THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE. IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2016 Cisco Systems, Inc. All rights reserved.
CONTENTS

PREFACE

Preface vii
Communications, Services, and Additional Information vii

CHAPTER 1

About Segment Routing 1
Scope 1
Need 2
Benefits 2
Workflow for Deploying Segment Routing 3

CHAPTER 2

Configure Segment Routing Global Block 5
About the Segment Routing Global Block 5
Setup a Non-Default Segment Routing Global Block Range 6

CHAPTER 3

Configure Segment Routing for IS-IS Protocol 9
Enabling Segment Routing for IS-IS Protocol 9
Configuring a Prefix-SID on the IS-IS Enabled Loopback Interface 11

CHAPTER 4

Configure Segment Routing for OSPF Protocol 15
Enabling Segment Routing for OSPF Protocol 15
Configuring a Prefix-SID on the OSPF-Enabled Loopback Interface 17

CHAPTER 5

Configure Segment Routing for BGP 21
Segment Routing for BGP 21
Configure BGP Prefix Segment Identifiers 22
Configure Segment Routing Egress Peer Engineering 23
Configure BGP Link-State 24
Example: Configuring SR-EPE and BGP-LS  25

CHAPTER 6  Configure SR-TE Policies  29
About SR-TE Policies  29
How to Configure SR-TE Policies  29
  Configure Local Dynamic SR-TE Policy  30
  Configure Explicit SR-TE Policy  31
Steering Traffic into an SR-TE Policy  33
  Configure Static Routes  33
  Configure Autoroute Announce  35
  Configure Autoroute Destination  37
Using Binding Segments  38
  Stitching SR-TE Policies Using Binding SID: Example  39

CHAPTER 7  Configure Topology-Independent Loop-Free Alternate (TI-LFA)  43
  Configuring TI-LFA for IS-IS  43
  Configuring TI-LFA for OSPF  45
  TI-LFA Node and SRLG Protection: Examples  47

CHAPTER 8  Configure Segment Routing Mapping Server  49
  Segment Routing Mapping Server  49
    Segment Routing Mapping Server Restrictions  50
  Segment Routing and LDP Interoperability  50
    Example: Segment Routing LDP Interoperability  50
  Configuring Mapping Server  52
  Enable Mapping Advertisement  54
    Configure Mapping Advertisement for IS-IS  54
    Configure Mapping Advertisement for OSPF  55
  Enable Mapping Client  56

CHAPTER 9  Using Segment Routing Traffic Matrix  57
  Segment Routing Traffic Matrix  57
  Traffic Collector Process  57
  Configuring Traffic Collector  58
Displaying Traffic Information 59

CHAPTER 10 Using Segment Routing OAM 61
  MPLS LSP Ping and Traceroute Nil FEC Target 61
  Examples: LSP Ping and Traceroute for Nil_FEC Target 62
Preface

The Cisco IOS XR Segment Routing Configuration Guide for the Cisco CRS Router preface contains these sections:

- Communications, Services, and Additional Information, on page vii

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.
About Segment Routing

Note
Segment Routing is supported on the CRS-X and CRS-3 cards.

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

• Scope, on page 1
• Need, on page 2
• Benefits, on page 2
• Workflow for Deploying Segment Routing, on page 3

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 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 .

**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 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
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 5
- Setup a Non-Default Segment Routing Global Block Range, on page 6

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>Step 1</td>
<td>configure</td>
<td>(Optional) Enter the router isis instance-id or router ospf process_name commands if you want to configure separate SRGBs for IS-IS and OSPF protocols.</td>
</tr>
<tr>
<td>Step 2</td>
<td>[router {isis instance-id</td>
<td>ospf process_name} ]</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# router isis 1</td>
<td></td>
</tr>
<tr>
<td>Step 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>Example:</td>
<td>RP/0/RP0/CPU0:router(config-isis)# segment-routing global-block 18000 19999</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>commit</td>
<td>verify the SRGB configuration:</td>
</tr>
</tbody>
</table>

Verify the SRGB configuration:

RP/0/RP0/CPU0:router# show mpls label table detail
Table Label Owner State Rewrite
----- ------ ------------------------------- -------
Configure Segment Routing Global Block

Setup a Non-Default Segment Routing Global Block Range

What to do next
Configure prefix SIDs and enable segment routing.
Setup a Non-Default Segment Routing Global Block Range
Configure Segment Routing for IS-IS Protocol


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

For additional information on implementing IS-IS on your Cisco CRS Router, see the Implementing IS-IS module in the Cisco IOS XR Routing Configuration Guide for the Cisco CRS Router.

