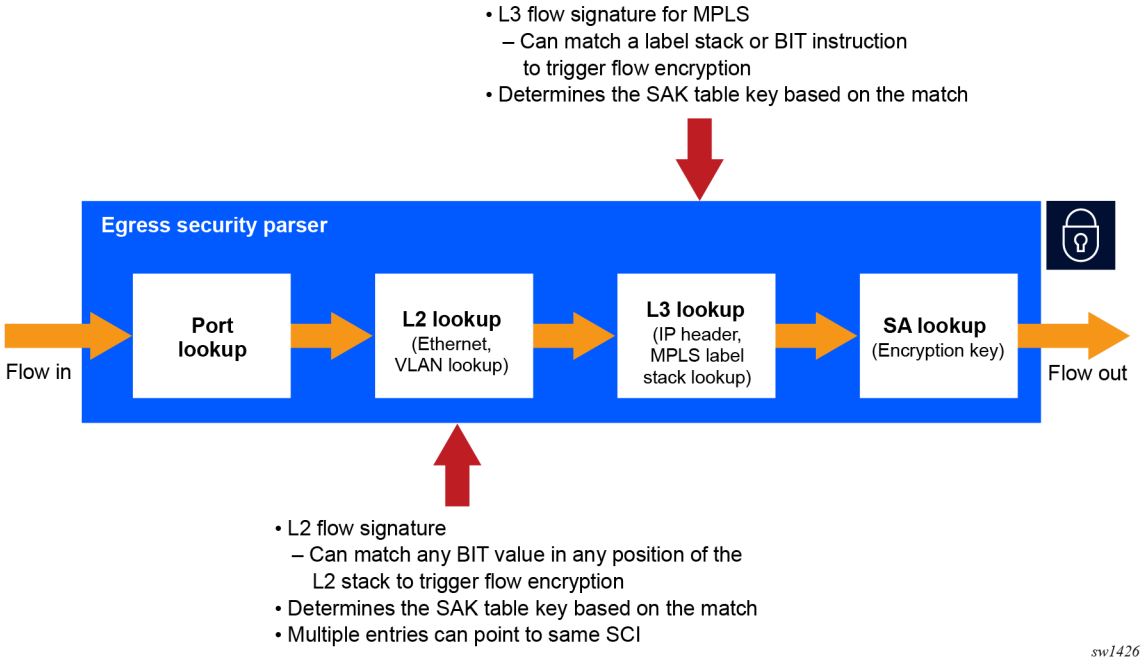
7.3.3 per-flow encryption

ANYsec uses a different encryption key for each encrypted LSP, thereby ensuring that all LSPs are encrypted with a unique SAK. The ANYsec MKA key server distributes the SAK to the peer. The user configures the key server selection identically to the MACsec key server selection, using the following command.

```
configure macsec connectivity-association static-cak mka-key-server-priority
```

ANYsec uses the Layer 3 content addressable memory (CAM) build in the Nokia FP5 chipset to match the MPLS label stack to the appropriate SAK. When a SAK for an LSP is distributed via MKA, ANYsec downloads the LSP's label stack to the Layer 3 CAM and associates it with the corresponding SAK, which it downloads to the SA lookup table.

Figure 76: ANYsec per-flow encryption

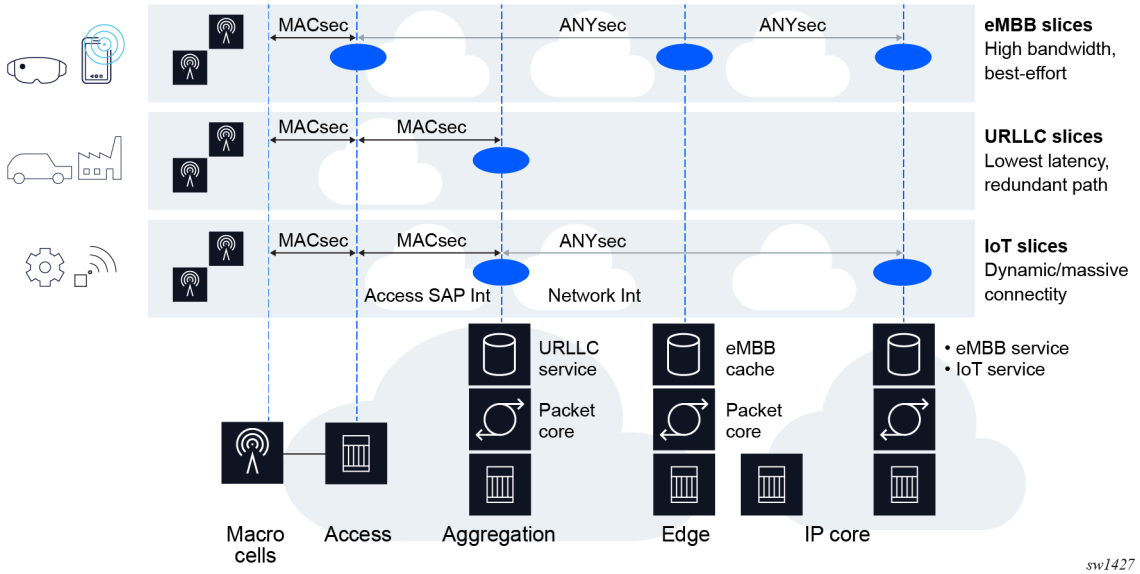


7.4 ANYsec and MACsec interaction

If ANYsec is enabled on the network ports, SR OS only supports enabling MACsec on the access ports. If MACsec and ANYsec are on the same IOM and FP5 chipset, MACsec uses scaling resources from ANYsec. For example, if a single MACsec port is configured on the same IOM as ANYsec, the ANYsec scale drops by 1 on that IOM. This is because both MACsec and ANYsec use the same SA lookup table, which has a defined number of entries.

The following figure shows the supported interaction between ANYsec and MACsec.

Figure 77: ANYsec and MACsec interaction



7.5 ANYsec and LAG and ECMP interaction

When LSPs are enabled with ANYsec on ILER, the LSPs do not hash their internal flows over multiple ECMP interfaces or over LAG members. Instead, each LSP on ILER maps to a dedicated ECMP interface or LAG member. If an ANYsec LSP is transporting multiple services or IP flows, all the services and IP flows map to a single interface in an ECMP group, or to a single member in a LAG bundle on the ILER. However, if there are multiple ANYsec LSPs on the ILER, each ANYsec LSP and all the flows within the LSP can also hash to a different interface in an ECMP group or a different LAG member in a LAG bundle.

The LSR routers can hash the ANYsec LSPs over an ECMP group interface or a LAG member freely without any restrictions.

Note: ANYsec does not support entropy label in clear or EL and ELI encryption. The CLI does not block the entropy-label-capable flag, but entropy label is not inserted in the datapath, and therefore EL and ELI is not present (either encrypted or unencrypted) in the actual packet, regardless of the configuration.

7.6 Inter AS and Inter Area solutions

ANYsec LSPs support a single-area solution only. ANYsec does not support inter-AS or inter-AREA option B or option C (for example, seamless MPLS).

ANYsec supports inter-AS or inter-AREA option A with the following conditions:

- The termination of the ANYsec LSP is at the ABR or ASBR router.

- Another ANYsec LSP originates toward the other inter-AS or inter-AREA option A.

In addition, ANYsec does not support tunnels within tunnels. For example, an ANYsec LSP that is encrypted cannot be encapsulated in a second MPLS tunnel such as BGP LU, RSVP-TE, or LDP.

