



Segment Routing Configuration Guide for Cisco NCS 540 Series Routers, IOS XR Release 6.3.x

First Published: 2018-03-30

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

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 NON-INFRINGEMENT 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: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. 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)

© 2018 Cisco Systems, Inc. All rights reserved.



CONTENTS

PREFACE

Preface vii

Communications, Services, and Additional Information vii

CHAPTER 1

Scope 1

Need 2

Benefits 2

Workflow for Deploying Segment Routing 2

CHAPTER 2

About the Segment Routing Global Block 5

About the Segment Routing Local Block 7

Setup a Non-Default Segment Routing Global Block Range 8

Setup a Non-Default Segment Routing Local Block Range 9

CHAPTER 3

Enabling Segment Routing for IS-IS Protocol 13

Configuring a Prefix-SID on the IS-IS Enabled Loopback Interface 15

Configuring an Adjacency SID 18

Configuring Bandwidth-Based Local UCMP 21

IS-IS Multi-Domain Prefix SID and Domain Stitching: Example 22

Configure IS-IS Multi-Domain Prefix SID 22

Configure Common Router ID 23

Distribute IS-IS Link-State Data 24

CHAPTER 4

Enabling Segment Routing for OSPF Protocol 25

Configuring a Prefix-SID on the OSPF-Enabled Loopback Interface 27

CHAPTER 5

Segment Routing for BGP 31

Configure BGP Prefix Segment Identifiers 31

Segment Routing Egress Peer Engineering 32

 Configure Segment Routing Egress Peer Engineering 33

Configure BGP Link-State 34

Example: Configuring SR-EPE and BGP-LS 35

CHAPTER 6

SR-TE Policy Overview 37

Auto-Route Announce for SR-TE 37

Instantiation of an SR Policy 38

 Manually Provisioned SR Policy 38

SR-TE Policy Path Types 38

 Dynamic Paths 38

 Optimization Objectives 38

 Constraints 40

 Configure SR Policy with Dynamic Path 41

 Explicit Paths 42

 Configure SR-TE Policy with Explicit Path 42

 Configuring Explicit Path with Affinity Constraint Validation 45

Protocols 47

 Path Computation Element Protocol 47

 BGP SR-TE 47

 Configure Explicit BGP SR-TE 48

Traffic Steering 50

 Automated Steering 50

 Using Binding Segments 51

 Stitching SR-TE Policies Using Binding SID: Example 52

 L2VPN Preferred Path 55

CHAPTER 7

About SR-PCE 57

Configure SR-PCE 57

Configure the Disjoint Policy (Optional) 59

CHAPTER 8

Limitations 63

Usage Guidelines and Limitations 63

Configuring TI-LFA for IS-IS	64
Configuring TI-LFA for OSPF	66
TI-LFA Node and SRLG Protection: Examples	67

CHAPTER 9	About Segment Routing Microloop Avoidance	69
	Segment Routing Microloop Avoidance Limitations	69
	Configure Segment Routing Microloop Avoidance for IS-IS	69

CHAPTER 10	Segment Routing Mapping Server	71
	Usage Guidelines and Restrictions	71
	Segment Routing and LDP Interoperability	72
	Example: Segment Routing LDP Interoperability	72
	Configuring Mapping Server	74
	Enable Mapping Advertisement	76
	Configure Mapping Advertisement for IS-IS	76
	Configure Mapping Advertisement for OSPF	77
	Enable Mapping Client	78

CHAPTER 11	Traffic Collector Process	79
	Configuring Traffic Collector	80
	Displaying Traffic Information	81



Preface



Note This release has reached end-of-life status. For more information, see the [End-of-Life and End-of-Sale Notices](#).

The *Segment Routing Configuration Guide for* 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.



CHAPTER 1

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. An adjacency SID can be allocated dynamically from the dynamic label range or configured manually from the segment routing local block (SRLB) range of labels. 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 policy 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 policy.

- [Need, on page 2](#)
- [Benefits, on page 2](#)
- [Workflow for Deploying Segment Routing, on page 2](#)

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 policies 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 for the IGP

