CONTENTS

PREFACE

Preface vii
Changes to This Document vii
Communications, Services, and Additional Information vii

CHAPTER 1

New and Changed Information for Segment Routing Features 1
New and Changed Information 1

CHAPTER 2

About Segment Routing 3
Scope 3
Need 4
Benefits 4
Workflow for Deploying Segment Routing 5

CHAPTER 3

Configure Segment Routing Global Block 7
About the Segment Routing Global Block 7
Setup a Non-Default Segment Routing Global Block Range 8

CHAPTER 4

Configure Segment Routing for IS-IS Protocol 11
Enabling Segment Routing for IS-IS Protocol 11
Configuring a Prefix-SID on the IS-IS Enabled Loopback Interface 13

CHAPTER 5

Configure Segment Routing for OSPF Protocol 17
Enabling Segment Routing for OSPF Protocol 17
Configuring a Prefix-SID on the OSPF-Enabled Loopback Interface 19

CHAPTER 6

Configure Segment Routing for BGP 23
Segment Routing Traffic Matrix  55
Traffic Collector Process    55
Configuring Traffic Collector  56
Displaying Traffic Information  57

CHAPTER 11

Using Segment Routing OAM  59
MPLS LSP Ping and Traceroute Nil FEC Target  59
Examples: LSP Ping and Traceroute for Nil_FEC Target  60
Preface

The Cisco ASR 9000 Series Aggregation Services Segment Router Routing Configuration Guide preface contains these sections:

- Changes to This Document, on page vii
- Communications, Services, and Additional Information, on page vii

Changes to This Document

Note

This document contains features for IOS XR Release 6.1.x and earlier.

This table lists the changes made to this document since it was first printed.

<table>
<thead>
<tr>
<th>Date</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>November 2016</td>
<td>Republished with documentation updates for Cisco IOS XR Release 6.1.2 features.</td>
</tr>
<tr>
<td>April 2016</td>
<td>Republished with documentation updates for Cisco IOS XR Release 6.0.1 features.</td>
</tr>
<tr>
<td>January 2015</td>
<td>Initial release of this document.</td>
</tr>
</tbody>
</table>

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you’re looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
• To find warranty information for a specific product or product family, access Cisco Warranty Finder.

**Cisco Bug Search Tool**

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.
New and Changed Information for Segment Routing Features

This table summarizes the new and changed feature information for the Segment Routing Configuration Guide for Cisco ASR 9000 Aggregation Services Routers, and lists where they are documented.

- New and Changed Information, on page 1

New and Changed Information

This document contains features for IOS XR Release 6.1.x and earlier.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Introduced/Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seamless BFD</td>
<td>This feature was introduced.</td>
<td>Release 6.1.2</td>
<td>Configuring SR-TE Policies chapter</td>
</tr>
<tr>
<td>Autoroute Destination</td>
<td>This feature was introduced.</td>
<td>Release 6.1.2</td>
<td>Configure SR-TE Policies chapter</td>
</tr>
<tr>
<td>TI-LFA Node and SRLG Protection</td>
<td>This feature was introduced.</td>
<td>Release 6.1.2</td>
<td>Configure Topology-Independent Loop-Free Alternate (TI-LFA) chapter</td>
</tr>
<tr>
<td>BGP Egress Peer Engineering</td>
<td>This feature was introduced.</td>
<td>Release 6.1.2</td>
<td>Configure Segment Routing for BGP chapter</td>
</tr>
<tr>
<td>Distribution of segment routing information through BGP Link State (BGP-LS)</td>
<td>This feature was introduced.</td>
<td>Release 6.1.2</td>
<td>Configure Segment Routing for BGP chapter</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Introduced/Changed in Release</td>
<td>Where Documented</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------------</td>
<td>------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Segment Routing Traffic</td>
<td>This feature was introduced.</td>
<td>Release 6.0.1</td>
<td>Collecting Traffic</td>
</tr>
<tr>
<td>Collector</td>
<td></td>
<td></td>
<td>Statistics chapter</td>
</tr>
</tbody>
</table>
CHAPTER 2

About Segment Routing

Note
Segment Routing is not supported on 1st generation Cisco ASR 9000 Ethernet Line Cards or the Cisco ASR 9000 SIP-700 SPA Interface Processor. Refer to the Cisco ASR 9000 Ethernet Line Card Installation Guide for details about 1st generation line cards.

This chapter introduces the concept of segment routing and provides a workflow for configuring segment routing.

• Scope, on page 3
• Need, on page 4
• Benefits, on page 4
• Workflow for Deploying Segment Routing, on page 5

Scope

Segment routing is a method of forwarding packets on the network based on the source routing paradigm. The source chooses a path and encodes it in the packet header as an ordered list of segments. Segments are an identifier for any type of instruction. For example, topology segments identify the next hop toward a destination. Each segment is identified by the segment ID (SID) consisting of a flat unsigned 20-bit integer.

Segments

Interior gateway protocol (IGP) distributes two types of segments: prefix segments and adjacency segments. Each router (node) and each link (adjacency) has an associated segment identifier (SID).

• A prefix SID is associated with an IP prefix. The prefix SID is manually configured from the segment routing global block (SRGB) range of labels, and is distributed by IS-IS or OSPF. The prefix segment steers the traffic along the shortest path to its destination. A node SID is a special type of prefix SID that identifies a specific node. It is configured under the loopback interface with the loopback address of the node as the prefix.

A prefix segment is a global segment, so a prefix SID is globally unique within the segment routing domain.

• An adjacency segment is identified by a label called an adjacency SID, which represents a specific adjacency, such as egress interface, to a neighboring router. The adjacency SID is distributed by IS-IS or OSPF. The adjacency segment steers the traffic to a specific adjacency.
An adjacency segment is a local segment, so the adjacency SID is locally unique relative to a specific router.

By combining prefix (node) and adjacency segment IDs in an ordered list, any path within a network can be constructed. At each hop, the top segment is used to identify the next hop. Segments are stacked in order at the top of the packet header. When the top segment contains the identity of another node, the receiving node uses equal cost multipaths (ECMP) to move the packet to the next hop. When the identity is that of the receiving node, the node pops the top segment and performs the task required by the next segment.

Dataplane

Segment routing can be directly applied to the Multiprotocol Label Switching (MPLS) architecture with no change in the forwarding plane. A segment is encoded as an MPLS label. An ordered list of segments is encoded as a stack of labels. The segment to process is on the top of the stack. The related label is popped from the stack, after the completion of a segment.

Services

Segment Routing integrates with the rich multi-service capabilities of MPLS, including Layer 3 VPN (L3VPN), Virtual Private Wire Service (VPWS), Virtual Private LAN Service (VPLS), and Ethernet VPN (EVPN).

Segment Routing for Traffic Engineering

Segment routing for traffic engineering (SR-TE) takes place through a tunnel between a source and destination pair. Segment routing for traffic engineering uses the concept of source routing, where the source calculates the path and encodes it in the packet header as a segment. Each segment is an end-to-end path from the source to the destination, and instructs the routers in the provider core network to follow the specified path instead of the shortest path calculated by the IGP. The destination is unaware of the presence of the tunnel.

Need

With segment routing for traffic engineering (SR-TE), the network no longer needs to maintain a per-application and per-flow state. Instead, it simply obeys the forwarding instructions provided in the packet.

SR-TE utilizes network bandwidth more effectively than traditional MPLS-TE networks by using ECMP at every segment level. It uses a single intelligent source and relieves remaining routers from the task of calculating the required path through the network.

Benefits

• **Ready for SDN**: Segment routing was built for SDN and is the foundation for Application Engineered Routing (AER). SR prepares networks for business models, where applications can direct network behavior. SR provides the right balance between distributed intelligence and centralized optimization and programming.

• **Minimal configuration**: Segment routing for TE requires minimal configuration on the source router.

• **Load balancing**: Unlike in RSVP-TE, load balancing for segment routing can take place in the presence of equal cost multiple paths (ECMPs).
• **Supports Fast Reroute (FRR):** Fast reroute enables the activation of a pre-configured backup path within 50 milliseconds of path failure.

• **Plug-and-Play deployment:** Segment routing tunnels are interoperable with existing MPLS control and data planes and can be implemented in an existing deployment.

---

**Workflow for Deploying Segment Routing**

Follow this workflow to deploy segment routing.

1. Configure the Segment Routing Global Block (SRGB)
2. Enable Segment Routing and Node SID on the IGP
3. Configure Segment Routing on the BGP
4. Configure the SR-TE Policy
5. Configure TI-LFA
6. Configure the Segment Routing Mapping Server
7. Collect Traffic Statistics
CHAPTER 3

Configure Segment Routing Global Block

Local label allocation is managed by the label switching database (LSD). The Segment Routing Global Block (SRGB) is the range of label values preserved for segment routing in the LSD.

- About the Segment Routing Global Block, on page 7
- Setup a Non-Default Segment Routing Global Block Range, on page 8

About the Segment Routing Global Block

The SRGB label values are assigned as prefix segment identifiers (SIDs) to SR-enabled nodes and have global significance throughout the domain.

Note

Because the values assigned from the range have domain-wide significance, we recommend that all routers within the domain be configured with the same range of values.

The default SRGB range is from 16000 to 23999.

Note

On SR-capable routers, the default starting value of the dynamic label range is increased from 16000 to 24000, so that the default SRGB label values (16000 to 23999) are available when SR is enabled on a running system. If a dynamic label range has been configured with a starting value of 16000, then the default SRGB label values may already be in use when SR is enabled on a running system. Therefore, you must reload the router after enabling SR to release the currently allocated labels and allocate the SRGB.