**Note**

- Enabling Segment Routing for IS-IS Protocol, on page 9
- Configuring a Prefix-SID on the IS-IS Enabled Loopback Interface, on page 11

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><code>configure</code></td>
<td></td>
</tr>
</tbody>
</table>
| Step 2 | `router isis instance-id`  
**Example:**  
RP/0/RP0/CPU0:router(config)# `router isis isp` | Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.  
**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 | `address-family { ipv4 | ipv6 } [ unicast ]`  
**Example:**  
RP/0/RP0/CPU0:router(config-isis)# `address-family ipv4 unicast` | Specifies the IPv4 or IPv6 address family, and enters router address family configuration mode. |
| Step 4 | `metric-style wide [ level { 1 | 2 } ]`  
**Example:**  
RP/0/RP0/CPU0:router(config-isis-af)# `metric-style wide level 1` | Configures a router to generate and accept only wide link metrics in the Level 1 area. |
| Step 5 | `mpls traffic-eng level`  
**Example:**  
RP/0/RP0/CPU0:router(config-isis-af)# `mpls traffic-eng level-2-only` | Enables RSVP traffic engineering functionality. |
Configure Segment Routing for IS-IS Protocol

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

A prefix segment identifier (SID) is associated with an IP prefix. The prefix SID is manually configured from the segment routing global block (SRGB) range of labels. A prefix SID is configured under the loopback interface.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong></td>
<td>Sets the traffic engineering loopback interface.</td>
</tr>
<tr>
<td>mpls traffic-eng router-id interface</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>Step 7</td>
<td>Configures router ID for each address-family (ipv4/ipv6).</td>
</tr>
<tr>
<td>router-id loopback</td>
<td>Segment routing is enabled by the following actions:</td>
</tr>
<tr>
<td>Example:</td>
<td>• MPLS forwarding is enabled on all interfaces where IS-IS is active.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-isis-af)# mpls traffic-eng router-id Loopback0</td>
<td>• All known prefix-SIDs in the forwarding plane are programmed, with the prefix-SIDs advertised by remote routers or learned through local or remote mapping server.</td>
</tr>
<tr>
<td>Step 8</td>
<td>• The prefix-SIDs locally configured are advertised.</td>
</tr>
<tr>
<td>segment-routing mpls</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-isis-af)# segment-routing mpls</td>
<td></td>
</tr>
<tr>
<td>Step 9</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>exit</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-isis-af)# exit</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-isis)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 10</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>mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>Step 11</td>
<td>What to do next Configure the prefix SID.</td>
</tr>
<tr>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

**What to do next**

Configure the prefix SID.
interface with the loopback address of the node as the prefix. The prefix segment steers the traffic along the shortest path to its destination.

A prefix SID can be a node SID or an Anycast SID. A node SID is a type of prefix SID that identifies a specific node. An Anycast SID is a type of prefix SID that identifies a set of nodes, and is configured with n-flag clear. The set of nodes (Anycast group) is configured to advertise a shared prefix address and prefix SID. Anycast routing enables the steering of traffic toward multiple advertising nodes. Packets addressed to an Anycast address are forwarded to the topologically nearest nodes.

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><strong>configure</strong></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>router isis instance-id</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# router isis 1</td>
</tr>
<tr>
<td>Purpose:</td>
<td>Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.</td>
</tr>
</tbody>
</table>
| &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&n...
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong></td>
<td>Configures the prefix-SID index or absolute value for the interface. Specify <code>index SID-index</code> for each node to create a prefix SID based on the lower boundary of the SRGB + the index. Specify <code>absolute SID-value</code> 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 <code>n-flag-clear</code> 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 <code>explicit-null</code> keyword. IS-IS sets the E flag in the prefix-SID sub TLV.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>prefix-sid index 1001</code></td>
<td></td>
</tr>
<tr>
<td><code>prefix-sid absolute 17001</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>