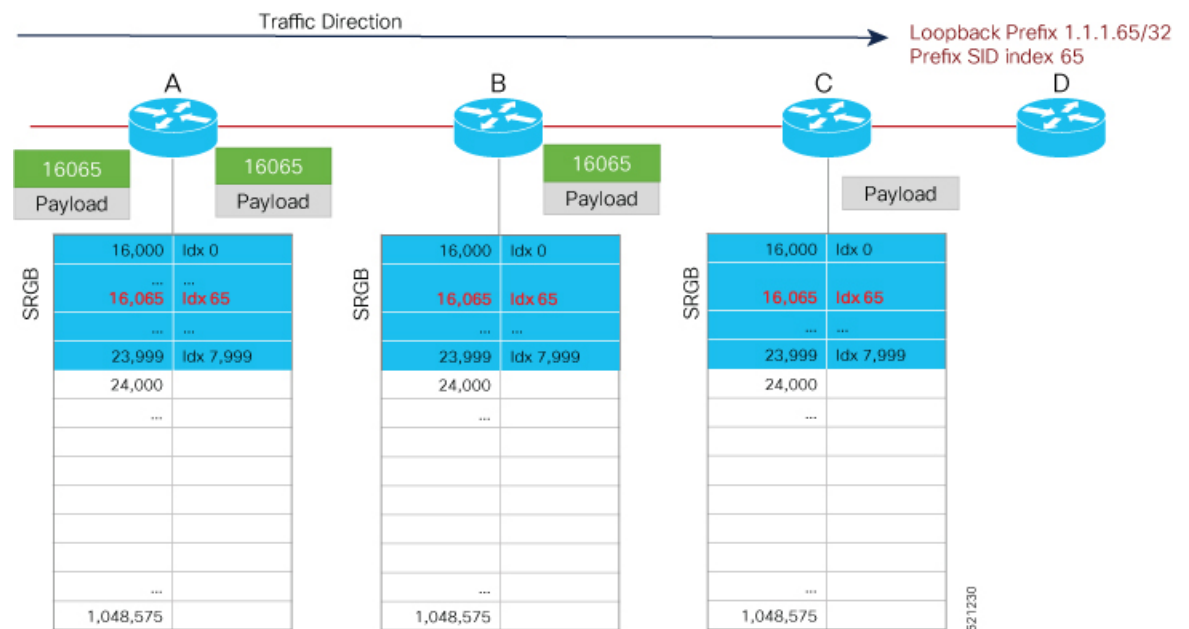
3. Configure the SR-TE Policy
4. Configure the Segment Routing Mapping Server



CHAPTER 2

About the Segment Routing Global Block

The Segment Routing Global Block (SRGB) is a range of labels reserved for Segment Routing global segments. A prefix-SID is advertised as a domain-wide unique index. The prefix-SID index points to a unique label within the SRGB range. The index is zero-based, meaning that the first index is 0. The MPLS label assigned to a prefix is derived from the Prefix-SID index plus the SRGB base. For example, considering an SRGB range of 16,000 to 23,999, a prefix 1.1.1.65/32 with prefix-SID index of **65** is assigned the label value of **16065**.



To keep the configuration simple and straightforward, we strongly recommended that you use a homogenous SRGB (meaning, the same SRGB range across all nodes). Using a heterogenous SRGB (meaning, a different SRGB range of the same size across nodes) is also supported but is not recommended.

Behaviors and Limitations

- The default SRGB in IOS XR has a size of 8000 starting from label value 16000. The default range is 16000 to 23,999. With this size, and assuming one loopback prefix per router, an operator can assign prefix SIDs to a network with 8000 routers.
- There are instances when you might need to define a different SRGB range. For example:

- Non-IOS XR nodes with a SRGB range that is different than the default IOS XR SRGB range.
- The default SRGB range is not large enough to accommodate all required prefix SIDs.
- A non-default SRGB can be configured following these guidelines:
 - The SRGB starting value can be configured anywhere in the dynamic label range space (16,000 to 1,048,575).
 - In Cisco IOS XR release earlier than 6.6.3, the SRGB can have a maximum configurable size of 262,143.
 - In Cisco IOS XR release 6.6.3 and later, the SRGB can be configured to any size value that fits within the dynamic label range space.
- Allocating an SRGB label range does not mean that all the labels in this range are programmed in the forwarding table. The label range is just reserved for SR and not available for other purposes. Furthermore, a platform may limit the number of local labels that can be programmed.
- We recommend that the non-default SRGB be configured under the **segment-routing** global configuration mode. By default, all IGP instances and BGP use this SRGB.
- You can also configure a non-default SRGB under the IGP, but it is not recommended.

SRGB Label Conflicts

When you define a non-default SRGB range, there might be a label conflict (for example, if labels are already allocated, statically or dynamically, in the new SRGB range). The following system log message indicates a label conflict:

```
%ROUTING-ISIS-4-SRGB_ALLOC_FAIL : SRGB allocation failed: 'SRGB reservation not
successful for [16000,80000], SRGB (16000 80000, SRGB_ALLOC_CONFIG_PENDING, 0x2)
(So far 16 attempts). Make sure label range is free'
```

To remove this conflict, you must reload the router to release the currently allocated labels and to allocate the new SRGB.

After the system reloads, LSD does not accept any dynamic label allocation before IS-IS/OSPF/BGP have registered with LSD. Upon IS-IS/OSPF/BGP registration, LSD allocates the requested SRGB (either the default range or the customized range).

After IS-IS/OSPF/BGP have registered and their SRGB is allocated, LSD starts serving dynamic label requests from other clients.