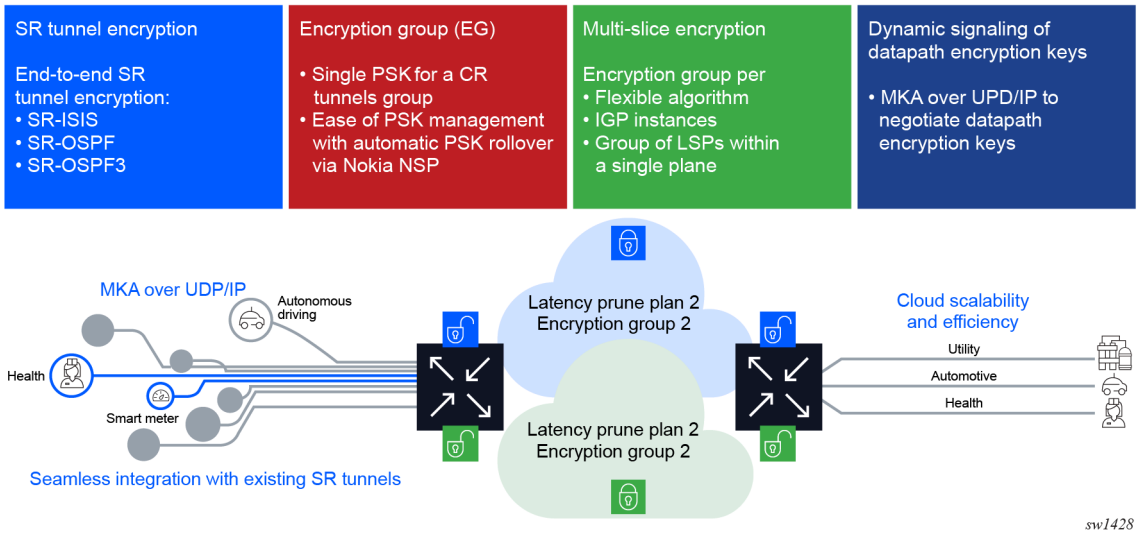
7.7 ANYsec implementation design

ANYsec supports the following implementation design:

- At the MKA signaling layer, ANYsec supports assigning the PSKs and the CAs to a group of Segment Routing (SR) tunnels. This assignment is known as Encryption Groups (EG) in ANYsec. EGs support easy management of PSKs for ANYsec tunnels, because multiple tunnels can use the same PSK.
- MKA is established per pair of peers, thereby allowing encryption of each tunnel with its own dedicated SAK. The use of a dedicated SAK per tunnel ensures a high level of security. For ease of management, ANYsec supports using PSKs per group of tunnels. However, for maximum security, the SAK is only supported per pair of peers.
- ANYsec supports tunnel slicing using Flex Algo or multi-instance IGP. Each slice is uniquely identified and a SAK is assigned per slice (per set of peers). ANYsec can also uniquely identify each LSP on each slice, and assign an EG to the LSPs through user configuration.
- ANYsec uses the MKA Layer 2 protocol for signaling, which uses IEEE 802.1x for encapsulation. To transport MKA over IP in a Layer 3 network, ANYsec encapsulates MKA over IP/UDP. The UDP port uniquely identifies the MKA packets to the ANYsec peers. The IP header transports the MKA packet from one ANYsec PE to another ANYsec PE.

The following figure shows the ANYsec design implementation and related requirements.

Figure 78: ANYsec encryption implementation design and requirements



7.8 ANYsec configuration guidelines

These topics describe the basic configuration requirements to implement Nokia ANYsec MPLS encryption.

7.8.1 Configuring ANYsec connectivity association and PSK

ANYsec uses MKA for its control plane and for signaling of SAKs, and therefore ANYsec reuses the MACsec connectivity association (CA) and the PSKs. SR OS supports creation of the CA uniquely for ANYsec. When the **anysec** command is enabled under the CA, SR OS supports the exclusive use of the CA with ANYsec encryption; this CA cannot be reused for MACsec any longer.

Use the following command to enable ANYsec CA.

```
configure macsec connectivity-association anysec
```

ANYsec supports the **static-cak** and the **cipher-suite** commands for the CA configuration. When a CA is configured for ANYsec, it does not support the following CA configurations:

- clear-tag mode
- delay protection
- encryption offset
- MACsec encryption
- replay-window size
- MAC policy

ANYsec also does not support configuration of a MACsec MAC policy.

The following example displays an ANYsec CA configuration.

Note: ANYsec supports a minimum **mka-hello-interval** of 5 seconds. This is because MKA hellos could transit over multiple routers from one ANYsec PE to another.

Example: MD-CLI

```
[ex:/configure macsec]
A:admin@node-2# info
    connectivity-association "CA-1" {
        admin-state enable
        cipher-suite gcm-aes-xpn-256
        anysec true
        static-cak {
            mka-hello-interval 5
            pre-shared-key 1 {
                encryption-type aes-256-cmac
                cak "2yzrsjg5sp7MYAnWpod+Nkn4SwXf7OPMEfAMRpNh9Gu/badNTWOoYEG9Qi1NDOBW
 hash2"
                cak-name "11"
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config>macsec# info
----------------------------------------------
        connectivity-association "CA-1" create.
            anysec
            cipher-suite gcm-aes-xpn-256
            static-cak
                pre-shared-key 1 encryption-type aes-256-cmac create
                    cak "2yzrsjg5sp7MYAnWpod+Nkn4SwXf7OPMEfAMRpNh9Gu/badNTWOoYEG9Qi1NDOBW"
 hash2
                    ckn "11"
                exit
                mka-hello-interval 5
            exit
            no shutdown
        exit
```

7.8.2 Identifying and configuring ANYsec LSP

After configuring the CA as described in Configuring ANYsec connectivity association and PSK, the next step is to identify the LSP to encrypt. The benefit of ANYsec is the ability to easily enable encryption on an LSP at any time, without changing the LSP's service lease agreement (SLA). ANYsec achieves this by doing the following:

- ANYsec uses the MACsec engine to achieve low latency, line rate encryption, with less than 1us and within 100s of ns of introduced encryption and decryption latency.
- ANYsec identifies an LSP to encrypt using a disjoint configuration, and encrypts it independently, allowing creation of MPLS networks with appropriate design and SLA that can be expanded to include encryption on LSPs in the future.
- ANYsec uses bidirectional configuration, even for LSPs that are unidirectional, and therefore the ANYsec configuration has to identify an outgoing LSP to encrypt and an incoming LSP to decrypt, under the same encryption group.

The following example displays a configuration to enable encryption on the egress LSP with the node SID 1.1.216.136, and decryption for the local node SID bound to 1.1.216.100. Both LSPs are on IGP instance ID 11.