Verify the prefix-SID configuration:

```
RP/0/RP0/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
  NLSPID: 0xcc
  NLSPID: 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.
Configuring a Prefix-SID on the IS-IS Enabled Loopback Interface
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 CRS Router, see the Implementing OSPF module in the Cisco IOS XR Routing Configuration Guide for the Cisco CRS Router.

- Enabling Segment Routing for OSPF Protocol, on page 15
- Configuring a Prefix-SID on the OSPF-Enabled Loopback Interface, on page 17

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> configure</td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> router ospf  process-name</td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# router ospf 1</td>
</tr>
<tr>
<td><strong>Step 3</strong> segment-routing mpls</td>
<td>Enables segment routing using the MPLS data plane on the routing process and all areas and interfaces in the routing process. Enables segment routing forwarding on all interfaces in the routing process and installs the SIDs received by OSPF in the forwarding table.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-ospf)# segment-routing mpls</td>
</tr>
<tr>
<td><strong>Step 4</strong> area 0</td>
<td>Enters area configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-ospf)# area 0</td>
</tr>
<tr>
<td><strong>Step 5</strong> mpls traffic-eng area</td>
<td>Enables IGP traffic engineering functionality.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-ospf-ar)# mpls traffic-eng area 0</td>
</tr>
<tr>
<td><strong>Step 6</strong> mpls traffic-eng router-id interface</td>
<td>Sets the traffic engineering loopback interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
Configure Segment Routing for OSPF Protocol

Configuring a Prefix-SID on the OSPF-Enabled Loopback Interface

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpls traffic-eng router-id Loopback0</td>
<td>(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.</td>
</tr>
<tr>
<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 SR-TE head-end requires the TED to calculate and validate the path of the SR-TE policy.</td>
</tr>
<tr>
<td>mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

What to do next
Configure the prefix SID.

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

A prefix SID can be a node SID or an Anycast SID. A node SID is a type of prefix SID that identifies a specific node. An Anycast SID is a type of prefix SID that identifies a set of nodes, and is configured with n-flag clear. The set of nodes (Anycast group) is configured to advertise a shared prefix address and prefix SID. Anycast routing enables the steering of traffic toward multiple advertising nodes. Packets addressed to an Anycast address are forwarded to the topologically nearest nodes.

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
4. interface Loopback  interface-instance
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> configure</td>
<td>Enables OSPF routing for the specified routing process, and places the router in router configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> router ospf  process-name</td>
<td>Enables OSPF routing for the specified routing process, and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# router ospf 1</td>
</tr>
<tr>
<td><strong>Step 3</strong> area  value</td>
<td>Enters area configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-ospf)# area 0</td>
</tr>
<tr>
<td><strong>Step 4</strong> interface Loopback  interface-instance</td>
<td>Specifies the loopback interface and instance.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-ospf-ar)# interface Loopback0 passive</td>
</tr>
<tr>
<td><strong>Step 5</strong> prefix-sid {index  SID-index</td>
<td>absolute  SID-value  }  [n-flag-clear  ]  [explicit-null]</td>
</tr>
<tr>
<td>Example:</td>
<td>Specify  index  SID-index  for each node to create a prefix SID based on the lower boundary of the SRGB + the index.</td>
</tr>
<tr>
<td>Specify  absolute  SID-value  for each node to create a specific prefix SID within the SRGB.</td>
<td></td>
</tr>
<tr>
<td>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. OSPF does not set the n flag in the prefix-SID sub Type Length Value (TLV).</td>
<td></td>
</tr>
<tr>
<td>To disable penultimate-hop-popping (PHP) and add an explicit-Null label, enter the explicit-null keyword. OSPF sets the n flag in the prefix-SID sub TLV.</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-ospf-ar)# prefix-sid index 1001</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-ospf-ar)# prefix-sid absolute 17001</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> commit</td>
<td>Verify the prefix-SID configuration:</td>
</tr>
</tbody>
</table>
RP/0/RP0/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
Configuring a Prefix-SID on the OSPF-Enabled Loopback Interface
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 CRS Router, see the Implementing BGP module in the Cisco IOS XR Routing Configuration Guide for the Cisco CRS Router.

- Segment Routing for BGP, on page 21
- Configure BGP Prefix Segment Identifiers, on page 22
- Configure Segment Routing Egress Peer Engineering, on page 23
- Configure BGP Link-State, on page 24
- Example: Configuring SR-EPE and BGP-LS, on page 25

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 IOS XR Routing Configuration Guide for the Cisco CRS Router.

---

**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)# address-family ipv4 unicast
RP/0/RSP0/CPU0:router(config-bgp-af)# network 1.1.1.3/32 route-policy SID(3)
RP/0/RSP0/CPU0:router(config-bgp-af)# commit
RP/0/RSP0/CPU0:router(config)#
```

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

Note
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(config-bgp-nbr))# remote-as 3
```

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
```
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**

![Topology Diagram]

**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
RP/0/RSP0/CPU0:router C(config-bgp-nbr)# egress-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
RP/0/RSP0/CPU0:router C(config-bgp)# neighbor 192.168.1.2
RP/0/RSP0/CPU0:router C(config-bgp-nbr)# remote-as 2
RP/0/RSP0/CPU0:router C(config-bgp-nbr)# description to D
egress-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.
Example:

```
RP/0/RSP0/CPU0:router_C(config-bgp)# neighbor 172.29.50.71
RP/0/RSP0/CPU0:router_C(config-bgp-nbr)# remote-as 1
RP/0/RSP0/CPU0:router_C(config-bgp-nbr)# description to EPE_controller
RP/0/RSP0/CPU0:router_C(config-bgp-nbr)# address-family link-state link-state
RP/0/RSP0/CPU0:router_C(config-bgp-nbr)# exit
RP/0/RSP0/CPU0:router_C(config-bgp)# exit
```