Note To avoid a potential router reload due to label conflicts, and assuming that the default SRGB size is large enough, we recommend that you use the default IOS XR SRGB range.



Note Allocating a non-default SRGB in the upper part of the MPLS label space increases the chance that the labels are available and a reload can be avoided.



Caution Modifying a SRGB configuration is disruptive for traffic and may require a reboot if the new SRGB is not available entirely.

- [About the Segment Routing Local Block, on page 7](#)
- [Setup a Non-Default Segment Routing Global Block Range, on page 8](#)
- [Setup a Non-Default Segment Routing Local Block Range, on page 9](#)

About the Segment Routing Local Block

A local segment is automatically assigned an MPLS label from the dynamic label range. In most cases, such as TI-LFA backup paths and SR-TE explicit paths defined with IP addresses, this dynamic label allocation is sufficient. However, in some scenarios, it could be beneficial to allocate manually local segment label values to maintain label persistency. For example, an SR-TE policy with a manual binding SID that is performing traffic steering based on incoming label traffic with the binding SID.

The Segment Routing Local Block (SRLB) is a range of label values preserved for the manual allocation of local segments, such as adjacency segment identifiers (adj-SIDs). These labels are locally significant and are only valid on the nodes that allocate the labels.

Behaviors and Limitations

- The default SRLB has a size of 1000 starting from label value 15000; therefore, the default SRLB range goes from 15000 to 15,999.
- A non-default SRLB can be configured following these guidelines:
 - The SRLB starting value can be configured anywhere in the dynamic label range space (16,000 to 1,048,575).
 - In Cisco IOS XR release earlier than 6.6.3, the SRLB can have a maximum configurable size of 262,143.
 - In Cisco IOS XR release 6.6.3 and later, the SRLB can be configured to any size value that fits within the dynamic label range space.

SRLB Label Conflicts

When you define a non-default SRLB range, there might be a label conflict (for example, if labels are already allocated, statically or dynamically, in the new SRLB range). In this case, the new SRLB range will be accepted, but not applied (pending state). The previous SRLB range (active) will continue to be in use.

To remove this conflict, you must reload the router to release the currently allocated labels and to allocate the new SRLB.



Caution You can use the **clear segment-routing local-block discrepancy all** command to clear label conflicts. However, using this command is disruptive for traffic since it forces all other MPLS applications with conflicting labels to allocate new labels.



Note To avoid a potential router reload due to label conflicts, and assuming that the default SRGB size is large enough, we recommend that you use the default IOS XR SRLB range.



Note Allocating a non-default SRLB in the upper part of the MPLS label space increases the chance that the labels are available and a reload can be avoided.

Setup a Non-Default Segment Routing Global Block Range

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

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	segment-routing global-block <i>starting_value</i> <i>ending_value</i> Example: RP/0/RP0/CPU0:router(config)# segment-routing global-block 16000 80000	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.
Step 3	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Use the **show mpls label table** [*label label-value*] command to verify the SRGB configuration:


```
Router# show mpls label table label 16000 detail
Table Label   Owner                               State Rewrite
-----
0      16000   ISIS(A):1                               InUse  No
      (lbl-blk SRGB, vers:0, (start_label=16000, size=64001)
```

What to do next

Configure prefix SIDs and enable segment routing.

Setup a Non-Default Segment Routing Local Block Range

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

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	segment-routing local-block <i>starting_value</i> <i>ending_value</i> Example: RP/0/RP0/CPU0:router(config)# segment-routing local-block 30000 30999	Enter the lowest value that you want the SRLB range to include as the starting value. Enter the highest value that you want the SRLB range to include as the ending value.
Step 3	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Use the **show mpls label table** [*label label-value*] [**detail**] command to verify the SRLB configuration:

```
Router# show mpls label table label 30000 detail
```

```

Table Label   Owner                               State Rewrite
-----
0      30000   LSD(A)                               InUse  No
(Lbl-blk SRLB, vers:0, (start_label=30000, size=1000, app_notify=0)

```

```
Router# show segment-routing local-block inconsistencies
```

```
No inconsistencies
```

The following example shows an SRLB label conflict in the range of 30000 and 30999. Note that the default SRLB is active and the configured SRLB is pending:

```
Router(config)# segment-routing local-block 30000 30999
```

```
%ROUTING-MPLS_LSD-3-ERR_SRLB_RANGE : SRLB allocation failed: 'SRLB reservation not successful
for [30000,30999]. Use with caution 'clear segment-routing local-block discrepancy all'
command
to force srlb allocation'
```



Caution

You can use the **clear segment-routing local-block discrepancy all** command to clear label conflicts. However, using this command is disruptive for traffic since it forces all other MPLS applications with conflicting labels to allocate new labels.

```
Router# show mpls label table label 30000 detail
```

```

Table Label   Owner                               State Rewrite
-----
0      30000   LSD(A)                               InUse  No
(Lbl-blk SRLB, vers:0, (start_label=30000, size=1000, app_notify=0)