Example: MD-CLI

```
[ex:/configure anysec]
A:admin@node-2# info
    reserved-label-block "rlb-1"
    mka-over-ip {
        mka-udp-port 12345
    }
    tunnel-encryption {
        security-termination-policy "STP-1" {
            admin-state enable
            local-address 1.1.216.100
            igp-instance-id 11
        }
        security-termination-policy "Sec-Term-Policy" {
        }
        security-termination-policy "test-policy" {
        }
        encryption-group "EG-1" {
```

```
            security-termination-policy "STP-1"
            ca-name "CA-1"
            peer-tunnel-attributes {
                igp-instance-id 11
            }
            peer 1.1.216.136 {
                admin-state enable
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config>anysec# info
----------------------------------------------
     reserved-label-block "rlb-1"
     mka-over-ip
         mka-udp-port 12345
     exit
     tunnel-encryption
         security-termination-policy "STP-1" create
             igp-instance-id 11
             local-address 1.1.216.100 //decrypt the incoming LSP for local node sid
             no shutdown
         exit
     encryption-group "EG-1" create
         ca-name "CA-1"
         security-termination-policy "STP-1"
         peer-tunnel-attributes
             igp-instance-id 11
         exit
         peer 1.1.216.136 create //encrypt the outgoing LSP for remote node sid
             no shutdown
         exit
         no shutdown
     exit
 exit
```

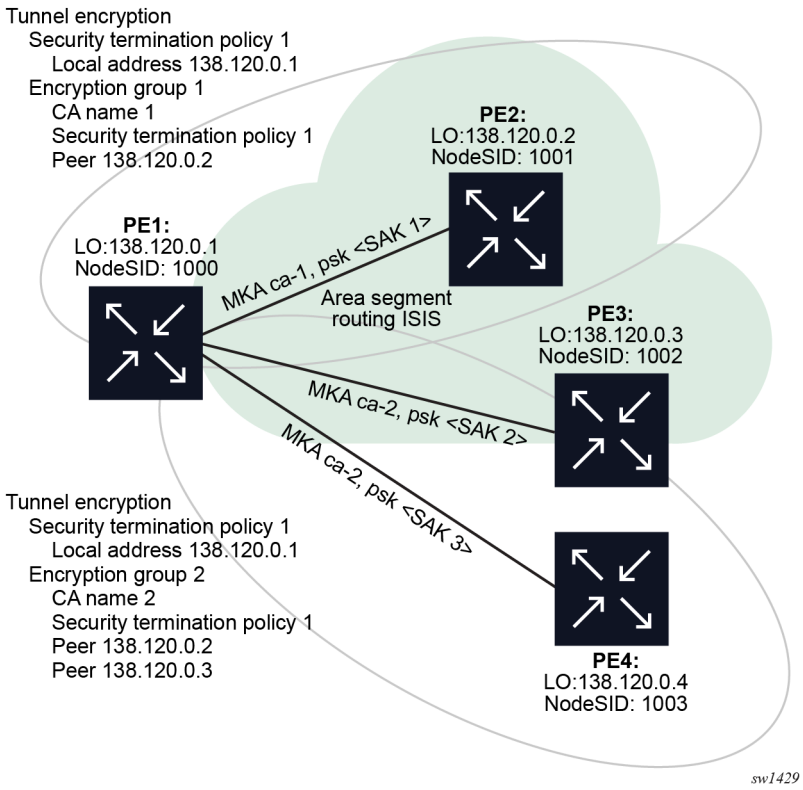
The security-termination policy identifies the decrypting LSP and the node SID on the local router. The decrypting LSPs node SID can arrive from multiple peers, and the SAKs used to decrypt these incoming LSPs are different for each peer.

The encryption group is a group of peers that use the same CA and PSK to secure the SAK over MKA. Each peer node SID and its local node SID use a different SAK to encrypt the datapath, but all these SAKs under the same encryption group are secured and signaled via the same CA and PSK.

A security-termination policy can be part of multiple encryption groups, however, a peer can only be part of a single encryption group. This allows a set of bidirectional LSPs in an encryption group to use the same CA and PSK to secure and signal SAKs between them.

The following figure shows the implementation of ANYsec tunnel encryption using two encryption groups with the same security termination policy, and different CAs and peers.

Figure 79: ANYsec tunnel encryption implementation



7.8.3 Configuring ANYsec MKA

ANYsec uses MKA to distribute SAKs. MKA is part of IEEE802.1x standard and is a Layer 2 protocol without an IP header. As an MPLS Layer 2.5 encryption protocol, ANYsec reuses MKA by encapsulating MKA in IP/UDP to distribute the SAK from one PE to the other.

The user-configurable UDP port identifies the MKA packets on the router. The following configuration guidelines apply:

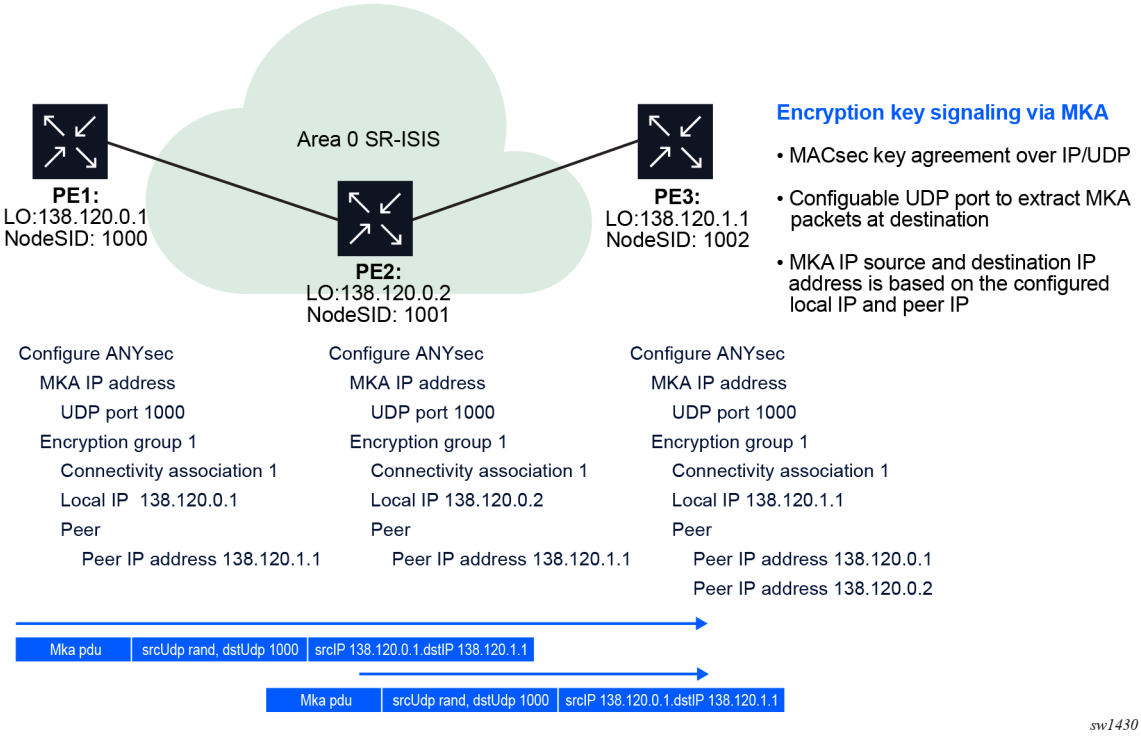
- Reserve the UDP port for MKA for the entire network.
- Ensure that the UDP port is not in use by any other application.

Use the following command to configure the MKA UDP port.

```
configure anysec mka-over-ip mka-udp-port
```

The following figure shows the ANYsec implementation using the MKA UDP port configuration.