**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(config)# show bgp egress-engineering
```

```
Egress Engineering Peer Set: 192.168.1.2/32 (10b87210)
   Nexthop: 192.168.1.2
   Version: 2, rn_version: 2
   Flags: 0x00000002
   Local ASN: 1
   Remote ASN: 2
   Local RID: 1.1.1.3
   Remote RID: 1.1.1.4
   First Hop: 192.168.1.2
   NHID: 3
   Label: 24002, Refcount: 3
   rpc_set: 10b9d408

Egress Engineering Peer Set: 192.168.1.3/32 (10be61d4)
   Nexthop: 192.168.1.3
   Version: 3, rn_version: 3
   Flags: 0x00000002
   Local ASN: 1
   Remote ASN: 3
   Local RID: 1.1.1.3
   Remote RID: 1.1.1.5
   First Hop: 192.168.1.3
   NHID: 4
   Label: 24003, Refcount: 3
   rpc_set: 10be6250
```

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
```

```
Local Outgoing Prefix Outgoing Next Hop Bytes
Label or ID Interface Switched
------- ----------- ------------------ ------------ --------------- ------------
Unlabelled No ID Te0/3/0/0 192.168.1.2 0
Unlabelled No ID Te0/1/0/0 192.168.1.3 0
```
The output shows that node C installed peer node SIDs in the Forwarding Information Base (FIB).
Example: Configuring SR-EPE and BGP-LS
Chapter 6

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 29
- How to Configure SR-TE Policies, on page 29
- Steering Traffic into an SR-TE Policy, on page 33
- Using Binding Segments, on page 38

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 30
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. `ipv6 enable`
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> <code>interface tunnel-te tunnel-id</code></td>
<td>Configures the tunnel interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config)# interface tunnel-te22</code></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 be performed on the new tunnel. Loopback is commonly used as the interface type.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback0</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>ipv6 enable</code></td>
<td>Enables IPv6 over the IPv4 tunnel.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>IPv6 over IPv4 tunnel is supported with CRS-FP-X next generation line cards.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-if)# ipv6 enable</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>destination ip-address</code></td>
<td>Assigns a destination address on the new tunnel.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-if)# destination 192.168.0.2</code></td>
<td></td>
</tr>
</tbody>
</table>
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. ipv6 enable
8. destination ip-address [verbatim]
9. path-option preference-priority explicit name path-name segment-routing
10. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
</tr>
<tr>
<td>Step 2</td>
<td>explicit-path name path-name</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# explicit-path name rir6_exp</td>
</tr>
<tr>
<td>Step 3</td>
<td>index index {next-address ip-address</td>
</tr>
<tr>
<td></td>
<td>Specifies a label or an address in an explicit path of a tunnel.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Example:</strong></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.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>• Each entry must have a unique index.</td>
</tr>
<tr>
<td></td>
<td>• 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>
<tr>
<td>Step 4</td>
<td>exit</td>
</tr>
<tr>
<td>Step 5</td>
<td>interface tunnel-te tunnel-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>ipv4 unnumbered type interface-path-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td>ipv6 enable</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>IPv6 over IPv4 tunnel is supported with CRS-FP-X next generation line cards.</td>
</tr>
<tr>
<td>Step 8</td>
<td>destination ip-address [verbatim]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Step 9</td>
<td>path-option preference-priority explicit name path-name segment-routing</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>
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. ipv6 enable
5. destination ip-address
6. path-option preference-priority dynamic segment-routing
7. exit
8. router static
9. address-family ipv4 unicast
10. prefix mask interface-type interface-instance
11. 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>Configures the tunnel interface.</td>
</tr>
<tr>
<td><strong>Step 2</strong> interface tunnel-te tunnel-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>例:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# interface tunnel-te22</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv4 unnumbered type interface-path-id</td>
<td>Enables IPv6 over the IPv4 tunnel.</td>
</tr>
<tr>
<td>例:</td>
<td>Note IPv6 over IPv4 tunnel is supported with CRS-FP-X next generation line cards.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ipv6 enable</td>
<td>Assigns a destination address on the new tunnel.</td>
</tr>
<tr>
<td>例:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# ipv6 enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> destination ip-address</td>
<td>Sets the path option to dynamic and assigns the path ID.</td>
</tr>
<tr>
<td>例:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# destination 192.168.0.2</td>
<td></td>
</tr>
</tbody>
</table>
### 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. ipv6 enable  
5. autoroute announce  
6. destination ip-address  
7. path-option preference-priority dynamic segment-routing  
8. path-protection  
9. commit
## Configure Autoroute Announce