```

```
Router# show segment-routing local-block inconsistencies
```

```
SRLB inconsistencies range: Start/End: 30000/30999
```

```
Router# show mpls lsd private | i SRLB
```

```

SRLB Lbl Mgr:
  Current Active SRLB block      = [15000, 15999]
  Configured Pending SRLB block = [30000, 30999]

```

Reload the router to release the currently allocated labels and to allocate the new SRLB:

```
Router# reload
```

```
Proceed with reload? [confirm]yes
```

After the system is brought back up, verify that there are no label conflicts with the SRLB configuration:

```
Router# show mpls lsd private | i SRLB
```

```

SRLB Lbl Mgr:
  Current Active SRLB block      = [30000, 30999]
  Configured Pending SRLB block = [0, 0]

```

```
Router# show segment-routing local-block inconsistencies
```

No inconsistencies

What to do next

Configure adjacency SIDs and enable segment routing.

Setup a Non-Default Segment Routing Local Block Range



CHAPTER 3

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.



Note You must enter the commands in the following task list on every IS-IS router in the traffic-engineered portion of your network.

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# <code>configure</code>	Enters mode.
Step 2	router isis <i>instance-id</i> Example: RP/0/RP0/CPU0:router(config)# <code>router isis isp</code>	Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.

	Command or Action	Purpose
		<p>Note You can change the level of routing to be performed by a particular routing instance by using the is-type router configuration command.</p>
Step 3	<p>address-family { ipv4 ipv6 } [unicast]</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast</pre>	Specifies the IPv4 or IPv6 address family, and enters router address family configuration mode.
Step 4	<p>metric-style wide [level { 1 2 }]</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-isis-af)# metric-style wide level 1</pre>	Configures a router to generate and accept only wide link metrics in the Level 1 area.
Step 5	<p>mpls traffic-eng level</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-isis-af)# mpls traffic-eng level-2-only</pre>	Enables RSVP traffic engineering functionality.
Step 6	<p>mpls traffic-eng router-id interface</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-isis-af)# mpls traffic-eng router-id Loopback0</pre>	Sets the traffic engineering loopback interface.
Step 7	<p>router-id loopback loopback interface used for prefix-sid</p> <p>Example:</p> <pre>RP/0/RP0(config-isis-af)#router-id loopback0</pre>	Configures router ID for each address-family (ipv4/ipv6).
Step 8	<p>segment-routing mpls</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-isis-af)# segment-routing mpls</pre>	<p>Segment routing is enabled by the following actions:</p> <ul style="list-style-type: none"> • MPLS forwarding is enabled on all interfaces where IS-IS is active. • All known prefix-SIDs in the forwarding plain are programmed, with the prefix-SIDs advertised by remote routers or learned through local or remote mapping server. • The prefix-SIDs locally configured are advertised.

	Command or Action	Purpose
Step 9	exit Example: <pre>RP/0/RP0/CPU0:router(config-isis-af)# exit RP/0/RP0/CPU0:router(config-isis)# exit</pre>	
Step 10	mpls traffic-eng Example: <pre>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</pre>	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.
Step 11	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

What to do next

Configure the prefix SID.

- [Configuring a Prefix-SID on the IS-IS Enabled Loopback Interface, on page 15](#)
- [Configuring an Adjacency SID, on page 18](#)
- [Configuring Bandwidth-Based Local UCMP, on page 21](#)
- [IS-IS Multi-Domain Prefix SID and Domain Stitching: Example, on page 22](#)

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 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.

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	router isis <i>instance-id</i> Example: RP/0/RP0/CPU0:router(config)# router isis 1	Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode. • You can change the level of routing to be performed by a particular routing instance by using the is-type router configuration command.
Step 3	interface Loopback <i>instance</i> Example: RP/0/RP0/CPU0:router(config-isis)# interface Loopback0	Specifies the loopback interface and instance.
Step 4	address-family { ipv4 ipv6 } [unicast] Example: The following is an example for ipv4 address family: RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast	Specifies the IPv4 or IPv6 address family, and enters router address family configuration mode.
Step 5	prefix-sid {index <i>SID-index</i> absolute <i>SID-value</i>} [n-flag-clear] [explicit-null] Example: RP/0/RP0/CPU0:router(config-isis-if-af)#	Configures the prefix-SID index or absolute value for the interface. Specify index <i>SID-index</i> for each node to create a prefix SID based on the lower boundary of the SRGB + the index.