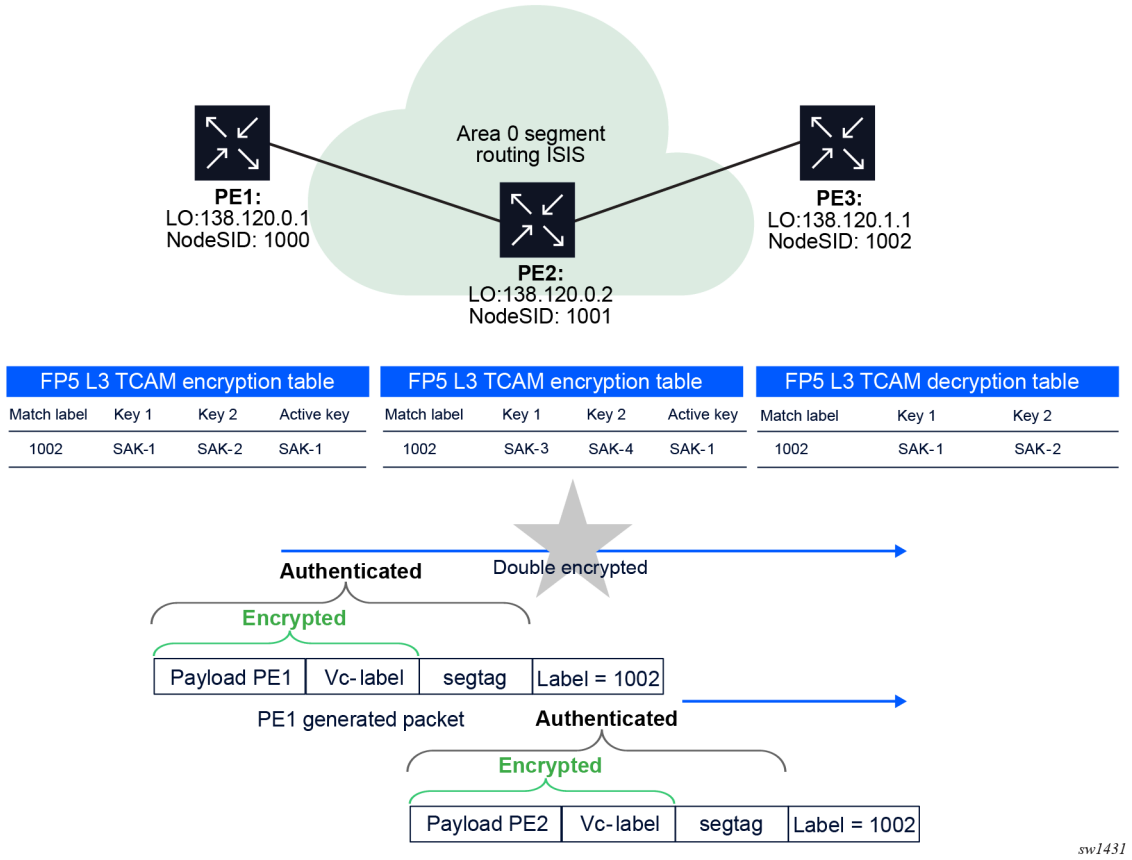
Figure 80: ANYsec using MKA UDP port configuration



7.8.4 Configuring ANYsec encryption SIDs

With ANYsec, the encryption engine locks on a unique label stack and encrypts it with a unique SAK. This label stack needs to be programmed by SR OS in the encryption engine. This creates a problem with double encryption in segment routing technology. In segment routing, all peers wanting to send a packet to a local router, send the packet to the node SID of the local router. As such, the label stack that is pushed on the packet by different routers to switch the packet to the local router may be identical. This means if an encrypting node is also an LSR node for an upstream router, it would encrypt its own local packets to the downstream ANYsec router and would double encrypt the upstream ANYsec router packets that are going through it to the same downstream ANYsec router. For example, in the following figure PE2 would encrypt its own packet destined for PE3 and double encrypt the PE1 packets with same MPLS label stack destined for PE3. This double encryption would mean that PE1 packets are not decrypted correctly at PE3.

Figure 81: ANYsec double-encryption scenario



To solve this problem, SR OS introduced the encryption SID concept. The encryption SID uniquely identifies the encrypting router within the network. The encryption SID is pushed by the encrypting router at the bottom of the label stack with S bit set.

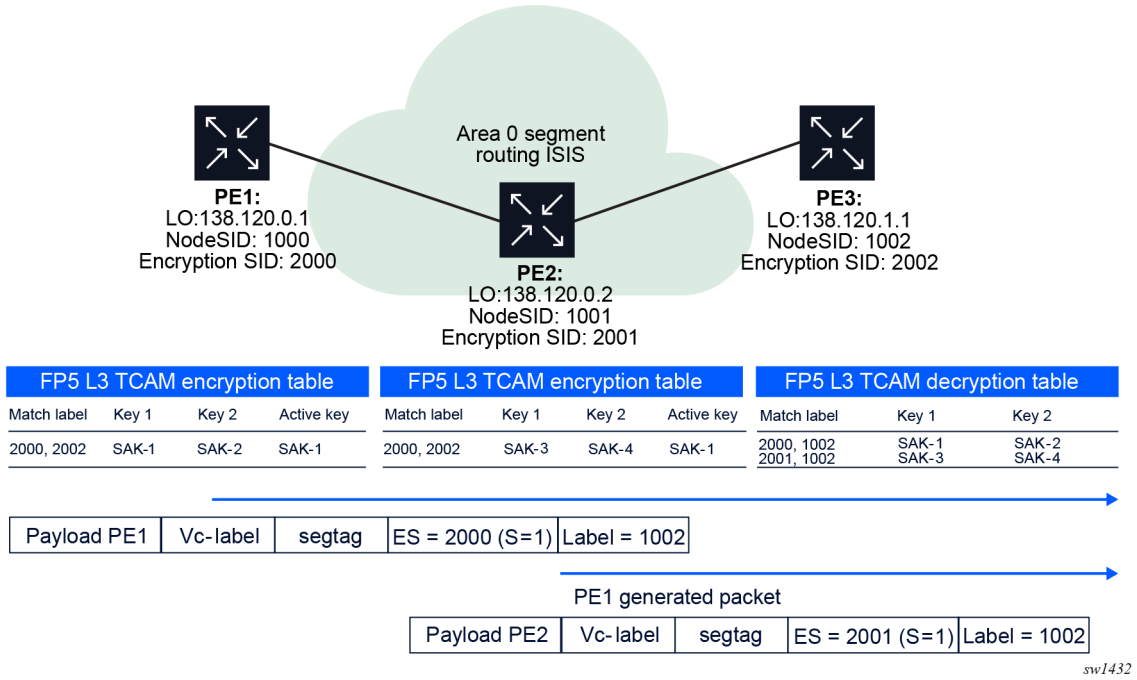
In the following figure, the encryption SID 2000 is assigned to PE1 encrypting node and encryption SID 2001 is assigned to PE2 encrypting node. Both sending packets to PE3.

The encryption SID pushed at the bottom of the label stack ensures that each encrypting node has a unique label stack so double encryption does not occur on the PE2. The PE2 ANYsec CAM is only programmed to encrypt the label stack of <2001(S=1),1002> and does not match the label stack of <2000(S=1),1002> from PE1.

Ensure the following conditions are met for the encryption SID used in the ANYsec configuration:

- The encryption SID for the ANYsec configuration must be uniquely assigned to each encrypting router.
- The encryption SID for the ANYsec configuration must be assigned from a reserved block of labels on SR OS.

Figure 82: ANYsec encryption SID



The following example displays a configuration of an encryption SID on a encrypting node.