Also, if you need to increase the SRGB range after you have enabled SR, you must reload the router to release the currently allocated labels and allocate the new SRGB.

To keep the segment routing configuration simple and to make it easier to troubleshoot segment routing issues, we recommend that you use the default SRGB range on each node in the domain. However, there are instances when you might need to define a different range. For example:

- The nodes of another vendor support a label range that is different from the default SRGB, and you want to use the same SRGB on all nodes.

- The default range is too small.

- To specify separate SRGBs for IS-IS and OSPF protocols, as long as the ranges do not overlap.
Restrictions:
- In Cisco IOS XR release 6.2.x and earlier, LSD label values 0-15999 are reserved.
- In Cisco IOS XR release 6.2.x and earlier, the maximum SRGB size is 65536.
- The SRGB upper bound cannot exceed the platform's capability.

Note: Label values that are not previously reserved are available for dynamic assignment.

The SRGB can be disabled if SR is not used.

Setup a Non-Default Segment Routing Global Block Range

This task explains how to configure a non-default SRGB range.

SUMMARY STEPS

1. configure
2. [router {isis instance-id | ospf process_name} ]
3. segment-routing global-block starting_value ending_value
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>[router {isis instance-id</td>
<td>ospf process_name} ]</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# router isis 1</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>segment-routing global-block starting_value ending_value</td>
<td>Enter the lowest value that you want the SRGB range to include as the starting value. Enter the highest value that you want the SRGB range to include as the ending value.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-isis)# segment-routing global-block 18000 19999</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

Verify the SRGB configuration:

RP/0/RSP0/CPU0:router# show mpls label table detail
Table Label   Owner State Rewrite
------ ------ ------------------------------------------------------- ------ ------
Configure prefix SIDs and enable segment routing.
CHAPTER 4

Configure Segment Routing for IS-IS Protocol


This module provides the configuration information used to enable segment routing for IS-IS.

Note

For additional information on implementing IS-IS on your Cisco ASR 9000 Series Router, see the Implementing IS-IS module in the Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide.

- Enabling Segment Routing for IS-IS Protocol, on page 11
- Configuring a Prefix-SID on the IS-IS Enabled Loopback Interface, on page 13

Enabling Segment Routing for IS-IS Protocol

Segment routing on the IS-IS control plane supports the following:

- IPv4 and IPv6 control plane
- Level 1, level 2, and multi-level routing
- Prefix SIDs for host prefixes on loopback interfaces
- Adjacency SIDs for adjacencies
- MPLS penultimate hop popping (PHP) and explicit-null signaling

This task explains how to enable segment routing for IS-IS.

Before you begin

Your network must support the MPLS Cisco IOS XR software feature before you enable segment routing for IS-IS on your router.
You must enter the commands in the following task list on every IS-IS router in the traffic-engineered portion of your network.

### SUMMARY STEPS

1. `configure`
2. `router isis instance-id`
3. `address-family { ipv4 | ipv6 } [ unicast ]`
4. `metric-style wide [ level { 1 | 2 } ]`
5. `mpls traffic-eng level`
6. `mpls traffic-eng router-id interface`
7. `router-id loopback loopback interface used for prefix-sid`
8. `segment-routing mpls`
9. `exit`
10. `mpls traffic-eng`
11. `commit`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enabling IS-IS routing for the specified routing instance, and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>router isis instance-id</code></td>
<td>Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td><strong>Note</strong> You can change the level of routing to be performed by a particular routing instance by using the is-type router configuration command.</td>
</tr>
<tr>
<td>Step 3</td>
<td>`address-family { ipv4</td>
<td>ipv6 } [ unicast ]`</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>Step 4</td>
<td>`metric-style wide [ level { 1</td>
<td>2 } ]`</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>Step 5</td>
<td><code>mpls traffic-eng level</code></td>
<td>Enables RSVP traffic engineering functionality.</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>6</td>
<td>mpls traffic-eng router-id interface</td>
<td>Sets the traffic engineering loopback interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-isis-af)# mpls traffic-eng router-id Loopback0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>router-id loopback loopback interface used for prefix-sid</td>
<td>Configures router ID for each address-family (ipv4/ipv6).</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/(config-isis-af)#router-id loopback0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>segment-routing mpls</td>
<td>Segment routing is enabled by the following actions:</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-isis-af)# segment-routing mpls</td>
<td>• MPLS forwarding is enabled on all interfaces where IS-IS is active.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• All known prefix-SIDs in the forwarding plain are programmed, with the prefix-SIDs advertised by remote routers or learned through local or remote mapping server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The prefix-SIDs locally configured are advertised.</td>
</tr>
<tr>
<td>9</td>
<td>exit</td>
<td>Enables traffic engineering functionality on the node. The node advertises the traffic engineering link attributes in IGP which populates the traffic engineering database (TED) on the head-end. The RSVP-TE head-end requires the TED to calculate and validate the path of the RSVP-TE policy.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-isis-af)# exit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-isis)# exit</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>mpls traffic-eng</td>
<td>Enables traffic engineering functionality on the node. The node advertises the traffic engineering link attributes in IGP which populates the traffic engineering database (TED) on the head-end. The RSVP-TE head-end requires the TED to calculate and validate the path of the RSVP-TE policy.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

**What to do next**

Configure the prefix SID.

**Configuring a Prefix-SID on the IS-IS Enabled Loopback Interface**

A prefix SID is associated with an IP prefix. The prefix SID is manually configured from the segment routing global block (SRGB) range of labels. The prefix segment steers the traffic along the shortest path to its
destination. A node SID is a special type of prefix SID that identifies a specific node. It is configured under the loopback interface with the loopback address of the node as the prefix.

The prefix SID is globally unique within the segment routing domain.

This task explains how to configure prefix segment identifier (SID) index or absolute value on the IS-IS enabled Loopback interface.

Before you begin

Ensure that segment routing is enabled on the corresponding address family.

**SUMMARY STEPS**

1. configure
2. router isis instance-id
3. interface Loopback instance
4. address-family { ipv4 | ipv6 } [ unicast ]
5. prefix-sid { index SID-index | absolute SID-value } [ n-flag-clear ] [ explicit-null ]
6. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>router isis instance-id</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# router isis 1</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>interface Loopback instance</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-isis)# interface Loopback0</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>address-family { ipv4</td>
</tr>
<tr>
<td>Example:</td>
<td>The following is an example for ipv4 address family:</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-isis-if)# address-family ipv4 unicast</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>prefix-sid { index SID-index</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-isis-if-af)#</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>prefix-sid index 1001</td>
<td>Specify absolute SID-value for each node to create a specific prefix SID within the SRGB. By default, the n-flag is set on the prefix-SID, indicating that it is a node SID. For specific prefix-SID (for example, Anycast prefix-SID), enter the n-flag-clear keyword. IS-IS does not set the n flag in the prefix-SID sub Type Length Value (TLV). To disable penultimate-hop-popping (PHP) and add explicit-Null label, enter explicit-null keyword. IS-IS sets the e flag in the prefix-SID sub TLV.</td>
</tr>
<tr>
<td>prefix-sid absolute 17001</td>
<td></td>
</tr>
</tbody>
</table>

**Step 6**
commit

Verify the prefix-SID configuration:

```bash
RP/0/RSP0/CPU0:router# show isis database verbose
IS-IS 1 (Level-2) Link State Database
LSPID LSP Seq Num LSP Checksum LSP Holdtime ATT/P/OL
router.00-00 * 0x0000039b 0xfc27 1079 0/0/0
Area Address: 49.0001
NLPID: 0xcc
NLPID: 0x8e
MT: Standard (IPv4 Unicast)
MT: IPv6 Unicast 0/0/0
Hostname: router
IP Address: 10.0.0.1
IPv6 Address: 2001:0db8:1234::0a00:0001
Router Cap: 10.0.0.1, D:0, S:0
Segment Routing: I:1 V:1, SRGB Base: 16000 Range: 8000
SR Algorithm:
  Algorithm: 0
...
Metric: 0 IP-Extended 10.0.0.1/32
Prefix-SID Index: 1001, Algorithm:0, R:0 N:1 P:0 E:0 V:0 L:0
...
```

**What to do next**
Configure the SR-TE policy.
CHAPTER 5

Configure Segment Routing for OSPF Protocol

Open Shortest Path First (OSPF) is an Interior Gateway Protocol (IGP) developed by the OSPF working group of the Internet Engineering Task Force (IETF). Designed expressly for IP networks, OSPF supports IP subnetting and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending and receiving packets.

This module provides the configuration information to enable segment routing for OSPF.

Note

For additional information on implementing OSPF on your Cisco ASR 9000 Series Router, see the Implementing OSPF module in the Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide.

- Enabling Segment Routing for OSPF Protocol, on page 17
- Configuring a Prefix-SID on the OSPF-Enabled Loopback Interface, on page 19

Enabling Segment Routing for OSPF Protocol

Segment routing on the OSPF control plane supports the following:

- OSPFv2 control plane
- Multi-area
- IPv4 prefix SIDs for host prefixes on loopback interfaces
- Adjacency SIDs for adjacencies
- MPLS penultimate hop popping (PHP) and explicit-null signaling

This section describes how to enable segment routing MPLS and MPLS forwarding in OSPF. Segment routing can be configured at the instance, area, or interface level.

Before you begin

Your network must support the MPLS Cisco IOS XR software feature before you enable segment routing for OSPF on your router.
You must enter the commands in the following task list on every OSPF router in the traffic-engineered portion of your network.