	Command or Action	Purpose
	<pre>prefix-sid index 1001 RP/0/RP0/CPU0:router(config-isis-if-af)# prefix-sid absolute 17001</pre>	<p>Specify absolute <i>SID-value</i> for each node to create a specific prefix SID within the SRGB.</p> <p>By default, the <code>n-flag</code> 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 <code>N</code> flag in the prefix-SID sub Type Length Value (TLV).</p> <p>To disable penultimate-hop-popping (PHP) and add explicit-Null label, enter <code>explicit-null</code> keyword. IS-IS sets the <code>E</code> flag in the prefix-SID sub TLV.</p>
Step 6	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

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
  NLPID:           0xcc
  NLPID:           0x8e
  MT:              Standard (IPv4 Unicast)
  MT:              IPv6 Unicast                               0/0/0
  Hostname:        router
  IP Address:      10.0.0.1
  IPv6 Address:    2001:0db8:1234::0a00:0001
  Router Cap:      10.0.0.1, D:0, S:0
    Segment Routing: I:1 V:1, SRGB Base: 16000 Range: 8000
    SR Algorithm:
      Algorithm: 0
<...>
Metric: 0           IP-Extended 10.0.0.1/32
  Prefix-SID Index: 1001, Algorithm:0, R:0 N:1 P:0 E:0 V:0 L:0
<...>
```

What to do next

Configure the SR-TE policy.

Configuring an Adjacency SID

An adjacency SID (Adj-SID) is associated with an adjacency to a neighboring node. The adjacency SID steers the traffic to a specific adjacency. Adjacency SIDs have local significance and are only valid on the node that allocates them.

An adjacency SID can be allocated dynamically from the dynamic label range or configured manually from the segment routing local block (SRLB) range of labels.

Adjacency SIDs that are dynamically allocated do not require any special configuration, however there are some limitations:

- A dynamically allocated Adj-SID value is not known until it has been allocated, and a controller will not know the Adj-SID value until the information is flooded by the IGP.
- Dynamically allocated Adj-SIDs are not persistent and can be reallocated after a reload or a process restart.
- Each link is allocated a unique Adj-SID, so the same Adj-SID cannot be shared by multiple links.

Manually allocated Adj-SIDs are persistent over reloads and restarts. They can be provisioned for multiple adjacencies to the same neighbor or to different neighbors. You can specify that the Adj-SID is protected. If the Adj-SID is protected on the primary interface and a backup path is available, a backup path is installed. By default, manual Adj-SIDs are not protected.

Adjacency SIDs are advertised using the existing IS-IS Adj-SID sub-TLV. The S and P flags are defined for manually allocated Adj-SIDs.

```

 0 1 2 3 4 5 6 7
+---+---+---+---+---+---+
|F|B|V|L|S|P|   |
+---+---+---+---+---+---+

```

Table 1: Adjacency Segment Identifier (Adj-SID) Flags Sub-TLV Fields

Field	Description
S (Set)	This flag is set if the same Adj-SID value has been provisioned on multiple interfaces.
P (Persistent)	This flag is set if the Adj-SID is persistent (manually allocated).

Manually allocated Adj-SIDs are supported on point-to-point (P2P) interfaces.

This task explains how to configure an Adj-SID on an interface.

Before you begin

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

Use the **show mpls label table detail** command to verify the SRLB range.

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	router isis instance-id Example: RP/0/RP0/CPU0:router(config)# router isis 1	Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode. • You can change the level of routing to be performed by a particular routing instance by using the is-type router configuration command.
Step 3	interface type interface-path-id Example: RP/0/RP0/CPU0:router(config-isis)# interface GigabitEthernet0/0/0/7	Specifies the interface and enters interface configuration mode.
Step 4	point-to-point Example: RP/0/RP0/CPU0:router(config-isis-if)# point-to-point	Specifies the interface is a point-to-point interface.
Step 5	address-family { ipv4 ipv6 } [unicast] Example: The following is an example for ipv4 address family: RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast	Specifies the IPv4 or IPv6 address family, and enters router address family configuration mode.
Step 6	adjacency-sid {index adj-SID-index absolute adj-SID-value } [protected] Example: RP/0/RP0/CPU0:router(config-isis-if-af)# adjacency-sid index 10	Configures the Adj-SID index or absolute value for the interface. Specify index adj-SID-index for each link to create an Adj-SID based on the lower boundary of the SRLB + the index. Specify absolute adj-SID-value for each link to create a specific Adj-SID within the SRLB.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router(config-isis-if-af)# adjacency-sid absolute 15010	Specify if the Adj-SID is protected . For each primary path, if the Adj-SID is protected on the primary interface and a backup path is available, a backup path is installed. By default, manual Adj-SIDs are not protected.
Step 7	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Verify the Adj-SID configuration:

```
RP/0/RP0/CPU0:router# show isis segment-routing label adjacency persistent
Mon Jun 12 02:44:07.085 PDT