### 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>Configures the tunnel interface.</td>
</tr>
</tbody>
</table>
| **Step 2** interface tunnel-te *tunnel-id*  
Example:  
RP/0/RP0/CPU0:router(config)# interface tunnel-te22 | Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type. |
| **Step 3** ipv4 unnumbered type interface-path-id  
Example:  
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback0 | Enables IPv6 over the IPv4 tunnel. |
| **Step 4** ipv6 enable  
Example:  
RP/0/RP0/CPU0:router(config-if)# ipv6 enable | Enables messages that notify the neighbor nodes about the routes that are forwarding. |
| **Step 5** autoroute announce  
Example:  
RP/0/RP0/CPU0:router(config-if)# autoroute announce | Enables IPv6 over IPv4 tunnel is supported with CRS-FP-X next generation line cards. |
| **Step 6** destination *ip-address*  
Example:  
RP/0/RP0/CPU0:router(config-if)# destination 192.168.0.2 | Assigns a destination address on the new tunnel. |
| **Step 7** path-option *preference-priority* dynamic segment-routing  
Example:  
RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic segment-routing | Sets the path option to dynamic and assigns the path ID. |
| **Step 8** path-protection  
Example:  
RP/0/RP0/CPU0:router(config-if)# path-protection | Enables path protection on the tunnel-te interface. |
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. `ipv6 enable`
5. `autoroute destination destination-ip-address`
6. `destination ip-address`
7. `path-option preference-priority dynamic segment-routing`
8. `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 <code>interface tunnel-te tunnel-id</code></td>
<td>Configures the tunnel interface.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# interface tunnel-te22</td>
<td></td>
</tr>
<tr>
<td>Step 3 <code>ipv4 unnumbered type interface-path-id</code></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: RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback0</td>
<td></td>
</tr>
<tr>
<td>Step 4 <code>ipv6 enable</code></td>
<td>Enables IPv6 over the IPv4 tunnel.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-if)# ipv6 enable</td>
<td></td>
</tr>
<tr>
<td>Note IPv6 over IPv4 tunnel is supported with CRS-FP-X next generation line cards.</td>
<td></td>
</tr>
<tr>
<td>Step 5 <code>autoroute destination destination-ip-address</code></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>Example: RP/0/RP0/CPU0:router(config-if)# autoroute destination 192.168.0.1</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# autoroute</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.

### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>destination 192.168.0.2 (the default route)</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# autoroute</td>
<td></td>
</tr>
<tr>
<td>destination 192.168.0.3</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# autoroute</td>
<td></td>
</tr>
<tr>
<td>destination 192.168.0.4</td>
<td></td>
</tr>
</tbody>
</table>

### Using Binding Segments

#### Step 6

**destination ip-address**

**Example:**

RP/0/RP0/CPU0:router(config-if)# destination 192.168.0.2

Assigns a destination address on the new tunnel.

#### Step 7

**path-option preference-priority dynamic segment-routing**

**Example:**

RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic segment-routing

Sets the path option to dynamic and assigns the path ID.

#### Step 8

**commit**

Using Binding Segments
Stitching SR-TE Poles 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/CPU0:xrvr-5(config)# explicit-path name PATH5-9_10
RP/0/CPU0:xrvr-5(config-expl-path)# index 10 next-address strict ipv4 unicast 192.168.59.9
RP/0/CPU0:xrvr-5(config-expl-path)# index 20 next-address strict ipv4 unicast 10.1.1.10
RP/0/CPU0:xrvr-5(config-expl-path)# exit

RP/0/CPU0:xrvr-5(config)# interface tunnel-te1
RP/0/CPU0:xrvr-5(config-if)# ipv4 unnumbered Loopback0
RP/0/CPU0:xrvr-5(config-if)# destination 10.1.1.10
RP/0/CPU0:xrvr-5(config-if)# path-option 1 explicit name PATH5-9_10 segment-routing
RP/0/CPU0:xrvr-5(config-if)# commit
```

```
RP/0/CPU0:xrvr-5# show mpls traffic-eng tunnels 1 detail
Name: tunnel-te1 Destination: 10.1.1.10 Ifhandle:0x680
Signalled-Name: xrvr-5_t1
Status:
  Admin: up Oper: up Path: valid Signalling: connected
  path option 1, (Segment-Routing) type dynamic (Basis for Setup, path weight 10)
<...>
Binding SID: 24012
<...>
Segment-Routing Path Info (IS-IS 1 level-2)
  Segment0[Link]: 192.168.59.5 = 192.168.59.9, Label: 24007
  Segment1[Node]: 10.1.1.10, Label: 16010
```

**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:**

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