### SUMMARY STEPS

1. `configure`
2. `router ospf process-name`
3. `segment-routing mpls`
4. `area 0`
5. `mpls traffic-eng area`
6. `mpls traffic-eng router-id interface`
7. `segment-routing mpls`
8. `exit`
9. `mpls traffic-eng`
10. `commit`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>router ospf process-name</code>&lt;br&gt;Example: &lt;br&gt;<code>RP/0/RSP0/CPU0:router(config)# router ospf 1</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>segment-routing mpls</code>&lt;br&gt;Example: &lt;br&gt;<code>RP/0/RSP0/CPU0:router(config-ospf)# segment-routing mpls</code></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>area 0</code>&lt;br&gt;Example: &lt;br&gt;<code>RP/0/RSP0/CPU0:router(config-ospf)# area 0</code></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><code>mpls traffic-eng area</code>&lt;br&gt;Example: &lt;br&gt;<code>RP/0/RSP0/CPU0:router(config-ospf-ar)# mpls traffic-eng area 0</code></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><code>mpls traffic-eng router-id interface</code>&lt;br&gt;Example:</td>
</tr>
</tbody>
</table>
Configure a Prefix-SID on the OSPF-Enabled Loopback Interface

A prefix SID is associated with an IP prefix. The prefix SID is manually configured from the segment routing global block (SRGB) range of labels. The prefix segment steers the traffic along the shortest path to its destination. A node SID is a special type of prefix SID that identifies a specific node. It is configured under the loopback interface with the loopback address of the node as the prefix.

The prefix SID is globally unique within the segment routing domain.

This task describes how to configure prefix segment identifier (SID) index or absolute value on the OSPF-enabled Loopback interface.

Before you begin

Ensure that segment routing is enabled on an instance, area, or interface.

SUMMARY STEPS

1. configure
2. router ospf process-name
3. area value

Configure Segment Routing for OSPF Protocol

Command or Action | Purpose
--- | ---
RP/0/RSP0/CPU0:router(config-ospf-ar)# mpls traffic-eng router-id Loopback0 | (Optional) Enables segment routing using the MPLS data plane on the area and all interfaces in the area. Enables segment routing forwarding on all interfaces in the area and installs the SIDs received by OSPF in the forwarding table.

Step 7

segment-routing mpls

Example:
RP/0/RSP0/CPU0:router(config-ospf-ar)# segment-routing mpls

Step 8

exit

Example:
RP/0/RSP0/CPU0:router(config-ospf-ar)# exit

Step 9

mpls traffic-eng

Example:
RP/0/RSP0/CPU0:router(config)# mpls traffic-eng

Step 10

commit

What to do next

Configure the prefix SID.
configure

Step 2  router ospf  process-name
Example:
RP/0/RSP0/CPU0:router(config)# router ospf 1

Step 3  area  value
Example:
RP/0/RSP0/CPU0:router(config-ospf)# area 0

Step 4  interface Loopback  interface-instance
Example:
RP/0/RSP0/CPU0:router(config-ospf-ar)# interface Loopback0 passive

Step 5  prefix-sid  {index  SID-index  |  absolute  SID-value }  [n-flag-clear]  [explicit-null]
Example:
RP/0/RSP0/CPU0:router(config-ospf-ar)# prefix-sid index 1001
RP/0/RSP0/CPU0:router(config-ospf-ar)# prefix-sid absolute 17001

Step 6  commit

Verify the prefix-SID configuration:

RP/0/RSP0/CPU0:router# show ospf database opaque-area 7.0.0.1 self-originate
OSPF Router with ID (10.0.0.1) (Process ID 1)
    Type-10 Opaque Link Area Link States (Area 0)
<...>
    Extended Prefix TLV: Length: 20
Route-type: 1
AF : 0
Flags : 0x40
Prefix : 10.0.0.1/32

SID sub-TLV: Length: 8
  Flags : 0x0
  MTID : 0
  Algo : 0
  SID Index : 1001

What to do next

Configure SR-TE Policies
CHAPTER 6

Configure Segment Routing for BGP

Border Gateway Protocol (BGP) is an Exterior Gateway Protocol (EGP) that allows you to create loop-free inter-domain routing between autonomous systems. An autonomous system is a set of routers under a single technical administration. Routers in an autonomous system can use multiple Interior Gateway Protocols (IGPs) to exchange routing information inside the autonomous system and an EGP to route packets outside the autonomous system.

This module provides the configuration information used to enable segment routing for BGP.

For additional information on implementing BGP on your Cisco ASR 9000 Series Router, see the Implementing BGP module in the Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide.

- Segment Routing for BGP, on page 23
- Configure BGP Prefix Segment Identifiers, on page 24
- Configure Segment Routing Egress Peer Engineering, on page 25
- Configure BGP Link-State, on page 26
- Example: Configuring SR-EPE and BGP-LS, on page 27

Segment Routing for BGP

In a traditional BGP-based data center (DC) fabric, packets are forwarded hop-by-hop to each node in the autonomous system. Traffic is directed only along the external BGP (eBGP) multipath ECMP. No traffic engineering is possible.

In an MPLS-based DC fabric, the eBGP sessions between the nodes exchange BGP labeled unicast (BGP-LU) network layer reachability information (NLRI). An MPLS-based DC fabric allows any leaf (top-of-rack or border router) in the fabric to communicate with any other leaf using a single label, which results in higher packet forwarding performance and lower encapsulation overhead than traditional BGP-based DC fabric. However, since each label value might be different for each hop, an MPLS-based DC fabric is more difficult to troubleshoot and more complex to configure.

BGP has been extended to carry segment routing prefix-SID index. BGP-LU helps each node learn BGP prefix SIDs of other leaf nodes and can use ECMP between source and destination. Segment routing for BGP simplifies the configuration, operation, and troubleshooting of the fabric. With segment routing for BGP, you can enable traffic steering capabilities in the data center using a BGP prefix SID.
Configure BGP Prefix Segment Identifiers

Segments associated with a BGP prefix are known as BGP prefix SIDs. The BGP prefix SID is global within a segment routing or BGP domain. It identifies an instruction to forward the packet over the ECMP-aware best-path computed by BGP to the related prefix. The BGP prefix SID is manually configured from the segment routing global block (SRGB) range of labels.

Each BGP speaker must be configured with an SRGB using the `segment-routing global-block` command. See the About the Segment Routing Global Block section for information about the SRGB.

**Note**
Because the values assigned from the range have domain-wide significance, we recommend that all routers within the domain be configured with the same range of values.

To assign a BGP prefix SID, first create a routing policy using the `set label-index index` attribute, then associate the index to the node.

**Note**
A routing policy with the `set label-index` attribute can be attached to a network configuration or redistribute configuration. Other routing policy language (RPL) configurations are possible. For more information on routing policies, refer to the "Implementing Routing Policy" chapter in the Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide.

**Example**
The following example shows how to configure the SRGB, create a BGP route policy using a $SID parameter and `set label-index` attribute, and then associate the prefix-SID index to the node.

```
RP/0/RSP0/CPU0:router(config)# segment-routing global-block 16000 23999
RP/0/RSP0/CPU0:router(config)# route-policy SID($SID)
RP/0/RSP0/CPU0:router(config-rpl)# set label-index $SID
RP/0/RSP0/CPU0:router(config-rpl)# end policy
RP/0/RSP0/CPU0:router(config)# router bgp 1
RP/0/RSP0/CPU0:router(config-bgp)# bgp router-id 1.1.1.1
RP/0/RSP0/CPU0:router(config-bgp)# address-family ipv4 unicast
RP/0/RSP0/CPU0:router(config-bgp-af)# network 1.1.3/32 route-policy SID(3)
RP/0/RSP0/CPU0:router(config-bgp-af)# allocate-label all
RP/0/RSP0/CPU0:router(config-bgp-af)# commit
RP/0/RSP0/CPU0:router(config)# show bgp 1.1.1.3/32
BGP routing table entry for 1.1.1.3/32
Versions:  
  Process bRIB/RIB  SendTblVer
  Speaker  74  74
  Local Label=16003
Last Modified: Sep 29 19:52:18.155 for 00:07:22
Paths: (1 available, best #1)
  Advertised to update-groups (with more than one peer):  
    0.2
```
Configure Segment Routing Egress Peer Engineering

Segment routing egress peer engineering (EPE) uses a controller to instruct an ingress provider edge, or a content source (node) within the segment routing domain, to use a specific egress provider edge (node) and a specific external interface to reach a destination. BGP peer SIDs are used to express source-routed inter-domain paths.

The controller learns the BGP peer SIDs and the external topology of the egress border router through BGP-LS EPE routes. The controller can program an ingress node to steer traffic to a destination through the egress node and peer node using BGP labeled unicast (BGP-LU).

EPE functionality is only required at the EPE egress border router and the EPE controller.

This task explains how to configure segment routing EPE on the EPE egress node.

SUMMARY STEPS

1. **router bgp** *as-number*
2. **neighbor** *ip-address*
3. **remote-as** *as-number*
4. **egress-engineering**

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Specifies the BGP AS number and enters the BGP configuration mode, allowing you to configure the BGP routing process.</td>
</tr>
</tbody>
</table>
| **router bgp** *as-number*  
*Example:*  
RP/0/RSP0/CPU0:router(config)# router bgp 1 | |
| **Step 2** | Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer. |
| **neighbor** *ip-address*  
*Example:*  
RP/0/RSP0/CPU0:router(config-bgp)# neighbor 192.168.1.3 | |
| **Step 3** | Creates a neighbor and assigns a remote autonomous system number to it. |
| **remote-as** *as-number*  
*Example:* | |
Configure BGP Link-State

BGP Link-State (LS) is an Address Family Identifier (AFI) and Sub-address Family Identifier (SAFI) defined to carry interior gateway protocol (IGP) link-state database through BGP. BGP LS delivers network topology information to topology servers and Application Layer Traffic Optimization (ALTO) servers. BGP LS allows policy-based control to aggregation, information-hiding, and abstraction. BGP LS supports IS-IS and OSPFv2.