IS-IS 1 Manual Adjacency SID Table

15010 AF IPv4
  GigabitEthernet0/0/0/3: IPv4, Protected 1/65/N, Active
  GigabitEthernet0/0/0/7: IPv4, Protected 2/66/N, Active

15100 AF IPv6
  GigabitEthernet0/0/0/3: IPv6, Not protected 255/255/N, Active
```

Verify the labels are added to the MPLS Forwarding Information Base (LFIB):

```
RP/0/RP0/CPU0:router# show mpls forwarding labels 15010
Mon Jun 12 02:50:12.172 PDT

Local  Outgoing  Prefix      Outgoing   Next Hop    Bytes
Label  Label        or ID      Interface  Hop         Switched
-----
15010  Pop          SRLB (idx 10)  Gi0/0/0/3  10.0.3.3    0
      Pop          SRLB (idx 10)  Gi0/0/0/7  10.1.0.5    0
      16004        SRLB (idx 10)  Gi0/0/0/7  10.1.0.5    0                (!)
      16004        SRLB (idx 10)  Gi0/0/0/3  10.0.3.3    0                (!)
```

What to do next

Configure the SR-TE policy.

Configuring Bandwidth-Based Local UCMP

Bandwidth-based local Unequal Cost Multipath (UCMP) allows you to enable UCMP functionality locally between Equal Cost Multipath (ECMP) paths based on the bandwidth of the local links.

Bandwidth-based local UCMP is performed for prefixes, segment routing Adjacency SIDs, and Segment Routing label cross-connects installed by IS-IS, and is supported on any physical or virtual interface that has a valid bandwidth.

For example, if the capacity of a bundle interface changes due to the link or line card up/down event, traffic continues to use the affected bundle interface regardless of the available provisioned bundle members. If some bundle members were not available due to the failure, this behavior could cause the traffic to overload the bundle interface. To address the bundle capacity changes, bandwidth-based local UCMP uses the bandwidth of the local links to load balance traffic when bundle capacity changes.

Before you begin

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# <code>configure</code>	Enters mode.
Step 2	router isis <i>instance-id</i> Example: RP/0/RP0/CPU0:router(config)# <code>router isis 1</code>	Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode. 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: The following is an example for ipv4 address family: RP/0/RP0/CPU0:router(config-isis)# <code>address-family ipv4 unicast</code>	Specifies the IPv4 or IPv6 address family, and enters IS-IS address family configuration mode.
Step 4	apply-weight ecmp-only bandwidth Example: RP/0/RP0/CPU0:router(config-isis-af)# <code>apply-weight ecmp-only bandwidth</code>	Enables UCMP functionality locally between ECMP paths based on the bandwidth of the local links.
Step 5	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session.

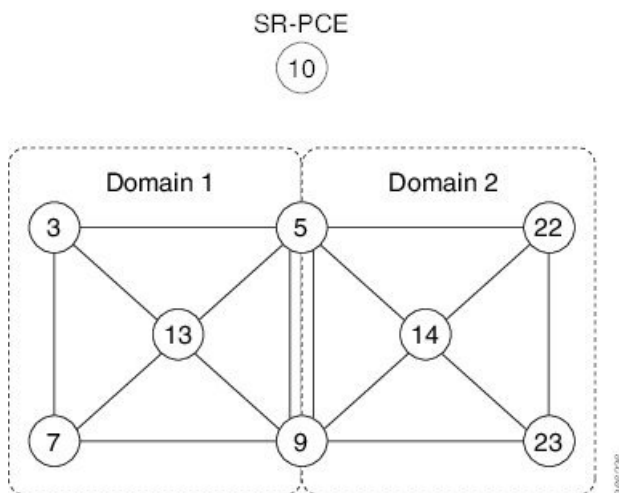
	Command or Action	Purpose
		<p>end—Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

IS-IS Multi-Domain Prefix SID and Domain Stitching: Example

IS-IS Multi-Domain Prefix SID and Domain Stitching allows you to configure multiple IS-IS instances on the same loopback interface for domain border nodes. You specify a loopback interface and prefix SID under multiple IS-IS instances to make the prefix and prefix SID reachable in different domains.

This example uses the following topology. Node 5 and 9 are border nodes between two IS-IS domains (Domain1 and Domain2). Node 10 is configured as the Segment Routing Path Computation Element (SR-PCE).

Figure 1: Multi-Domain Topology



Configure IS-IS Multi-Domain Prefix SID

Specify a loopback interface and prefix SID under multiple IS-IS instances on each border node:

```

Example: Border Node 5
router isis Domain1
 interface Loopback0
  address-family ipv4 unicast
  prefix-sid absolute 16005

```

```
router isis Domain2
 interface Loopback0
  address-family ipv4 unicast
  prefix-sid absolute 16005
```

Example: Border Node 9

```
router isis Domain1
 interface Loopback0
  address-family ipv4 unicast
  prefix-sid absolute 16009
```