```
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
RP/0/0/CPU0:xrvr-1(config-expl-path)# index 20 next-label 24008
RP/0/0/CPU0:xrvr-1(config-expl-path)# exit
RP/0/0/CPU0:xrvr-1(config)# interface tunnel-te1
RP/0/0/CPU0:xrvr-1(config-if)# ipv4 unnumbered Loopback0
RP/0/0/CPU0:xrvr-1(config-if)# destination 10.1.1.10
RP/0/0/CPU0:xrvr-1(config-if)# path-option 1 explicit name PATH3_BSID segment-routing
RP/0/0/CPU0:xrvr-1(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:0x2f80
Signalled-Name: xrvr-3_t1
Status:
  Admin: up Oper: up Path: valid Signalling: connected
  path option 1, (Segment-Routing) type explicit PATH4_6_5 (Basis for Setup)
<...> Binding SID: 24008
<...>
  Segment-Routing Path Info (IS-IS 1 level-2)
  Segment0[Node]: 10.1.1.4, Label: 16004
  Segment1[Link]: 192.168.46.4 - 192.168.46.6, Label: 24003
  Segment2[Node]: 10.1.1.5, Label: 16005
  Segment3[ - ]: Label: 24012
```

The path is a chain of SR-TE policies stitched together using the binding-SIDs, providing a seamless end-to-end path.
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, node, and Shared Risk Link Groups (SRLG) 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.

TI-LFA node protection provides protection from node failures. The neighbor node is excluded during the post convergence backup path calculation.

Shared Risk Link Groups (SRLG) refer to situations in which links in a network share a common fiber (or a common physical attribute). These links have a shared risk: when one link fails, other links in the group might also fail. TI-LFA SRLG protection attempts to find the post-convergence backup path that excludes the SRLG of the protected link. All local links that share any SRLG with the protecting link are excluded.

When you enable link protection, you can also enable node protection, SRLG protection, or both, and specify a tiebreaker priority in case there are multiple LFAs.

For IS-IS, TI-LFA node protection and SRLG protection can be configured on the interface or the instance. For OSPF, TI-LFA node protection and SRLG protection are configured on the interface.

- Configuring TI-LFA for IS-IS, on page 43
- Configuring TI-LFA for OSPF, on page 45
- TI-LFA Node and SRLG Protection: Examples, on page 47

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, node, and SRLG 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 for IS-IS is configured. See Enabling Segment Routing for IS-IS Protocol, on page 9.
- Enter the following commands in global configuration mode:

```
Router(config)# ipv4 unnumbered mpls traffic-eng Loopback0
Router(config)# mpls traffic-eng
Router(config-mpls-te)# exit
Router(config)#
```

### SUMMARY STEPS

1. **configure**
2. **router isis instance-id**
3. **interface type interface-path-id**
4. **address-family ipv4 [unicast]**
5. **fast-reroute per-prefix**
6. **fast-reroute per-prefix ti-lfa**
7. **fast-reroute per-prefix tiebreaker {node-protecting | srlg-disjoint} index priority**

### 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></td>
</tr>
<tr>
<td>Step 2</td>
<td><strong>router isis instance-id</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config)# router isis 1</td>
<td>Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.&lt;br&gt;&lt;br&gt;<strong>Note</strong> You can change the level of routing to be performed by a particular routing instance by using the <strong>is-type</strong> router configuration command.</td>
</tr>
<tr>
<td>Step 3</td>
<td><strong>interface type interface-path-id</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config-isis))# interface GigabitEthernet0/0/2/1</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>Step 4</td>
<td><strong>address-family ipv4 [unicast]</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast</td>
<td>Specifies the IPv4 address family, and enters router address family configuration mode.</td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong></td>
<td>Enables per-prefix fast reroute.</td>
</tr>
<tr>
<td>fast-reroute per-prefix</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Enables per-prefix TI-LFA fast reroute link protection.</td>
</tr>
<tr>
<td>fast-reroute per-prefix ti-lfa</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix ti-lfa</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Enables TI-LFA node or SRLG protection and specifies the tiebreaker priority. Valid priority values are from 1 to 255. The lower the priority value, the higher the priority of the rule. Link protection always has a lower priority than node or SRLG protection.</td>
</tr>
<tr>
<td>fast-reroute per-prefix tiebreaker { node-protecting</td>
<td>srlg-disjoint} index priority</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix srlg-disjoint index 100</td>
<td></td>
</tr>
</tbody>
</table>

Note: The same attribute cannot be configured more than once on an interface.

---

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, node, and SRLG 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 for OSPF is configured. See Enabling Segment Routing for OSPF Protocol, on page 15.
- Enter the following commands in global configuration mode:

```
Router(config)# ipv4 unnumbered mpls traffic-eng Loopback0
Router(config)# mpls traffic-eng
```
Configure Topology-Independent Loop-Free Alternate (TI-LFA)

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
7. fast-reroute per-prefix tiebreaker { node-protecting | srlg-disjoint } index priority

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enables OSPF routing for the specified routing process, and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Step 2 router ospf process-name</td>
<td>Enables OSPF routing for the specified routing process, and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# router ospf 1</td>
<td></td>
</tr>
<tr>
<td>Step 3 area area-id</td>
<td>Enters area configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-ospf)# area 1</td>
<td></td>
</tr>
<tr>
<td>Step 4 interface type interface-path-id</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-ospf-ar)# interface GigabitEthernet0/0/2/1</td>
<td></td>
</tr>
<tr>
<td>Step 5 fast-reroute per-prefix</td>
<td>Enables per-prefix fast reroute.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-ospf-ar-if)# fast-reroute per-prefix</td>
<td></td>
</tr>
<tr>
<td>Step 6 fast-reroute per-prefix ti-lfa</td>
<td>Enables per-prefix TI-LFA fast reroute link protection.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-ospf-ar-if)# fast-reroute per-prefix ti-lfa</td>
<td></td>
</tr>
<tr>
<td>Step 7 fast-reroute per-prefix tiebreaker { node-protecting</td>
<td>srlg-disjoint } index priority</td>
</tr>
</tbody>
</table>
Configure Topology-Independent Loop-Free Alternate (TI-LFA)

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP/0/RP0/CPU0:router{config-isis-ar-if}# fast-reroute per-prefix srlg-disjoint index 100</td>
<td>rule. Link protection always has a lower priority than node or SRLG protection.</td>
</tr>
<tr>
<td></td>
<td>Note: The same attribute cannot be configured more than once on an interface.</td>
</tr>
</tbody>
</table>

TI-LFA has been successfully configured for segment routing.

**TI-LFA Node and SRLG Protection: Examples**

The following examples show the configuration of the tiebreaker priority for TI-LFA node and SRLG protection, and the behavior of post-convergence backup-path. These examples use OSPF, but the same configuration and behavior applies to IS-IS.

**Example: Enable link-protecting and node-protecting TI-LFA**

```
router ospf 1
area 1
  interface GigabitEthernet0/0/2/1
    fast-reroute per-prefix
    fast-reroute per-prefix ti-lfa
    fast-reroute per-prefix tiebreaker node-protecting index 100
```

Both link-protecting and node-protecting TI-LFA backup paths will be computed. If the priority associated with the node-protecting tiebreaker is higher than any other tiebreakers, then node-protecting post-convergence backup paths will be selected, if it is available.

**Example: Enable link-protecting and SRLG-protecting TI-LFA**

```
router ospf 1
area 1
  interface GigabitEthernet0/0/2/1
    fast-reroute per-prefix
    fast-reroute per-prefix ti-lfa
    fast-reroute per-prefix tiebreaker srlg-disjoint index 100
```

Both link-protecting and SRLG-protecting TI-LFA backup paths will be computed. If the priority associated with the SRLG-protecting tiebreaker is higher than any other tiebreakers, then SRLG-protecting post-convergence backup paths will be selected, if it is available.

**Example: Enable link-protecting, node-protecting and SRLG-protecting TI-LFA**

```
router ospf 1
area 1
  interface GigabitEthernet0/0/2/1
    fast-reroute per-prefix
    fast-reroute per-prefix ti-lfa
    fast-reroute per-prefix tiebreaker node-protecting index 100
    fast-reroute per-prefix tiebreaker srlg-disjoint index 200
```
Link-protecting, node-protecting, and SRLG-protecting TI-LFA backup paths will be computed. If the priority associated with the node-protecting tiebreaker is highest from all tiebreakers, then node-protecting post-convergence backup paths will be selected, if it is available. If the node-protecting backup path is not available, SRLG-protecting post-convergence backup path will be used, if it is available.
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 49
- Segment Routing and LDP Interoperability, on page 50
- Configuring Mapping Server, on page 52
- Enable Mapping Advertisement, on page 54
- Enable Mapping Client, on page 56

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>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 segment-routing</td>
<td>Enables segment routing.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# segment-routing</td>
</tr>
<tr>
<td>Step 3 mapping-server</td>
<td>Enables mapping server configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-sr)# mapping-server</td>
</tr>
<tr>
<td>Step 4 prefix-sid-map</td>
<td>Enables prefix-SID mapping configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-sr-ms)# prefix-sid-map</td>
</tr>
<tr>
<td></td>
<td>Note: Two-way prefix SID can be enabled directly under IS-IS or through a mapping server.</td>
</tr>
<tr>
<td>Step 5 address-family</td>
<td>Configures address-family for IS-IS.</td>
</tr>
<tr>
<td>ipv4</td>
<td>ipv6</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This example shows the address-family for ipv4:</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-sr-ms-map)# address-family ipv4</td>
</tr>
<tr>
<td></td>
<td>This example shows the address-family for ipv6:</td>
</tr>
</tbody>
</table>
Configure Segment Routing Mapping Server

### Purpose

Command or Action

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adds SID-mapping entries in the active local mapping policy. In the configured example:</td>
</tr>
<tr>
<td>- 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</td>
</tr>
<tr>
<td>- 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.</td>
</tr>
</tbody>
</table>

**Step 6**

<table>
<thead>
<tr>
<th>Command or Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip-address/prefix-length first-SID-value range range</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RP0/CPU0:router(config-sr-ms-map-af) #
10.1.1.1/32 10 range 200
RP/0/RP0/CPU0:router(config-sr-ms-map-af) #
20.1.0.0/16 400 range 300
```