IGPs do not use BGP LS data from remote peers. BGP does not download the received BGP LS data to any other component on the router.

For segment routing, the following attributes have been added to BGP LS:

- **Node**—Segment routing capability (including SRGB range) and algorithm
- **Link**—Adjacency SID and LAN adjacency SID
- **Prefix**—Prefix SID and segment routing mapping server (SRMS) prefix range

The following example shows how to exchange link-state information with a BGP neighbor:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# router bgp 1
RP/0/RSP0/CPU0:router(config-bgp)# neighbor 10.0.0.2
RP/0/RSP0/CPU0:router(config-bgp-nbr)# remote-as 1
RP/0/RSP0/CPU0:router(config-bgp-nbr)# address-family link-state link-state
RP/0/RSP0/CPU0:router(config-bgp-nbr-af)# exit
```

**IGP Extensions**

A given BGP node may have connections to multiple, independent routing domains; IGP link state distribution into BGP has been added for both OSPF and ISIS protocols to enable that node to pass this information, in a similar fashion, on to applications that desire to build paths spanning or including these multiple domains.

To distribute ISIS link-state data using BGP LS, use the `distribute bgp-ls` command in router configuration mode.

```
RP/0/RSP0/CPU0:router# configure
```
To distribute OSPFv2 and OSPFv3 link-state data using BGP LS, use the distribute bgp-ls command in router configuration mode.

Example: Configuring SR-EPE and BGP-LS

In the following figure, segment routing is enabled on autonomous system AS1 with ingress node A and egress nodes B and C. In this example, we configure EPE on egress node C.

**Figure 1: Topology**

### Step 1
Configure node C with EPE for eBGP peers D and E.

**Example:**

```
RP/0/RSP0/CPU0:router_C(config)# router bgp 1
RP/0/RSP0/CPU0:router_C(config-bgp)# neighbor 192.168.1.3
RP/0/RSP0/CPU0:router_C(config-bgp-nbr)# remote-as 3
RP/0/RSP0/CPU0:router_C(config-bgp-nbr)# description to E
gress-engineering
RP/0/RSP0/CPU0:router_C(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RSP0/CPU0:router_C(config-bgp-nbr-af)# route-policy bgp_in in
RP/0/RSP0/CPU0:router_C(config-bgp-nbr-af)# route-policy bgp_out out
RP/0/RSP0/CPU0:router_C(config-bgp-nbr-af)# exit
RP/0/RSP0/CPU0:router_C(config-bgp-nbr)# exit
```

### Step 2
Configure node C to advertise peer node SIDs to the controller using BGP-LS.
Step 3  Commit the configuration.

Example:

```
RP/0/RSP0/CPU0:router_C(config)# commit
```

Step 4  Verify the configuration.

Example:

```
RP/0/RSP0/CPU0:router_C# show bgp egress-engineering
```

The output shows that node C has allocated peer SIDs for each eBGP peer.

Example:

```
RP/0/RSP0/CPU0:router_C# show mpls forwarding labels 24002 24003
```

<table>
<thead>
<tr>
<th>Local Label</th>
<th>Outgoing Prefix</th>
<th>Outgoing Label or ID</th>
<th>Outgoing Interface</th>
<th>Next Hop</th>
<th>Switched Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>24002</td>
<td>Unlabelled</td>
<td>No ID</td>
<td>Te0/3/0/0</td>
<td>192.168.1.2</td>
<td>0</td>
</tr>
<tr>
<td>24003</td>
<td>Unlabelled</td>
<td>No ID</td>
<td>Te0/1/0/0</td>
<td>192.168.1.3</td>
<td>0</td>
</tr>
</tbody>
</table>
The output shows that node C installed peer node SIDs in the Forwarding Information Base (FIB).
Example: Configuring SR-EPE and BGP-LS
Configure SR-TE Policies

This module provides information about segment routing for traffic engineering (SR-TE) policies, how to configure SR-TE policies, and how to steer traffic into an SR-TE policy.

• About SR-TE Policies, on page 31
• How to Configure SR-TE Policies, on page 31
• Steering Traffic into an SR-TE Policy, on page 35
• Using Binding Segments, on page 39

About SR-TE Policies

Segment routing for traffic engineering (SR-TE) uses a “policy” to steer traffic through the network. An SR-TE policy path is expressed as a list of segments that specifies the path, called a segment ID (SID) list. Each segment is an end-to-end path from the source to the destination, and instructs the routers in the network to follow the specified path instead of the shortest path calculated by the IGP. If a packet is steered into an SR-TE policy, the SID list is pushed on the packet by the head-end. The rest of the network executes the instructions embedded in the SID list.

There are two types of SR-TE policies: dynamic and explicit.

Local Dynamic SR-TE Policy

When you configure local dynamic SR-TE, the head-end locally calculates the path to the destination address. Dynamic path calculation results in a list of interface IP addresses that traffic engineering (TE) maps to adj-SID labels. Routes are learned by way of forwarding adjacencies over the TE tunnel.

Explicit SR-TE Policy

An explicit path is a list of IP addresses or labels, each representing a node or link in the explicit path. This feature is enabled through the explicit-path command that allows you to create an explicit path and enter a configuration submode for specifying the path.

How to Configure SR-TE Policies

This section contains the following procedures:

• Configure Local Dynamic SR-TE Policy, on page 32
Configure Local Dynamic SR-TE Policy

This task explains how to configure a local dynamic SR-TE policy.

**SUMMARY STEPS**

1. `configure`
2. `interface tunnel-te tunnel-id`
3. `ipv4 unnumbered type interface-path-id`
4. `destination ip-address`
5. `path-option preference-priority dynamic segment-routing`
6. `path-protection`
7. `commit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>interface tunnel-te tunnel-id</code></td>
<td>Configures the tunnel interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# interface</td>
<td></td>
</tr>
<tr>
<td>tunnel-te22</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>ipv4 unnumbered type interface-path-id</code></td>
<td>Assigns a source address so that forwarding can</td>
</tr>
<tr>
<td>Example:</td>
<td>be performed on the new tunnel. Loopback is commonly used as the interface type.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered</td>
<td></td>
</tr>
<tr>
<td>loopback0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>destination ip-address</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-if)# destination</td>
<td></td>
</tr>
<tr>
<td>192.168.0.2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>path-option preference-priority dynamic segment-routing</code></td>
<td>Sets the path option to dynamic and assigns the path ID.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-if)# path-option 1</td>
<td></td>
</tr>
<tr>
<td>dynamic segment-routing</td>
<td></td>
</tr>
</tbody>
</table>
Purpose

Command or Action

<table>
<thead>
<tr>
<th>Step 6</th>
<th>path-protection</th>
<th>Enables path protection on the tunnel-te interface.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if)# path-protection</td>
<td></td>
</tr>
</tbody>
</table>

| Step 7 | commit |

This completes the configuration of the dynamic SR-TE policy.

Configure Explicit SR-TE Policy

This task explains how to configure an explicit SR-TE policy.

SUMMARY STEPS

1. configure
2. explicit-path name path-name
3. index index {next-address ip-address | next-label label}
4. exit
5. interface tunnel-te tunnel-id
6. ipv4 unnumbered type interface-path-id
7. destination ip-address [verbatim]
8. path-option preference-priority explicit name path-name segment-routing
9. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 explicit-path name path-name</td>
<td>Enters a name for the explicit path and enters the explicit path configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config)# explicit-path name rlr6_exp</td>
<td></td>
</tr>
<tr>
<td>Step 3 index index {next-address ip-address</td>
<td>next-label label}</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-expl-path)# index 1</code>&lt;br&gt;<code>next-label 16001</code>&lt;br&gt;<code>RP/0/RSP0/CPU0:router(config-expl-path)# index 2</code>&lt;br&gt;<code>next-label 16006</code></td>
<td>- You can include multiple addresses, labels, or both. However, you cannot configure addresses after you have configured labels. Once you start configuring labels, you need to continue with labels.&lt;br&gt;- Each entry must have a unique index.&lt;br&gt;- If the first hop is specified as next-label, that label must be an Adj-SID of the head-end or a prefix-SID label value known by the head-end.</td>
</tr>
</tbody>
</table>

**Step 4**<br>exit

**Step 5**<br>`interface tunnel-te tunnel-id`<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface tunnel-te22`<br>Configures the tunnel interface.

**Step 6**<br>`ipv4 unnumbered type interface-path-id`<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered loopback0`<br>Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.

**Step 7**<br>`destination ip-address [verbatim]`<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# destination 192.168.0.2`<br>Assigns a destination address on the new tunnel. Typically, the tunnel destination must have a match in the routing information base (RIB). For inter-area or inter-domain policies to destinations that are otherwise not reachable, use the `verbatim` option to disable the RIB verification on a tunnel destination.

**Step 8**<br>`path-option preference-priority explicit name path-name segment-routing`<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# path-option 1 explicit name rir6_exp segment-routing`<br>SPECIFIES THE EXPLICIT PATH NAME AND ASSIGN PATH ID.

**Step 9**<br>`commit`<br>This completes the configuration of the explicit SR-TE policy.
Steering Traffic into an SR-TE Policy

This section describes the following traffic steering methods:

**Static Routes**

Static routes can use the segment routing tunnel as a next-hop interface. Both IPv4 and IPv6 prefixes can be routed through the tunnel.

A static route to a destination with a prefix-SID removes the IGP-installed SR-forwarding entry of that prefix.

**Autoroute Announce**

The SR-TE policy can be advertised into an IGP as a next hop by configuring the autoroute announce statement on the source router. The IGP then installs routes in the Routing Information Base (RIB) for shortest paths that involve the tunnel destination. Autoroute announcement of IPv4 prefixes can be carried through either OSPF or IS-IS. Autoroute announcement of IPv6 prefixes can be carried only through IS-IS.

**Autoroute Destination**

Autoroute destination allows you to automatically route traffic through a segment routing tunnel instead of manually configuring static routes. Multiple autoroute destination addresses can be added in the routing information base (RIB) per tunnel.

Static routes are always added with zero cost metric, which can result in traffic that is mapped on multiple tunnels to always load-balance due to ECMP. This load-balancing may be undesirable when some of those tunnels have sub-optimal paths. With autoroute destination, only the tunnel whose IGP cost to its endpoint is lowest will be considered for carrying traffic.

- **Interaction Between Static Routes and Autoroute Destination**

  If there is a manually configured static route to the same destination as a tunnel with autoroute destination enabled, traffic for that destination is load-shared between the static route and the tunnel with autoroute destination enabled.

- **Interaction Between Autoroute Announce and Autoroute Destination**

  For intra-area tunnels, if a tunnel is configured with both autoroute announce and autoroute destination, the tunnel is announced to the RIB by both the IGP and the static process. RIBs prefer static routes, not IGP routes, so the autoroute destination features takes precedence over autoroute announce.

**Configure Static Routes**

This task explains how to configure a static route.

**SUMMARY STEPS**

1. configure
2. interface tunnel-te tunnel-id
3. ipv4 unnumbered type interface-path-id
4. destination ip-address
5. path-option preference-priority dynamic segment-routing
6. `exit`
7. `router static`
8. `address-family ipv4 unicast`
9. `prefix mask interface-type interface-instance`
10. `commit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td></td>
</tr>
</tbody>
</table>
| **Step 2** `interface tunnel-te tunnel-id`  
**Example:**  
RP/0/RSP0/CPU0:router(config)# interface tunnel-te22 | Configures the tunnel interface. |
| **Step 3** `ipv4 unnumbered type interface-path-id`  
**Example:**  
RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered loopback0 | Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type. |
| **Step 4** `destination ip-address`  
**Example:**  
RP/0/RSP0/CPU0:router(config-if)# destination 192.168.0.2 | Assigns a destination address on the new tunnel. |
| **Step 5** `path-option preference-priority dynamic segment-routing`  
**Example:**  
RP/0/RSP0/CPU0:router(config-if)# path-option 1 dynamic segment-routing | Sets the path option to dynamic and assigns the path ID. |
| **Step 6** `exit` | | |
| **Step 7** `router static`  
**Example:**  
RP/0/RSP0/CPU0:router(config)# router static | Configures the static route and enters static configuration mode. |
| **Step 8** `address-family ipv4 unicast`  
**Example:** | Enters address family mode. |
Configure SR-TE Policies

Configure Autoroute Announce

This task explains how to configure autoroute announce to steer traffic through the SR-TE policy.

**SUMMARY STEPS**

1. `configure`
2. `interface tunnel-te tunnel-id`
3. `ipv4 unnumbered type interface-path-id`
4. `autoroute announce`
5. `destination ip-address`
6. `path-option preference-priority dynamic segment-routing`
7. `path-protection`
8. `commit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface tunnel-te tunnel-id</td>
<td>Configures the tunnel interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# interface tunnel-te22</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv4 unnumbered type interface-path-id</td>
<td>Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered loopback0</td>
<td></td>
</tr>
</tbody>
</table>
### Configure Autoroute Destination

This task explains how to configure autoroute destination to steer traffic through the SR-TE policy.

**SUMMARY STEPS**

1. `configure`
2. `interface tunnel-te tunnel-id`
3. `ipv4 unnumbered type interface-path-id`
4. `autoroute destination destination-ip-address`
5. `destination ip-address`
6. `path-option preference-priority dynamic segment-routing`
7. `commit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>Command or Action</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------</td>
</tr>
<tr>
<td></td>
<td>interface tunnel-te tunnel-id</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# interface tunnel-te22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ipv4 unnumbered type interface-path-id</td>
<td>Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered loopback0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>autoroute destination destination-ip-address</td>
<td>(Optional) Adds a route (destination-ip-address) in the RIB with the tunnel as outgoing interface to the tunnel destination.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if)# autoroute destination 192.168.0.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if)# autoroute destination 192.168.0.2 (the default route)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if)# autoroute destination 192.168.0.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if)# autoroute destination 192.168.0.4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>destination ip-address</td>
<td>Assigns a destination address on the new tunnel.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if)# destination 192.168.0.2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>path-option preference-priority dynamic segment-routing</td>
<td>Sets the path option to dynamic and assigns the path ID.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if)# path-option 1 dynamic segment-routing</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

### Using Binding Segments

The binding segment is a local segment identifying an SR-TE policy. Each SR-TE policy is associated with a binding segment ID (BSID). The BSID is a local label that is automatically allocated for each SR-TE policy when the SR-TE policy is instantiated.
BSID can be used to steer traffic into the SR-TE policy and across domain borders, creating seamless end-to-end inter-domain SR-TE policies. Each domain controls its local SR-TE policies; local SR-TE policies can be validated and rerouted if needed, independent from the remote domain’s head-end. Using binding segments isolates the head-end from topology changes in the remote domain.

Packets received with a BSID as top label are steered into the SR-TE policy associated with the BSID. When the BSID label is popped, the SR-TE policy’s SID list is pushed.

BSID can be used in the following cases:

- **Multi-Domain (inter-domain, inter-autonomous system)**—BSIDs can be used to steer traffic across domain borders, creating seamless end-to-end inter-domain SR-TE policies.
- **Large-Scale within a single domain**—The head-end can use hierarchical SR-TE policies by nesting the end-to-end (edge-to-edge) SR-TE policy within another layer of SR-TE policies (aggregation-to-aggregation). The SR-TE policies are nested within another layer of policies using the BSIDs, resulting in seamless end-to-end SR-TE policies.
- **Label stack compression**—If the label-stack size required for an SR-TE policy exceeds the platform capability, the SR-TE policy can be seamlessly stitched to, or nested within, other SR-TE policies using a binding segment.
- **BGP SR-TE Dynamic**—The head-end steers the packet into a BGP-based FIB entry whose next hop is a binding-SID.

**Stitching SR-TE Polices Using Binding SID: Example**

In this intra-domain example, three SR-TE policies are stitched together to form a seamless end-to-end path from node 1 to node 10.

*Figure 2: Intra-Domain Topology*

**Step 1**

Configure an SR-TE policy on node 5 to node 10 via node 9. Node 5 automatically allocates a binding-SID (24012) for the SR-TE policy.

**Example:**

```
RP/0/0/CPU0:xrvr-5(config)# explicit-path name PATH5-9_10
RP/0/0/CPU0:xrvr-5(config-expl-path)# index 10 next-address strict ipv4 unicast 192.168.59.9
RP/0/0/CPU0:xrvr-5(config-expl-path)# index 20 next-address strict ipv4 unicast 10.1.1.10
RP/0/0/CPU0:xrvr-5(config-expl-path)# exit
```
**Step 2**  Configure an SR-TE policy on node 3 to node 5 via node 4 and Link4-6, and push the binding-SID of the SR-TE policy at node 5 (24012) to stitch to the SR-TE policy on node 5. Node 3 automatically allocates a binding-SID (24008) for this SR-TE policy.

**Example:**

```plaintext
RP/0/0/CPU0:xrvr-3(config)# explicit-path name PATH4_4-6_5_BSID
RP/0/0/CPU0:xrvr-3(config-expl-path)# index 10 next-address strict ipv4 unicast 10.1.1.4
RP/0/0/CPU0:xrvr-3(config-expl-path)# index 20 next-address strict ipv4 unicast 192.168.46.6
RP/0/0/CPU0:xrvr-3(config-expl-path)# index 30 next-address strict ipv4 unicast 10.1.1.5
RP/0/0/CPU0:xrvr-3(config-expl-path)# index 40 next-label 24012
RP/0/0/CPU0:xrvr-3(config-expl-path)# exit

RP/0/0/CPU0:xrvr-3(config)# interface tunnel-te1
RP/0/0/CPU0:xrvr-3(config-if)# ipv4 unnumbered Loopback0
RP/0/0/CPU0:xrvr-3(config-if)# destination 10.1.1.10
RP/0/0/CPU0:xrvr-3(config-if)# path-option 1 explicit name PATH4_4-6_5_BSID segment-routing
RP/0/0/CPU0:xrvr-3(config-if)# commit

RP/0/0/CPU0:xrvr-3# show mpls traffic-eng tunnels 1 detail
Name: tunnel-te1  Destination: 10.1.1.10  Ifhandle:0x780
Signalled-Name: xrvr-3_t1
Status:
  Admin: up Oper: up Path: valid Signalling: connected
  path option 1, (Segment-Routing) type explicit PATH4_6_5_BSID (Basis for Setup)
<...>
Binding SID: 24008
<...>
Segment-Routing Path Info (IS-IS 1 level-2)
  Segment0[Node]: 10.1.1.4, Label: 16004
  Segment1[Node]: 10.1.1.5, Label: 16005
  Segment2[ ]: Label: 24012
```

**Step 3**  Configure an SR-TE policy on node 1 to node 3 and push the binding-SID of the SR-TE policy at node 3 (24008) to stitch to the SR-TE policy on node 3.

**Example:**

```plaintext
RP/0/0/CPU0:xrvr-1(config)# explicit-path name PATH3_BSID
RP/0/0/CPU0:xrvr-1(config-expl-path)# index 10 next-address strict ipv4 unicast 10.1.1.3
```
Configure SR-TE Policies

The path is a chain of SR-TE policies stitched together using the binding-SIDs, providing a seamless end-to-end path.