```
router isis Domain2
 interface Loopback0
  address-family ipv4 unicast
  prefix-sid absolute 16009
```

Border nodes 5 and 9 each run two IS-IS instances (Domain1 and Domain2) and advertise their Loopback0 prefix and prefix SID in both domains.

Nodes in both domains can reach the border nodes by using the same prefix and prefix SID. For example, Node 3 and Node 22 can reach Node 5 using prefix SID 16005.

Configure Common Router ID

On each border node, configure a common TE router ID under each IS-IS instance:

Example: Border Node 5

```
router isis Domain1
 address-family ipv4 unicast
  router-id loopback0
```

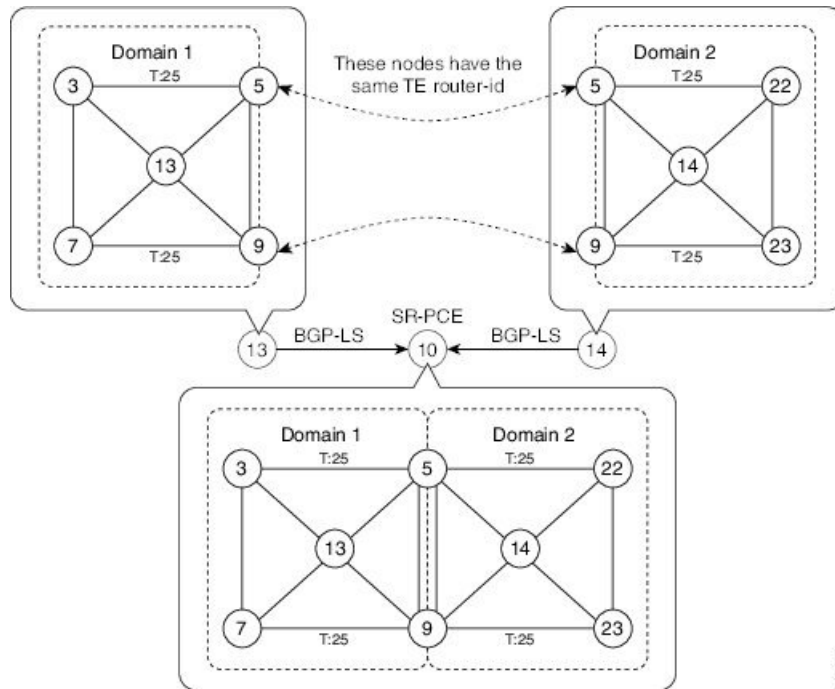
```
router isis Domain2
 address-family ipv4 unicast
  router-id loopback0
```

Example: Border Node 9

```
router isis Domain1
 address-family ipv4 unicast
  router-id loopback0
```

```
router isis Domain2
 address-family ipv4 unicast
  router-id loopback0
```

Distribute IS-IS Link-State Data



Configure BGP Link-state (BGP-LS) on Node 13 and Node 14 to report their local domain to Node 10:

Example: Node 13

```
router isis Domain1
  distribute link-state instance-id instance-id
```

Example: Node 14