Example: MD-CLI

```
[ex:/configure router "Base" mpls-labels]
A:admin@node-2# info
    sr-labels {
        start 60000
        end 120000
    }
    reserved-label-block "r1b-1" {
        start-label 50000
        end-label 59999
    }

[ex:/configure anysec]
A:admin@node-2# info
    reserved-label-block "r1b-1"
    mka-over-ip {
        mka-udp-port 12345
    }
    tunnel-encryption {
        security-termination-policy "STP-1" {
            admin-state enable
            local-address 1.1.216.100
            igp-instance-id 11
        }
        security-termination-policy "Sec-Term-Policy" {
        }
        security-termination-policy "test-policy" {
        }
        encryption-group "EG-1" {
            admin-state enable
```

```
            security-termination-policy "STP-1"
            encryption-label 50012
            ca-name "CA-1"
            peer-tunnel-attributes {
                igp-instance-id 11
            }
            peer 1.1.216.136 {
                admin-state enable
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config>router>mpls-labels# info
----------------------------------------------
            sr-labels start 60000 end 120000
            reserved-label-block "rlb-1" //reserved block for encryption sid
                start-label 50000 end-label 59999
            exit
----------------------------------------------

A:node-2>config>anysec# info
----------------------------------------------
        reserved-label-block "rlb-1" //reserve blocked assigned to ANYsec
        mka-over-ip
            mka-udp-port 12345
        exit
        tunnel-encryption
            security-termination-policy "STP-1" create
                igp-instance-id 11
                local-address 1.1.216.100
                no shutdown
            exit
            encryption-group "EG-1" create
                ca-name "CA-1"
                encryption-label 50012 //encryption sid that uniquely identifies this
 encrypting node and is pushed on the bottom of label stack when the node encrypts the
 LSP.
                security-termination-policy "STP-1"
                peer-tunnel-attributes
                    igp-instance-id 11
                exit
                peer 1.1.216.136 create
                    no shutdown
                exit
                no shutdown
            exit
        exit
```

7.9 ANYsec OAM MPLS support

OAM support with Segment Routing tunnels

ANYsec supports OAM with following Segment Routing tunnels:

- SR-ISIS
- SR-OSPF

- SR-OSPF3

OAM command support

ANYsec supports OAM tools including P2MP LSP ping and services OAM tools if any services are riding over the ANYsec tunnel. ANYsec supports the use of the following inband OAM commands:

- LSP ping
- SDP ping
- VCCV ping
- Service ping
- ping over a non default service
- L2 service ping

Note: Inband OAM packets are transmitted encrypted over an MPLS tunnel and outband OAM packets are transmitted unencrypted. For example, ANYsec does not encrypt an ICMP reply that comes out of band, such as in IP/UDP.

8 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.

8.1 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*

8.2 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)*

8.3 Border Gateway Protocol (BGP)

draft-gredler-idr-bgplu-epe-14, *Egress Peer Engineering using BGP-LU*

draft-hares-idr-update-attrib-low-bits-fix-01, *Update Attribute Flag Low Bits Clarification*

draft-ietf-idr-add-paths-guidelines-08, *Best Practices for Advertisement of Multiple Paths in IBGP*

draft-ietf-idr-best-external-03, *Advertisement of the best external route in BGP*

draft-ietf-idr-bgp-flowspec-oid-03, *Revised Validation Procedure for BGP Flow Specifications*

draft-ietf-idr-bgp-gr-notification-01, *Notification Message support for BGP Graceful Restart*

draft-ietf-idr-bgp-ls-app-specific-attr-16, *Application-Specific Attributes Advertisement with BGP Link-State*

draft-ietf-idr-bgp-ls-flex-algo-06, *Flexible Algorithm Definition Advertisement with BGP Link-State*

draft-ietf-idr-bgp-optimal-route-reflection-10, *BGP Optimal Route Reflection (BGP-ORR)*

draft-ietf-idr-error-handling-03, *Revised Error Handling for BGP UPDATE Messages*

draft-ietf-idr-flowspec-interfaceset-03, *Applying BGP flowspec rules on a specific interface set*

draft-ietf-idr-flowspec-path-redirect-05, *Flowspec Indirection-id Redirect* – localised ID

draft-ietf-idr-flowspec-redirect-ip-02, *BGP Flow-Spec Redirect to IP Action*

draft-ietf-idr-link-bandwidth-03, *BGP Link Bandwidth Extended Community*

RFC 1772, *Application of the Border Gateway Protocol in the Internet*

RFC 1997, *BGP Communities Attribute*

RFC 2385, *Protection of BGP Sessions via the TCP MD5 Signature Option*

RFC 2439, *BGP Route Flap Damping*

RFC 2545, *Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*

RFC 2858, *Multiprotocol Extensions for BGP-4*

RFC 2918, *Route Refresh Capability for BGP-4*

RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*

RFC 4360, *BGP Extended Communities Attribute*

RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 4456, *BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)*

RFC 4486, *Subcodes for BGP Cease Notification Message*

RFC 4659, *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*

RFC 4684, *Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*

RFC 4724, *Graceful Restart Mechanism for BGP* – helper mode

RFC 4760, *Multiprotocol Extensions for BGP-4*

RFC 4798, *Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*

RFC 5004, *Avoid BGP Best Path Transitions from One External to Another*

RFC 5065, *Autonomous System Confederations for BGP*

RFC 5291, *Outbound Route Filtering Capability for BGP-4*

RFC 5396, *Textual Representation of Autonomous System (AS) Numbers* – asplain

RFC 5492, *Capabilities Advertisement with BGP-4*

RFC 5668, *4-Octet AS Specific BGP Extended Community*

RFC 6286, *Autonomous-System-Wide Unique BGP Identifier for BGP-4*

RFC 6368, *Internal BGP as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 6793, *BGP Support for Four-Octet Autonomous System (AS) Number Space*

RFC 6810, *The Resource Public Key Infrastructure (RPKI) to Router Protocol*

RFC 6811, *Prefix Origin Validation*

RFC 6996, *Autonomous System (AS) Reservation for Private Use*

RFC 7311, *The Accumulated IGP Metric Attribute for BGP*

RFC 7606, *Revised Error Handling for BGP UPDATE Messages*

RFC 7607, *Codification of AS 0 Processing*

RFC 7674, *Clarification of the Flowspec Redirect Extended Community*

RFC 7752, *North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP*

RFC 7854, *BGP Monitoring Protocol (BMP)*

RFC 7911, *Advertisement of Multiple Paths in BGP*

RFC 7999, *BLACKHOLE Community*

RFC 8092, *BGP Large Communities Attribute*

RFC 8097, *BGP Prefix Origin Validation State Extended Community*

RFC 8212, *Default External BGP (EBGP) Route Propagation Behavior without Policies*

RFC 8277, *Using BGP to Bind MPLS Labels to Address Prefixes*

RFC 8571, *BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions*

RFC 8950, *Advertising IPv4 Network Layer Reachability Information (NLRI) with an IPv6 Next Hop*

RFC 8955, *Dissemination of Flow Specification Rules*

RFC 8956, *Dissemination of Flow Specification Rules for IPv6*

RFC 9086, *Border Gateway Protocol - Link State (BGP-LS) Extensions for Segment Routing BGP Egress Peer Engineering*

RFC 9494, *Long-Lived Graceful Restart for BGP*

8.4 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*

8.5 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)*

8.6 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*

8.7 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*

8.8 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*

8.9 Ethernet VPN (EVPN)

draft-ietf-bess-bgp-srv6-args-00, *SRv6 Argument Signaling for BGP Services*

draft-ietf-bess-evpn-ip-aliasing-00, *EVPN Support for L3 Fast Convergence and Aliasing/Backup Path* – IP Prefix routes

draft-ietf-bess-evpn-ipvpn-interworking-06, *EVPN Interworking with IPVPN*

draft-ietf-bess-evpn-irb-mcast-09, *EVPN Optimized Inter-Subnet Multicast (OISM) Forwarding* – ingress replication and mLDP

draft-ietf-bess-evpn-pref-df-06, *Preference-based EVPN DF Election*

draft-ietf-bess-evpn-unequal-lb-16, *Weighted Multi-Path Procedures for EVPN Multi-Homing* – section 9

draft-ietf-bess-evpn-virtual-eth-segment-06, *EVPN Virtual Ethernet Segment*

draft-ietf-bess-pbb-evpn-isid-cmacflush-00, *PBB-EVPN ISID-based CMAC-Flush*

draft-sr-bess-evpn-vpws-gateway-03, *Ethernet VPN Virtual Private Wire Services Gateway Solution*