**Step 7**

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

**Step 8**

show segment-routing mapping-server prefix-sid-map [ipv4 | ipv6] [detail]

**Example:**

```
RP/0/RP0/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

RP/0/RP0/CPU0:router# show segment-routing mapping-server prefix-sid-map ipv4 detail
Prefix
20.1.1.0/24
SID Index: 400
Range: 300
Last Prefix: 20.2.44.0/24
Last SID Index: 699
Flags:
10.1.1.1/32
SID Index: 10
Range: 200
Last Prefix: 10.1.1.200/32
Last SID Index: 209
Flags:
Number of mapping entries: 2
```

**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></td>
</tr>
<tr>
<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><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config)# router isis 1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>`address-family { ipv4</td>
<td>ipv6 } [ unicast ]`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td><code>segment-routing prefix-sid-map advertise-local</code></td>
<td>Configures IS-IS to advertise locally configured prefix-SID mappings.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/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></td>
</tr>
<tr>
<td><code>show isis database verbose</code></td>
<td>Displays IS-IS prefix-SID mapping advertisement and TLV.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router# show isis database verbose</code></td>
<td></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><strong>Step 1</strong></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>
</tbody>
</table>

  **Example:**
  
  RP/0/RP0/CPU0:router(config)# `router ospf 1`

| Step 2 | `segment-routing prefix-sid-map advertise-local` | Configures OSPF to advertise locally configured prefix-SID mappings. |

  **Example:**
  
  RP/0/RP0/CPU0:router(config-ospf)# `segment-routing prefix-sid-map advertise-local`

| Step 3 | `commit` | |

| Step 4 | `show ospf database opaque-area` | Displays OSPF prefix-SID mapping advertisement and TLV. |

  **Example:**
  
  RP/0/RP0/CPU0:router# `show ospf database opaque-area`

  `<...removed...>`

  Extended Prefix Range TLV: Length: 24
  
<table>
<thead>
<tr>
<th>AF</th>
<th>Prefix</th>
<th>Range Size</th>
<th>Flags</th>
<th>SID sub-TLV</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10.1.1.1/32</td>
<td>200</td>
<td>0x0</td>
<td>8</td>
</tr>
</tbody>
</table>
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/RP0/CPU0:router(config)# router isis 1
RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast
RP/0/RP0/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/RP0/CPU0:router(config)# router ospf 1
RP/0/RP0/CPU0:router(config-ospf)# segment-routing prefix-sid-map receive
```
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 57
- Traffic Collector Process, on page 57
- Configuring Traffic Collector, on page 58
- Displaying Traffic Information, on page 59

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>Step 1</td>
<td>configure</td>
</tr>
</tbody>
</table>
| Step 2 | traffic-collector  
Example:  
`RP/0/RP0/CPU0:router(config)# traffic-collector` | Enables traffic collector and places the router in traffic collector configuration mode. |
| Step 3 | `statistics collection-interval value`  
Example:  
`RP/0/RP0/CPU0:router(config-tc)# statistics collection-interval 5` | (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. |
| Step 4 | `statistics history-size value`  
Example:  
`RP/0/RP0/CPU0:router(config-tc)# statistics history-size value` | (Optional) Specifies the number of entries kept in the history database. Valid values are from 1 to 10. The default is 5. |
Purpose

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.

Note

(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.

Enter 0 to disable the history timeout. (No history is retained.)

Identifies interfaces that handle external traffic. Only L3 interfaces are supported for external traffic.

This completes the configuration for the traffic collector.

Displaying Traffic Information

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

For detailed information about the command syntax for the following show commands, see the Segment Routing Command Reference Guide.

- Display the configured external interfaces:

```
RP/0/RP0/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/RP0/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
```
Displays Traffic Information

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
of the Base and TM 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
23:12 - 23:13: Packets 9647265, Bytes: 9550792350
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 61
- Examples: LSP Ping and Traceroute for Nil_FEC Target, on page 62

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 1: 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/RP0/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 No ID Gi0/2/0/2 10.1.2.2 0
16005 No ID Gi0/2/0/0 10.1.1.4 0
16007 No ID Gi0/2/0/0 10.1.1.4 4752
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/RP0/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:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'L' - labeled output interface, 'B' - unlabeled output interface,
'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
'P' - no rx intf label prot, 'p' - premature termination of LSP,
'R' - transit router, 'I' - unknown upstream index,
'l' - Label switched with FEC change, 'd' - see DDMAP for return code,
'X' - unknown return code, 'x' - return code 0

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/RP0/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