```
RP/0/0/CPU0:xrvr-1# traceroute 10.1.1.10
Type escape sequence to abort.
Tracing the route to 10.1.1.10
1 99.1.2.2 [MPLS: Labels 16003/24008 Exp 0] 29 msec 19 msec 19 msec
2 99.2.3.3 [MPLS: Label 24008 Exp 0] 29 msec 19 msec 19 msec
3 99.3.4.4 [MPLS: Labels 24003/16005/24012 Exp 0] 29 msec 19 msec 19 msec
4 99.4.6.6 [MPLS: Labels 16005/24012 Exp 0] 29 msec 29 msec 19 msec
5 99.5.6.5 [MPLS: Label 24012 Exp 0] 29 msec 29 msec 19 msec
6 99.5.9.9 [MPLS: Label 16010 Exp 0] 19 msec 19 msec 19 msec
7 99.9.10.10 29 msec 19 msec 19 msec
```
Configure Topology-Independent Loop-Free Alternate (TI-LFA)

Topology-Independent Loop-Free Alternate (TI-LFA) uses segment routing to provide link protection in topologies where other fast reroute techniques cannot provide protection. The goal of TI-LFA is to reduce the packet loss that results while routers converge after a topology change due to a link failure. Rapid failure repair (< 50 msec) is achieved through the use of pre-calculated backup paths that are loop-free and safe to use until the distributed network convergence process is completed.

**Note**

TI-LFA supports IPv4 only.

TI-LFA provides link protection. The link is excluded during the post convergence backup path calculation.

- Configuring TI-LFA for IS-IS, on page 43
- Configuring TI-LFA for OSPF, on page 45

### Configuring TI-LFA for IS-IS

This task describes how to enable per-prefix Topology Independent Loop-Free Alternate (TI-LFA) computation to converge traffic flows around link failures.

**Before you begin**

Ensure that the following topology requirements are met:

- Router interfaces are configured as per the topology.
- Routers are configured with IS-IS.
- Segment routing LSPs are configured.

**SUMMARY STEPS**

1. `configure`
2. `router isis instance-id`
3. `interface type interface-path-id`
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.</td>
</tr>
</tbody>
</table>
| **Step 2** `router isis instance-id` | Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.  
**Example:** `RP/0/RSP0/CPU0:router(config)# router isis 1`  
**Note**: You can change the level of routing to be performed by a particular routing instance by using the `is-type` router configuration command. |
| **Step 3** `interface type interface-path-id` | Enters interface configuration mode.  
**Example:** `RP/0/RSP0/CPU0:router(config-isis)# interface GigabitEthernet0/0/2/1`  
`RP/0/RSP0/CPU0:router(config-isis)# interface Bundle-Ether1`  
**Note**: You can configure TI-LFA under Ethernet-based interfaces and logical Bundle-Ether interfaces. |
| **Step 4** `address-family ipv4 [unicast]` | Specifies the IPv4 address family, and enters router address family configuration mode.  
**Example:** `RP/0/RSP0/CPU0:router(config-isis-if)# address-family ipv4 unicast` |
| **Step 5** `fast-reroute per-prefix` | Enables per-prefix fast reroute.  
**Example:** `RP/0/RSP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix` |
| **Step 6** `fast-reroute per-prefix ti-lfa` | Enables per-prefix TI-LFA fast reroute link protection.  
**Example:** `RP/0/RSP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix ti-lfa` |

TI-LFA has been successfully configured for segment routing.
Configuring **TI-LFA for OSPF**

This task describes how to enable per-prefix Topology Independent Loop-Free Alternate (TI-LFA) computation to converge traffic flows around link failures.

**Note**

TI-LFA can be configured on the instance, area, or interface. When configured on the instance or area, all interfaces in the instance or area inherit the configuration.

**Before you begin**

Ensure that the following topology requirements are met:

- Router interfaces are configured as per the topology.
- Routers are configured with OSPF.
- Segment routing LSPs are configured.

**SUMMARY STEPS**

1. configure
2. router ospf process-name
3. area area-id
4. interface type interface-path-id
5. fast-reroute per-prefix
6. fast-reroute per-prefix ti-lfa

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>router ospf process-name</td>
</tr>
<tr>
<td>Example:</td>
<td>Enables OSPF routing for the specified routing process, and places the router in router configuration mode.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# router ospf 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>area area-id</td>
</tr>
<tr>
<td>Example:</td>
<td>Enters area configuration mode.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# area 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>interface type interface-path-id</td>
</tr>
<tr>
<td>Example:</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>Note</td>
<td>You can configure TI-LFA under Ethernet-based interfaces and logical Bundle-Ethernet interfaces.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar)# interface GigabitEthernet0/0/2/1</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>```</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar)# interface Bundle-Ether1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> fast-reroute per-prefix</td>
<td>Enables per-prefix fast reroute.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar-if)# fast-reroute per-prefix</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> fast-reroute per-prefix ti-lfa</td>
<td>Enables per-prefix TI-LFA fast reroute link protection.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar-if)# fast-reroute per-prefix ti-lfa</td>
<td></td>
</tr>
</tbody>
</table>

TI-LFA has been successfully configured for segment routing.
Configure Segment Routing Mapping Server

The mapping server is a key component of the interworking between LDP and segment routing. It enables SR-capable nodes to interwork with LDP nodes. The mapping server advertises Prefix-to-SID mappings in IGP on behalf of other non-SR-capable nodes.

- Segment Routing Mapping Server, on page 47
- Segment Routing and LDP Interoperability, on page 48
- Configuring Mapping Server, on page 50
- Enable Mapping Advertisement, on page 52
- Enable Mapping Client, on page 54

Segment Routing Mapping Server

The mapping server functionality in Cisco IOS XR segment routing centrally assigns prefix-SIDs for some or all of the known prefixes. A router must be able to act as a mapping server, a mapping client, or both.

- A router that acts as a mapping server allows the user to configure SID mapping entries to specify the prefix-SIDs for some or all prefixes. This creates the local SID-mapping policy. The local SID-mapping policy contains non-overlapping SID-mapping entries. The mapping server advertises the local SID-mapping policy to the mapping clients.
- A router that acts as a mapping client receives and parses remotely received SIDs from the mapping server to create remote SID-mapping entries.
- A router that acts as a mapping server and mapping client uses the remotely learnt and locally configured mapping entries to construct the non-overlapping consistent active mapping policy. IGP instance uses the active mapping policy to calculate the prefix-SIDs of some or all prefixes.

The mapping server automatically manages the insertions and deletions of mapping entries to always yield an active mapping policy that contains non-overlapping consistent SID-mapping entries.

- Locally configured mapping entries must not overlap each other.
- The mapping server takes the locally configured mapping policy, as well as remotely learned mapping entries from a particular IGP instance, as input, and selects a single mapping entry among overlapping mapping entries according to the preference rules for that IGP instance. The result is an active mapping policy that consists of non-overlapping consistent mapping entries.
- At steady state, all routers, at least in the same area or level, must have identical active mapping policies.
Segment Routing Mapping Server Restrictions

- The position of the mapping server in the network is not important. However, since the mapping advertisements are distributed in IGP using the regular IGP advertisement mechanism, the mapping server needs an IGP adjacency to the network.

- The role of the mapping server is crucial. For redundancy purposes, you should configure multiple mapping servers in the networks.

- The mapping server functionality does not support a scenario where SID-mapping entries learned through one IS-IS instance are used by another IS-IS instance to determine the prefix-SID of a prefix. For example, mapping entries learnt from remote routers by 'router isis 1' cannot be used to calculate prefix-SIDs for prefixes learnt, advertised, or downloaded to FIB by 'router isis 2'. A mapping server is required for each IS-IS area.

- Segment Routing Mapping Server does not support Virtual Routing and Forwarding (VRF) currently.

Segment Routing and LDP Interoperability

IGP provides mechanisms through which segment routing (SR) interoperate with label distribution protocol (LDP). The control plane of segment routing co-exists with LDP.

The Segment Routing Mapping Server (SRMS) functionality in SR is used to advertise SIDs for destinations, in the LDP part of the network, that do not support SR. SRMS maintains and advertises segment identifier (SID) mapping entries for such destinations. IGP propagates the SRMS mapping entries and interacts with SRMS to determine the SID value when programming the forwarding plane. IGP installs prefixes and corresponding labels, into routing information base (RIB), that are used to program the forwarding information base (FIB).