RFC 7432, *BGP MPLS-Based Ethernet VPN*

RFC 7623, *Provider Backbone Bridging Combined with Ethernet VPN (PBB-EVPN)*

RFC 8214, *Virtual Private Wire Service Support in Ethernet VPN*

RFC 8317, *Ethernet-Tree (E-Tree) Support in Ethernet VPN (EVPN) an Provider Backbone Bridging EVPN (PBB-EVPN)*

RFC 8365, *A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)*

RFC 8560, *Seamless Integration of Ethernet VPN (EVPN) with Virtual Private LAN Service (VPLS) and Their Provider Backbone Bridge (PBB) Equivalents*

RFC 8584, *DF Election and AC-influenced DF Election*

RFC 9047, *Propagation of ARP/ND Flags in an Ethernet Virtual Private Network (EVPN)*

RFC 9135, *Integrated Routing and Bridging in Ethernet VPN (EVPN)* – Asymmetric IRB Procedures and Mobility Procedure

RFC 9136, *IP Prefix Advertisement in Ethernet VPN (EVPN)*

RFC 9161, *Operational Aspects of Proxy ARP/ND in Ethernet Virtual Private Networks*

RFC 9251, *Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Proxies for Ethernet VPN (EVPN)*

8.10 gRPC Remote Procedure Calls (gRPC)

Protobuf version 0.1.0, *gNMI Commit Confirmed Extension*

Protobuf version 0.1.0, *gNOI Certificate Management Service*

Protobuf version 0.1.0, *gNOI File Service*

Protobuf version 0.8.0, *gNMI Service Specification*

Protobuf version 1.0.0, *gNOI System Service*

PROTOCOL-HTTP2, *gRPC over HTTP2*

8.11 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*

8.12 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*

8.13 Internet Protocol (IP) general

draft-grant-tacacs-02, *The TACACS+ Protocol*

RFC 768, *User Datagram Protocol*

RFC 793, *Transmission Control Protocol*

RFC 854, *Telnet Protocol Specifications*

RFC 1350, *The TFTP Protocol (revision 2)*

RFC 2347, *TFTP Option Extension*

RFC 2348, *TFTP Blocksize Option*

RFC 2349, *TFTP Timeout Interval and Transfer Size Options*

RFC 2428, *FTP Extensions for IPv6 and NATs*

RFC 2617, *HTTP Authentication: Basic and Digest Access Authentication*

RFC 2784, *Generic Routing Encapsulation (GRE)*

RFC 2818, *HTTP Over TLS*

RFC 2890, *Key and Sequence Number Extensions to GRE*

RFC 3164, *The BSD syslog Protocol*

RFC 4250, *The Secure Shell (SSH) Protocol Assigned Numbers*

RFC 4251, *The Secure Shell (SSH) Protocol Architecture*

RFC 4252, *The Secure Shell (SSH) Authentication Protocol* – publickey, password

RFC 4253, *The Secure Shell (SSH) Transport Layer Protocol*

RFC 4254, *The Secure Shell (SSH) Connection Protocol*

RFC 4511, *Lightweight Directory Access Protocol (LDAP): The Protocol*

RFC 4513, *Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms* – TLS

RFC 4632, *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*

RFC 5082, *The Generalized TTL Security Mechanism (GTSM)*

RFC 5246, *The Transport Layer Security (TLS) Protocol Version 1.2* – TLS client, RSA public key

RFC 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*

8.14 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)*

8.15 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

8.16 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*

8.17 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)*

8.18 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*

8.19 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"*

8.20 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*

8.21 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*

8.22 Network Address Translation (NAT)

draft-ietf-behave-address-format-10, *IPv6 Addressing of IPv4/IPv6 Translators*

draft-ietf-behave-v6v4-xlate-23, *IP/ICMP Translation Algorithm*

draft-miles-behave-l2nat-00, *Layer2-Aware NAT*

draft-nishitani-cgn-02, *Common Functions of Large Scale NAT (LSN)*

RFC 4787, *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*

RFC 5382, *NAT Behavioral Requirements for TCP*

RFC 5508, *NAT Behavioral Requirements for ICMP*

RFC 6146, *Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers*

RFC 6333, *Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion*

RFC 6334, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite*

RFC 6887, *Port Control Protocol (PCP)*

RFC 6888, *Common Requirements For Carrier-Grade NATs (CGNs)*

RFC 7753, *Port Control Protocol (PCP) Extension for Port-Set Allocation*

RFC 7915, *IP/ICMP Translation Algorithm*

8.23 Network Configuration Protocol (NETCONF)

RFC 5277, *NETCONF Event Notifications*

RFC 6020, *YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)*

RFC 6022, *YANG Module for NETCONF Monitoring*

RFC 6241, *Network Configuration Protocol (NETCONF)*

RFC 6242, *Using the NETCONF Protocol over Secure Shell (SSH)*

RFC 6243, *With-defaults Capability for NETCONF*

RFC 8342, *Network Management Datastore Architecture (NMDA)* – Startup, Candidate, Running and Intended datastores

RFC 8525, *YANG Library*

RFC 8526, *NETCONF Extensions to Support the Network Management Datastore Architecture* – <get-data> operation

8.24 Open Shortest Path First (OSPF)

RFC 1765, *OSPF Database Overflow*

RFC 2328, *OSPF Version 2*

RFC 3101, *The OSPF Not-So-Stubby Area (NSSA) Option*

RFC 3509, *Alternative Implementations of OSPF Area Border Routers*

RFC 3623, *Graceful OSPF Restart Graceful OSPF Restart* – helper mode

RFC 3630, *Traffic Engineering (TE) Extensions to OSPF Version 2*

RFC 4222, *Prioritized Treatment of Specific OSPF Version 2 Packets and Congestion Avoidance*

RFC 4552, *Authentication/Confidentiality for OSPFv3*

RFC 4576, *Using a Link State Advertisement (LSA) Options Bit to Prevent Looping in BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 4577, *OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 5185, *OSPF Multi-Area Adjacency*

RFC 5187, *OSPFv3 Graceful Restart* – helper mode

RFC 5243, *OSPF Database Exchange Summary List Optimization*

RFC 5250, *The OSPF Opaque LSA Option*

RFC 5309, *Point-to-Point Operation over LAN in Link State Routing Protocols*

RFC 5340, *OSPF for IPv6*

RFC 5642, *Dynamic Hostname Exchange Mechanism for OSPF*

RFC 5709, *OSPFv2 HMAC-SHA Cryptographic Authentication*

RFC 5838, *Support of Address Families in OSPFv3*

RFC 6549, *OSPFv2 Multi-Instance Extensions*

RFC 6987, *OSPF Stub Router Advertisement*

RFC 7471, *OSPF Traffic Engineering (TE) Metric Extensions* – Min/Max Unidirectional Link Delay metric for flex-algo, RSVP, SR-TE

RFC 7684, *OSPFv2 Prefix/Link Attribute Advertisement*

RFC 7770, *Extensions to OSPF for Advertising Optional Router Capabilities*

RFC 8362, *OSPFv3 Link State Advertisement (LSA) Extensibility*

RFC 8920, *OSPF Application-Specific Link Attributes*

8.25 OpenFlow

TS-007 Version 1.3.1, *OpenFlow Switch Specification* – OpenFlow-hybrid switches

8.26 Path Computation Element Protocol (PCEP)

draft-alvarez-pce-path-profiles-04, *PCE Path Profiles*

draft-dhs-spring-pce-sr-p2mp-policy-00, *PCEP extensions for p2mp sr policy*

draft-ietf-pce-binding-label-sid-15, *Carrying Binding Label/Segment Identifier (SID) in PCE-based Networks.* – MPLS binding SIDs

draft-ietf-pce-pceps-tls13-04, *Updates for PCEPS: TLS Connection Establishment Restrictions*

RFC 5440, *Path Computation Element (PCE) Communication Protocol (PCEP)*

RFC 8231, *Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE*

RFC 8253, *PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)*

RFC 8281, *PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model*

RFC 8408, *Conveying Path Setup Type in PCE Communication Protocol (PCEP) Messages*

RFC 8664, *Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing*

8.27 Point-to-Point Protocol (PPP)

RFC 1332, *The PPP Internet Protocol Control Protocol (IPCP)*

RFC 1990, *The PPP Multilink Protocol (MP)*

RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*

RFC 2516, *A Method for Transmitting PPP Over Ethernet (PPPoE)*

RFC 4638, *Accommodating a Maximum Transit Unit/Maximum Receive Unit (MTU/MRU) Greater Than 1492 in the Point-to-Point Protocol over Ethernet (PPPoE)*

RFC 5072, *IP Version 6 over PPP*

8.28 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*

8.29 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*

8.30 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)*

8.31 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*

8.32 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*

8.33 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*

8.34 Segment Routing (SR)

draft-ietf-bess-mvpn-evpn-sr-p2mp-07, *Multicast and Ethernet VPN with Segment Routing P2MP and Ingress Replication* – MVPN

draft-bashandy-rtgwg-segment-routing-uloop-15, *Loop avoidance using Segment Routing*

draft-filsfils-spring-net-pgm-extension-srv6-usid-15, *Network Programming extension: SRv6 uSID instruction*

draft-filsfils-spring-srv6-net-pgm-insertion-08, *SRv6 NET-PGM extension: Insertion*

draft-ietf-idr-bgpls-srv6-ext-14, *BGP Link State Extensions for SRv6*

draft-ietf-idr-segment-routing-te-policy-23, *Advertising Segment Routing Policies in BGP*

draft-ietf-idr-ts-flowspec-srv6-policy-03, *Traffic Steering using BGP FlowSpec with SR Policy*

draft-ietf-pim-p2mp-policy-ping-03, *P2MP Policy Ping*

draft-ietf-pim-sr-p2mp-policy-06, *Segment Routing Point-to-Multipoint Policy* – MPLS

draft-ietf-rtgwg-segment-routing-ti-lfa-11, *Topology Independent Fast Reroute using Segment Routing*

draft-ietf-spring-conflict-resolution-05, *Segment Routing MPLS Conflict Resolution*

draft-ietf-spring-sr-replication-segment-16, *SR Replication segment for Multi-point Service Delivery* – MPLS

draft-ietf-spring-srv6-srh-compression-xx, *Compressed SRv6 Segment List Encoding in SRH*

draft-voyer-6man-extension-header-insertion-10, *Deployments With Insertion of IPv6 Segment Routing Headers*

RFC 8287, *Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes*

RFC 8426, *Recommendations for RSVP-TE and Segment Routing (SR) Label Switched Path (LSP) Coexistence*

RFC 8476, *Signaling Maximum SID Depth (MSD) Using OSPF* – node MSD

RFC 8491, *Signaling Maximum SID Depth (MSD) Using IS-IS* – node MSD

RFC 8660, *Segment Routing with the MPLS Data Plane*

RFC 8661, *Segment Routing MPLS Interworking with LDP*

RFC 8663, *MPLS Segment Routing over IP* – BGP SR with SR-MPLS-over-UDP/IP

RFC 8665, *OSPF Extensions for Segment Routing*

RFC 8666, *OSPFv3 Extensions for Segment Routing*

RFC 8667, *IS-IS Extensions for Segment Routing*

RFC 8669, *Segment Routing Prefix Segment Identifier Extensions for BGP*

RFC 8754, *IPv6 Segment Routing Header (SRH)*

RFC 8814, *Signaling Maximum SID Depth (MSD) Using the Border Gateway Protocol - Link State*

RFC 8986, *Segment Routing over IPv6 (SRv6) Network Programming*

RFC 9085, *Border Gateway Protocol - Link State (BGP-LS) Extensions for Segment Routing*

RFC 9088, *Signaling Entropy Label Capability and Entropy Readable Label Depth Using IS-IS* – advertising ELC

RFC 9089, *Signaling Entropy Label Capability and Entropy Readable Label Depth Using OSPF* – advertising ELC

RFC 9252, *BGP Overlay Services Based on Segment Routing over IPv6 (SRv6)*

RFC 9256, *Segment Routing Policy Architecture*

RFC 9259, *Operations, Administration, and Maintenance (OAM) in Segment Routing over IPv6 (SRv6)*

RFC 9350, *IGP Flexible Algorithm*

RFC 9352, *IS-IS Extensions to Support Segment Routing over the IPv6 Data Plane*

8.35 Simple Network Management Protocol (SNMP)

draft-blumenthal-aes-usm-04, *The AES Cipher Algorithm in the SNMP's User-based Security Model* – CFB128-AES-192 and CFB128-AES-256

draft-ietf-isis-wg-mib-06, *Management Information Base for Intermediate System to Intermediate System (IS-IS)*

draft-ietf-mboned-msdp-mib-01, *Multicast Source Discovery protocol MIB*

draft-ietf-mpls-ldp-mib-07, *Definitions of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol (LDP)*

draft-ietf-mpls-lsr-mib-06, *Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base Using SMIv2*

draft-ietf-mpls-te-mib-04, *Multiprotocol Label Switching (MPLS) Traffic Engineering Management Information Base*

draft-ietf-ospf-mib-update-08, *OSPF Version 2 Management Information Base*

draft-ietf-vrrp-unified-mib-06, *Definitions of Managed Objects for the VRRP over IPv4 and IPv6* – IPv6

ESO-CONSORTIUM-MIB revision 200406230000Z, *esoConsortiumMIB*

IANA-ADDRESS-FAMILY-NUMBERS-MIB revision 200203140000Z, *ianaAddressFamilyNumbers*

IANAifType-MIB revision 200505270000Z, *ianaifType*

IANA-RTPROTO-MIB revision 200009260000Z, *ianaRtProtoMIB*

IEEE8021-CFM-MIB revision 200706100000Z, *ieee8021CfmMib*

IEEE8021-PAE-MIB revision 200101160000Z, *ieee8021paeMIB*

IEEE8023-LAG-MIB revision 200006270000Z, *lagMIB*

LLDP-MIB revision 200505060000Z, *lldpMIB*

RFC 1157, *A Simple Network Management Protocol (SNMP)*

RFC 1212, *Concise MIB Definitions*

RFC 1215, *A Convention for Defining Traps for use with the SNMP*

RFC 1724, *RIP Version 2 MIB Extension*

RFC 1901, *Introduction to Community-based SNMPv2*

RFC 2021, *Remote Network Monitoring Management Information Base Version 2 using SMIv2*

RFC 2206, *RSVP Management Information Base using SMIv2*

RFC 2213, *Integrated Services Management Information Base using SMIv2*

RFC 2494, *Definitions of Managed Objects for the DS0 and DS0 Bundle Interface Type*

RFC 2578, *Structure of Management Information Version 2 (SMIv2)*

RFC 2579, *Textual Conventions for SMIv2*

RFC 2580, *Conformance Statements for SMIv2*

RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol*

RFC 2819, *Remote Network Monitoring Management Information Base*

RFC 2856, *Textual Conventions for Additional High Capacity Data Types*

RFC 2863, *The Interfaces Group MIB*

RFC 2864, *The Inverted Stack Table Extension to the Interfaces Group MIB*

RFC 2933, *Internet Group Management Protocol MIB*

RFC 3014, *Notification Log MIB*

RFC 3165, *Definitions of Managed Objects for the Delegation of Management Scripts*