Example: Segment Routing LDP Interoperability

Consider a network with a mix of segment routing (SR) and label distribution protocol (LDP). A continuous multiprotocol label switching (MPLS) LSP (Labeled Switched Path) can be established by facilitating interoperability. One or more nodes in the SR domain act as segment routing mapping server (SRMS). SRMS advertises SID mappings on behalf of non-SR capable nodes. Each SR-capable node learns about SID assigned to non-SR capable nodes without explicitly configuring individual nodes.

Consider a network as shown in the following image. This network is a mix of both LDP and SR-capable nodes.

In this mixed network:

- Nodes P6, P7, P8, PE4 and PE3 are LDP-capable
- Nodes PE1, PE2, P5 and P6 are SR-capable
- Nodes PE1, PE2, P5 and P6 are configured with segment routing global block (SRGB) of (100, 200)
- Nodes PE1, PE2, P5 and P6 are configured with node segments of 101, 102, 105 and 106 respectively
A service flow must be established from PE1 to PE3 over a continuous MPLS tunnel. This requires SR and LDP to interoperate.

**LDP to SR**

The traffic flow from LDP to SR (right to left) involves:

1. PE3 learns a service route whose nhop is PE1. PE3 has an LDP label binding from the nhop P8 for the FEC PE1. PE3 forwards the packet P8.
2. P8 has an LDP label binding from its nhop P7 for the FEC PE1. P8 forwards the packet to P7.
3. P7 has an LDP label binding from its nhop P6 for the FEC PE1. P7 forwards the packet to P6.
4. P6 does not have an LDP binding from its nhop P5 for the FEC PE1. But P6 has an SR node segment to the IGP route PE1. P6 forwards the packet to P5 and swaps its local LDP label for FEC PE1 by the equivalent node segment 101. This process is called label merging.
5. P5 pops 101, assuming PE1 has advertised its node segment 101 with the penultimate-pop flag set and forwards to PE1.
6. PE1 receives the tunneled packet and processes the service label.

The end-to-end MPLS tunnel is established from an LDP LSP from PE3 to P6 and the related node segment from P6 to PE1.

**SR to LDP**

Suppose that the operator configures P5 as a Segment Routing Mapping Server (SRMS) and advertises the mappings (P7, 107), (P8, 108), (PE3, 103) and (PE4, 104). If PE3 was SR-capable, the operator may have configured PE3 with node segment 103. Because PE3 is non-SR capable, the operator configures that policy at the SRMS; the SRMS advertises the mapping on behalf of the non-SR capable nodes. Multiple SRMS servers can be provisioned in a network for redundancy. The mapping server advertisements are only understood by the SR-capable nodes. The SR capable routers install the related node segments in the MPLS data plane in exactly the same manner if node segments were advertised by the nodes themselves.

The traffic flow from SR to LDP (left to right) involves:

1. PE1 installs the node segment 103 with nhop P5 in exactly the same manner if PE3 had advertised node segment 103.
2. P5 swaps 103 for 103 and forwards to P6.
3. The nhop for P6 for the IGP route PE3 is non-SR capable. (P7 does not advertise the SR capability.) However, P6 has an LDP label binding from that nhop for the same FEC. (For example, LDP label 1037.) P6 swaps 103 for 1037 and forwards to P7. We refer to this process as label merging.
4. P7 swaps this label with the LDP label received from P8 and forwards to P8.
5. P8 pops the LDP label and forwards to PE3.
6. PE3 receives the packet and processes as required.

The end-to-end MPLS LSP is established from an SR node segment from PE1 to P6 and an LDP LSP from P6 to PE3.
Configuring Mapping Server

Perform these tasks to configure the mapping server and to add prefix-SID mapping entries in the active local mapping policy.

**SUMMARY STEPS**

1. **configure**
2. **segment-routing**
3. **mapping-server**
4. **prefix-sid-map**
5. **address-family ipv4 | ipv6**
6. **ip-address/prefix-length first-SID-value range range**
7. **commit**
8. **show segment-routing mapping-server prefix-sid-map [ipv4 | ipv6] [detail]**

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>segment-routing</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>mapping-server</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>prefix-sid-map</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>address-family ipv4</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| `RP/0/RSP0/CPU0:router(config-sr-ms-map)# address-family ipv6` | Adds SID-mapping entries in the active local mapping policy. In the configured example:  
  - Prefix 10.1.1.1/32 is assigned prefix-SID 10, prefix 10.1.1.2/32 is assigned prefix-SID 11,..., prefix 10.1.1.199/32 is assigned prefix-SID 200  
  - Prefix 20.1.0.0/16 is assigned prefix-SID 400, prefix 20.2.0.0/16 is assigned prefix-SID 401,..., and so on. |

**Step 6**  
ip-address/prefix-length first-SID-value range range  

**Example:**  
```
RP/0/RSP0/CPU0:router(config-sr-ms-map-af)# 10.1.1.1/32 10 range 200  
RP/0/RSP0/CPU0:router(config-sr-ms-map-af)# 20.1.0.0/16 400 range 300
```

**Step 7**  
`commit`  

**Step 8**  
`show segment-routing mapping-server prefix-sid-map [ipv4 | ipv6] [detail]`  

**Example:**  
```
RP/0/RSP0/CPU0:router# show segment-routing mapping-server prefix-sid-map ipv4  
Prefix SID Index Range  
Flags  
20.1.1.0/24 400 300  
10.1.1.1/32 10 200  
Number of mapping entries: 2
```

**Note**  
Specify the address family for IS-IS.

What to do next  
Enable the advertisement of the local SID-mapping policy in the IGP.
Enable Mapping Advertisement

In addition to configuring the static mapping policy, you must enable the advertisement of the mappings in the IGP.

Perform these steps to enable the IGP to advertise the locally configured prefix-SID mapping.

Configure Mapping Advertisement for IS-IS

SUMMARY STEPS

1. `router isis instance-id`
2. `address-family { ipv4 | ipv6 } [ unicast ]`
3. `segment-routing prefix-sid-map advertise-local`
4. `commit`
5. `show isis database verbose`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.</td>
</tr>
<tr>
<td><code>router isis instance-id</code></td>
<td>You can change the level of routing to be performed by a particular routing instance by using the <code>is-type</code> router configuration command.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config)# router isis 1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Specifies the IPv4 or IPv6 address family, and enters router address family configuration mode.</td>
</tr>
<tr>
<td>`address-family { ipv4</td>
<td>ipv6 } [ unicast ]`</td>
</tr>
<tr>
<td>Example:</td>
<td>The following is an example for ipv4 address family:</td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-isis)# address-family ipv4 unicast</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Configures IS-IS to advertise locally configured prefix-SID mappings.</td>
</tr>
<tr>
<td><code>segment-routing prefix-sid-map advertise-local</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-isis-af)# segment-routing prefix-sid-map advertise-local</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td><code>commit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Displays IS-IS prefix-SID mapping advertisement and TLV.</td>
</tr>
<tr>
<td><code>show isis database verbose</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router# show isis database verbose</code></td>
<td><code>&lt;...removed...&gt;</code></td>
</tr>
</tbody>
</table>
Configure Mapping Advertisement for OSPF

SUMMARY STEPS

1. `router ospf process-name`
2. `segment-routing prefix-sid-map advertise-local`
3. `commit`
4. `show ospf database opaque-area`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>router ospf process-name</code></td>
<td>Enables OSPF routing for the specified routing instance, and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# <code>router ospf 1</code></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>segment-routing prefix-sid-map advertise-local</code></td>
<td>Configures OSPF to advertise locally configured prefix-SID mappings.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-ospf)# <code>segment-routing prefix-sid-map advertise-local</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>commit</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>show ospf database opaque-area</code></td>
<td>Displays OSPF prefix-SID mapping advertisement and TLV.</td>
</tr>
</tbody>
</table>
| Example: | RP/0/RSP0/CPU0:router# `show ospf database opaque-area` | `<...removed...>`

Extended Prefix Range TLV: Length: 24
  AF         : 0
  Prefix     : 10.1.1.1/32
  Range Size : 200
  Flags      : 0x0

SID sub-TLV: Length: 8
Enable Mapping Client

By default, mapping client functionality is enabled.

You can disable the mapping client functionality by using the **segment-routing prefix-sid-map receive disable** command.

You can re-enable the mapping client functionality by using the **segment-routing prefix-sid-map receive** command.

The following example shows how to enable the mapping client for IS-IS:

```
RP/0/RSP0/CPU0:router(config)# router isis 1
RP/0/RSP0/CPU0:router(config-isis)# address-family ipv4 unicast
RP/0/RSP0/CPU0:router(config-isis-af)# segment-routing prefix-sid-map receive
```

The following example shows how to enable the mapping client for OSPF:

```
RP/0/RSP0/CPU0:router(config)# router ospf 1
RP/0/RSP0/CPU0:router(config-ospf)# segment-routing prefix-sid-map receive
```
CHAPTER 10

Using Segment Routing Traffic Matrix

This module provides information about the Segment Routing Traffic Matrix (SR-TM) and the Traffic Collector process, and describes how to configure the TM border and the Traffic Collector and to display traffic information.

- Segment Routing Traffic Matrix, on page 55
- Traffic Collector Process, on page 55
- Configuring Traffic Collector, on page 56
- Displaying Traffic Information, on page 57

Segment Routing Traffic Matrix

A network's traffic matrix is a description, measure, or estimation of the aggregated traffic flows that enter, traverse, and leave a network.

The Segment Routing Traffic Matrix (SR-TM) is designed to help users understand traffic patterns on a router. The Traffic Matrix border divides the network into two parts: internal (interfaces that are inside the border) and external (interfaces that are outside the border). By default, all interfaces are internal. You can configure an interface as external.

Traffic Collector Process

The Traffic Collector collects packet and byte statistics from router components such as prefix counters, tunnel counters, and the TM counter, which increments when traffic that comes from an external interface to the network is destined for a segment routing prefix-SID. The Traffic Collector keeps histories of the statistics and makes them persistent across process restarts, failovers, and ISSU. Histories are retained for a configurable length of time.

Pcounters

A Pcounter is a packet and byte pair of counters. There is one Pcounter per tunnel. There are two Pcounters per prefix-SID:

- Base Pcounter – any packet that is switched on the prefix-SID forwarding information base (FIB) entry
- TM Pcounter – any packet from an external interface and switched on the prefix-SID FIB entry
The Traffic Collector periodically collects the Base Pcounters and TM Pcounters of all prefix-SIDs, and the Pcounters of all tunnel interfaces.

For each Pcounter, the Traffic Collector calculates the number of packets and bytes that have been forwarded during the last interval. The Traffic Collector keeps a history of the per-interval statistics for each of the Pcounters. Each entry in the history contains:

- The start and end time of the interval
- The number of packets forwarded during the interval
- The number of bytes forwarded during the interval

### Configuring Traffic Collector

Perform these tasks to configure the traffic collector.

**SUMMARY STEPS**

1. `configure`
2. `traffic-collector`
3. `statistics collection-interval value`
4. `statistics history-size value`
5. `statistics history-timeout value`
6. `interface type l3-interface-address`
7. `commit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enables traffic collector and places the router in traffic collector configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> traffic-collector</td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# traffic-collector</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> statistics collection-interval value</td>
<td>(Optional) Sets the frequency that the traffic collector collects and posts data, in minutes. Valid values are 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, and 60. The default interval is 1.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-tc)# statistics collection-interval 5</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> statistics history-size value</td>
<td>(Optional) Specifies the number of entries kept in the history database. Valid values are from 1 to 10. The default is 5.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-tc)# statistics</code> <code>history-size 10</code></td>
<td><strong>Note</strong> The number of entries affects how the average packet and average byte rates are calculated. The rates are calculated over the range of the histories and are not averages based in real time.</td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>statistics history-timeout value</code> <strong>Example:</strong> <code>RP/0/RP0/CPU0:router(config-tc)# statistics history-timeout 24</code></td>
<td>(Optional) When a prefix SID or a tunnel-te interface is deleted, the history-timeout sets the length of time, in hours, that the prefix SID and tunnel statistics are retained in the history before they are removed. The minimum is one hour; the maximum is 720 hours. The default is 48. <strong>Note</strong> Enter 0 to disable the history timeout. (No history is retained.)</td>
</tr>
<tr>
<td><strong>Step 6</strong> <code>interface type l3-interface-address</code> <strong>Example:</strong> <code>RP/0/RP0/CPU0:router(config-tc)# interface TenGigE 0/1/0/3</code></td>
<td>Identifies interfaces that handle external traffic. Only L3 interfaces are supported for external traffic.</td>
</tr>
<tr>
<td><strong>Step 7</strong> <code>commit</code></td>
<td></td>
</tr>
</tbody>
</table>

This completes the configuration for the traffic collector.

## Displaying Traffic Information

The following show commands display information about the interfaces and tunnels:

- Display the configured external interfaces:

  ```
  RP/0/RSP0/CPU0:router# show traffic-collector external-interface
  Interface   Status
  --------------
  Te0/1/0/3    Enabled
  Te0/1/0/4    Enabled
  ```

- Display the counter history database for a prefix-SID:

  ```
  RP/0/RSP0/CPU0:router# show traffic-collector ipv4 counters prefix 1.1.1.10/32 detail
  Prefix: 1.1.1.10/32   Label: 16010   State: Active
  Base:
  Average over the last 5 collection intervals:
  Packet rate: 9496937 pps, Byte rate: 9363979882 Bps
  ```
History of counters:

23:01 – 23:02: Packets 9379529, Bytes: 9248215594
23:00 – 23:01: Packets 9687124, Bytes: 9551504264
22:59 – 23:00: Packets 9539200, Bytes: 9405651200

**TM Counters:**

Average over the last 5 collection intervals:

Packet rate: 9528754 pps, Byte rate: 9357236821 Bps

History of counters:

23:01 - 23:02: Packets 9400815, Bytes: 9231600330
23:00 - 23:01: Packets 9699455, Bytes: 9524864810
22:59 - 23:00: Packets 9579889, Bytes: 9407450998

This output shows the average Pcounter (packets, bytes), the Pcounter history, and the collection interval for the specified prefix-SID.

- Display the counter history database for a tunnel:

```
RP/0/RSP0/CPU0:router# show traffic-collector counters tunnels tunnel-te 1 detail
Tunnel: tt1 State: Active
Average over the last 5 collection intervals:
Packet rate: 9694434 pps, Byte rate: 9597489858 Bps
```

History of counters:

23:14 – 23:15: Packets 9870522 , Bytes: 9771816780
23:13 – 23:14: Packets 9553048 , Bytes: 9457517520
23:11 – 23:12: Packets 9756654 , Bytes: 9659087460
23:10 – 23:11: Packets 9694434 , Bytes: 9548235180

This output shows the average Pcounter (packets, bytes), the Pcounter history, and the collection interval for the tunnel.
Using Segment Routing OAM

Segment Routing Operations, Administration, and Maintenance (OAM) helps service providers to monitor label-switched paths (LSPs) and quickly isolate forwarding problems to assist with fault detection and troubleshooting in the network. The Segment Routing OAM feature provides support for Nil-FEC (forwarding equivalence classes) LSP Ping and Traceroute functionality.

- MPLS LSP Ping and Traceroute Nil FEC Target, on page 59
- Examples: LSP Ping and Traceroute for Nil_FEC Target, on page 60

MPLS LSP Ping and Traceroute Nil FEC Target

The Nil-FEC LSP ping and traceroute operations are extensions of regular MPLS ping and traceroute. Nil-FEC LSP Ping/Traceroute functionality supports segment routing and MPLS Static. It also acts as an additional diagnostic tool for all other LSP types. This feature allows operators to provide the ability to freely test any label stack by allowing them to specify the following:

- label stack
- outgoing interface
- nexthop address

In the case of segment routing, each segment nodal label and adjacency label along the routing path is put into the label stack of an echo request message from the initiator Label Switch Router (LSR); MPLS data plane forwards this packet to the label stack target, and the label stack target sends the echo message back.

The following table shows the syntax for the ping and traceroute commands.

<table>
<thead>
<tr>
<th>Table 2: LSP Ping and Traceroute Nil FEC Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Syntax</td>
</tr>
<tr>
<td>ping mpls nil-fec labels {label[,label]} [output {interface tx-interface} [nexthop nexthop-ip-addr]]</td>
</tr>
<tr>
<td>traceroute mpls nil-fec labels {label[,label]} [output {interface tx-interface} [nexthop nexthop-ip-addr]]</td>
</tr>
</tbody>
</table>
Examples: LSP Ping and Traceroute for Nil_FEC Target

These examples use the following topology:

- Node loopback IP address: 172.18.1.3 172.18.1.4 172.18.1.5 172.18.1.7
- Node label: 16004 16005 16007
- Nodes: Arizona ---- Utah ------- Wyoming ---- Texas
- Interface: GigabitEthernet0/2/0/1 GigabitEthernet0/2/0/1
- Interface IP address: 10.1.1.3 10.1.1.4

```
RP/0/RSP0/CPU0:router-utah# show mpls forwarding
```

```
Tue Jul 5 13:44:31.999 EDT
Local Outgoing Prefix Outgoing Next Hop Bytes
Label Label or ID Interface Switched
-------- ----------- ------------------ ------------ --------------- ------------
16004 Pop No ID Gi0/2/0/1 10.1.1.4 1392
16005 Pop No ID Gi0/2/0/2 10.1.2.2 0
16007 Pop No ID Gi0/2/0/0 10.1.1.4 0
------------------ --------------- --------------- ---------------
16005 16005 No ID Gi0/2/0/0 10.1.1.4 0
16005 No ID Gi0/2/0/1 10.1.2.2 0
16007 16007 No ID Gi0/2/0/0 10.1.1.4 4752
16007 No ID Gi0/2/0/1 10.1.2.2 0
24000 Pop SR Adj (idx 0) Gi0/2/0/0 10.1.1.4 0
24001 Pop SR Adj (idx 2) Gi0/2/0/0 10.1.1.4 0
24002 Pop SR Adj (idx 0) Gi0/2/0/1 10.1.2.2 0
24003 Pop SR Adj (idx 2) Gi0/2/0/1 10.1.2.2 0
24004 Pop No ID tt10 point2point 0
24005 Pop No ID tt11 point2point 0
24006 Pop No ID tt12 point2point 0
24007 Pop No ID tt13 point2point 0
24008 Pop No ID tt30 point2point 0
```

```
Ping Nil FEC Target
```

```
RP/0/RSP0/CPU0:router-arizona# ping mpls nil-fec labels 16005,16007 output interface GigabitEthernet 0/2/0/1 nexthop 10.1.1.4 repeat 1
```

Sending 1, 72-byte MPLS Echos with Nil FEC labels 16005,16007, timeout is 2 seconds, send interval is 0 msec:

```
```

Type escape sequence to abort.
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 1/1/1 ms
Total Time Elapsed 0 ms
Traceroute Nil FEC Target

RP/0/RSP0/CPU0:router-arizona# traceroute mpls nil-fec labels 16005,16007 output interface GigabitEthernet 0/2/0/1 nexthop 10.1.1.4
Tracing MPLS Label Switched Path with Nil FEC labels 16005,16007, timeout is 2 seconds


Type escape sequence to abort.

0 10.1.1.3 MRU 1500 [Labels: 16005/16007/explicit-null Exp: 0/0/0]
L 1 10.1.1.4 MRU 1500 [Labels: implicit-null/16007/explicit-null Exp: 0/0/0] 1 ms
L 2 10.1.1.5 MRU 1500 [Labels: implicit-null/explicit-null Exp: 0/0] 1 ms
! 3 10.1.1.7 1 ms
Examples: LSP Ping and Traceroute for Nil_FEC Target