RFC 3231, *Definitions of Managed Objects for Scheduling Management Operations*

RFC 3273, *Remote Network Monitoring Management Information Base for High Capacity Networks*

RFC 3410, *Introduction and Applicability Statements for Internet Standard Management Framework*

RFC 3411, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*

RFC 3412, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*

RFC 3413, *Simple Network Management Protocol (SNMP) Applications*

RFC 3414, *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*

RFC 3415, *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*

RFC 3416, *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*

RFC 3417, *Transport Mappings for the Simple Network Management Protocol (SNMP)* – SNMP over UDP over IPv4

RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*

RFC 3419, *Textual Conventions for Transport Addresses*

RFC 3498, *Definitions of Managed Objects for Synchronous Optical Network (SONET) Linear Automatic Protection Switching (APS) Architectures*

RFC 3584, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*

RFC 3592, *Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Interface Type*

RFC 3593, *Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals*

RFC 3635, *Definitions of Managed Objects for the Ethernet-like Interface Types*

RFC 3637, *Definitions of Managed Objects for the Ethernet WAN Interface Sublayer*

RFC 3826, *The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model*

RFC 3877, *Alarm Management Information Base (MIB)*

RFC 3895, *Definitions of Managed Objects for the DS1, E1, DS2, and E2 Interface Types*

RFC 3896, *Definitions of Managed Objects for the DS3/E3 Interface Type*

RFC 4001, *Textual Conventions for Internet Network Addresses*

RFC 4022, *Management Information Base for the Transmission Control Protocol (TCP)*

RFC 4113, *Management Information Base for the User Datagram Protocol (UDP)*

RFC 4220, *Traffic Engineering Link Management Information Base*

RFC 4273, *Definitions of Managed Objects for BGP-4*

RFC 4292, *IP Forwarding Table MIB*

RFC 4293, *Management Information Base for the Internet Protocol (IP)*

RFC 4631, *Link Management Protocol (LMP) Management Information Base (MIB)*

RFC 4878, *Definitions and Managed Objects for Operations, Administration, and Maintenance (OAM) Functions on Ethernet-Like Interfaces*

RFC 7420, *Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module*

RFC 7630, *HMAC-SHA-2 Authentication Protocols in the User-based Security Model (USM) for SNMPv3*

SFLOW-MIB revision 200309240000Z, *sFlowMIB*

8.36 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*

8.37 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

8.38 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)*

8.39 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*

8.40 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*

8.41 Yet Another Next Generation (YANG) OpenConfig Models

openconfig-aaa.yang version 0.4.0, *OpenConfig AAA Model*

openconfig-aaa-radius.yang version 0.3.0, *OpenConfig AAA RADIUS Model*

openconfig-aaa-tacacs.yang version 0.3.0, *OpenConfig AAA TACACS+ Model*

openconfig-acl.yang version 1.0.0, *OpenConfig ACL Model*

openconfig-alarms.yang version0.3.2, *OpenConfig System Alarms Model*

openconfig-bfd.yang version 0.2.2, *OpenConfig BFD Model*

openconfig-bgp.yang version 6.1.0, *OpenConfig BGP Model*

openconfig-bgp-common.yang version 6.0.0, *OpenConfig BGP Common Model*

openconfig-bgp-common-multiprotocol.yang version 6.0.0, *OpenConfig BGP Common Multiprotocol Model*

openconfig-bgp-common-structure.yang version 6.0.0, *OpenConfig BGP Common Structure Model*

openconfig-bgp-global.yang version 6.0.0, *OpenConfig BGP Global Model*

openconfig-bgp-neighbor.yang version 6.1.0, *OpenConfig BGP Neighbor Model*

openconfig-bgp-peer-group.yang version 6.1.0, *OpenConfig BGP Peer Group Model*

openconfig-bgp-policy.yang version 4.0.1, *OpenConfig BGP Policy Model*

openconfig-if-aggregate.yang version 2.4.3, *OpenConfig Interfaces Aggregated Model*

openconfig-if-ethernet.yang version 2.12.2, *OpenConfig Interfaces Ethernet Model*

openconfig-if-ip.yang version 3.1.0, *OpenConfig Interfaces IP Model*

openconfig-if-ip-ext.yang version 2.3.1, *OpenConfig Interfaces IP Extensions Model*

openconfig-igmp.yang version 0.3.1, *OpenConfig IGMP Model*

openconfig-interfaces.yang version 3.0.0, *OpenConfig Interfaces Model*

openconfig-isis.yang version 1.1.0, *OpenConfig IS-IS Model*

openconfig-isis-policy.yang version 0.5.0, *OpenConfig IS-IS Policy Model*

openconfig-isis-routing.yang version 1.1.0, *OpenConfig IS-IS Routing Model*

openconfig-lacp.yang version 1.3.0, *OpenConfig LACP Model*

openconfig-lldp.yang version 0.1.0, *OpenConfig LLDP Model*

openconfig-local-routing.yang version 1.2.0, *OpenConfig Local Routing Model*

openconfig-mpls.yang version 2.3.0, *OpenConfig MPLS Model*

openconfig-mpls-ldp.yang version 3.0.2, *OpenConfig MPLS LDP Model*

openconfig-mpls-rsvp.yang version 2.3.0, *OpenConfig MPLS RSVP Model*

openconfig-mpls-te.yang version 2.3.0, *OpenConfig MPLS TE Model*

openconfig-network-instance.yang version 1.1.0, *OpenConfig Network Instance Model*

openconfig-network-instance-l3.yang version 0.11.1, *OpenConfig L3 Network Instance Model* – static routes

openconfig-ospfv2.yang version 0.4.0, *OpenConfig OSPFv2 Model*

openconfig-ospfv2-area.yang version 0.4.0, *OpenConfig OSPFv2 Area Model*

openconfig-ospfv2-area-interface.yang version 0.4.0, *OpenConfig OSPFv2 Area Interface Model*

openconfig-ospfv2-common.yang version 0.4.0, *OpenConfig OSPFv2 Common Model*

openconfig-ospfv2-global.yang version 0.4.0, *OpenConfig OSPFv2 Global Model*

openconfig-packet-match.yang version 1.0.0, *OpenConfig Packet Match Model*

openconfig-pim.yang version 0.4.3, *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-relay-agent.yang version 0.1.0, *OpenConfig Relay Agent Model*

openconfig-routing-policy.yang version 3.0.0, *OpenConfig Routing Policy Model*

openconfig-rsvp-sr-ext.yang version 0.1.0, *OpenConfig RSVP-TE and SR Extensions Model*

openconfig-system.yang version 0.10.1, *OpenConfig System Model*

openconfig-system-grpc.yang version 1.0.0, *OpenConfig System gRPC Model*

openconfig-system-logging.yang version 0.3.1, *OpenConfig System Logging Model*

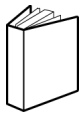
openconfig-system-terminal.yang version 0.3.0, *OpenConfig System Terminal Model*

openconfig-telemetry.yang version 0.5.0, *OpenConfig Telemetry Model*

openconfig-terminal-device.yang version 1.9.0, *OpenConfig Terminal Optics Device Model*

openconfig-vlan.yang version 2.0.0, *OpenConfig VLAN Model*

Customer document and product support



Customer documentation

Customer documentation welcome page

Technical support

Product support portal

Documentation feedback

Customer documentation feedback