```
router isis Domain2
  distribute link-state instance-id instance-id
```

Link-state ID starts from 32. One ID is required per IGP domain. Different domain IDs are essential to identify that the SR-TE TED belongs to a particular IGP domain.

Nodes 13 and 14 each reports its local domain in BGP-LS to Node 10.

Node 10 identifies the border nodes (Nodes 5 and 9) by their common advertised TE router ID, then combines (stitches) the domains on these border nodes for end-to-end path computations.



CHAPTER 4

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.



Note You must enter the commands in the following task list on every OSPF router in the traffic-engineered portion of your network.

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# <code>configure</code>	Enters mode.
Step 2	router ospf <i>process-name</i> Example: RP/0/RP0/CPU0:router(config)# <code>router ospf 1</code>	Enables OSPF routing for the specified routing process and places the router in router configuration mode.

	Command or Action	Purpose
Step 3	mpls traffic-eng router-id <i>interface</i> Example: RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng router-id Loopback0	Sets the traffic engineering loopback interface.
Step 4	segment-routing mpls Example: RP/0/RP0/CPU0:router(config-ospf)# segment-routing mpls	<p>Enables segment routing using the MPLS data plane on the routing process and all areas and interfaces in the routing process.</p> <p>Enables segment routing forwarding on all interfaces in the routing process and installs the SIDs received by OSPF in the forwarding table.</p>
Step 5	area <i>area</i> Example: RP/0/RP0/CPU0:router(config-ospf)# area 0	Enters area configuration mode.
Step 6	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config-ospf-ar)# mpls traffic-eng	Enables IGP traffic engineering functionality.
Step 7	segment-routing mpls Example: RP/0/RP0/CPU0:router(config-ospf-ar)# segment-routing mpls	(Optional) Enables segment routing using the MPLS data plane on the area and all interfaces in the area. Enables segment routing forwarding on all interfaces in the area and installs the SIDs received by OSPF in the forwarding table.
Step 8	exit Example: RP/0/RP0/CPU0:router(config-ospf-ar)# exit RP/0/RP0/CPU0:router(config-ospf)# exit	
Step 9	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng	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.

	Command or Action	Purpose
Step 10	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

What to do next

Configure the prefix SID.

- [Configuring a Prefix-SID on the OSPF-Enabled Loopback Interface, on page 27](#)

Configuring a Prefix-SID on the OSPF-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 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.

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	router ospf <i>process-name</i> Example: RP/0/RP0/CPU0:router (config)# router ospf 1	Enables OSPF routing for the specified routing process, and places the router in router configuration mode.
Step 3	area <i>value</i> Example: RP/0/RP0/CPU0:router (config-ospf)# area 0	Enters area configuration mode.
Step 4	interface Loopback <i>interface-instance</i> Example: RP/0/RP0/CPU0:router (config-ospf-ar)# interface loopback 0	Specifies the loopback interface and instance.
Step 5	prefix-sid {<i>index SID-index</i> absolute <i>SID-value</i> } [n-flag-clear] [explicit-null] Example: RP/0/RP0/CPU0:router (config-ospf-ar)# prefix-sid index 1001 RP/0/RP0/CPU0:router (config-ospf-ar)# prefix-sid absolute 17001	<p>Configures the prefix-SID index or absolute value for the interface.</p> <p>Specify index <i>SID-index</i> for each node to create a prefix SID based on the lower boundary of the SRGB + the index.</p> <p>Specify absolute <i>SID-value</i> for each node to create a specific prefix SID within the SRGB.</p> <p>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. OSPF does not set the N flag in the prefix-SID sub Type Length Value (TLV).</p> <p>To disable penultimate-hop-popping (PHP) and add an explicit-Null label, enter the <code>explicit-null</code> keyword. OSPF sets the E flag in the prefix-SID sub TLV.</p>
Step 6	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Verify the prefix-SID configuration:

```
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
```




CHAPTER 5

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, on page 31](#)
- [Segment Routing Egress Peer Engineering, on page 32](#)
- [Configure BGP Link-State, on page 34](#)
- [Example: Configuring SR-EPE and BGP-LS, on page 35](#)

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 *Routing Configuration Guide for Cisco NCS 540 Series Routers*.

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/RP0/CPU0:router(config)# segment-routing global-block 16000 23999

RP/0/RP0/CPU0:router(config)# route-policy SID($SID)
RP/0/RP0/CPU0:router(config-rpl)# set label-index $SID
RP/0/RP0/CPU0:router(config-rpl)# end policy

RP/0/RP0/CPU0:router(config)# router bgp 1
RP/0/RP0/CPU0:router(config-bgp)# bgp router-id 1.1.1.1
RP/0/RP0/CPU0:router(config-bgp)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-af)# network 1.1.1.3/32 route-policy SID(3)
RP/0/RP0/CPU0:router(config-bgp-af)# allocate-label all
RP/0/RP0/CPU0:router(config-bgp-af)# commit
RP/0/RP0/CPU0:router(config-bgp-af)# end

RP/0/RP0/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
  Path #1: Received by speaker 0
  Advertised to update-groups (with more than one peer):
    0.2
  3
  99.3.21.3 from 99.3.21.3 (1.1.1.3)
  Received Label 3
  Origin IGP, metric 0, localpref 100, valid, external, best, group-best
  Received Path ID 0, Local Path ID 1, version 74
  Origin-AS validity: not-found
  Label Index: 3
```

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.

Below are the BGP-EPE peering SID types:

- PeerNode SID—To an eBGP peer. Pops the label and forwards the traffic on any interface to the peer.
- PeerAdjacency SID—To an eBGP peer via interface. Pops the label and forwards the traffic on the related interface.

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.

Configure Segment Routing Egress Peer Engineering

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

Procedure

	Command or Action	Purpose
Step 1	<p>router bgp <i>as-number</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# router bgp 1</pre>	Specifies the BGP AS number and enters the BGP configuration mode, allowing you to configure the BGP routing process.
Step 2	<p>neighbor <i>ip-address</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-bgp)# neighbor 192.168.1.3</pre>	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.
Step 3	<p>remote-as <i>as-number</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 3</pre>	Creates a neighbor and assigns a remote autonomous system number to it.
Step 4	<p>egress-engineering</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-bgp-nbr)# egress-engineering</pre>	Configures the egress node with EPE for the eBGP peer.

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 IGP data does 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/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# router bgp 1
RP/0/RP0/CPU0:router(config-bgp)# neighbor 10.0.0.2
RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 1
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family link-state link-state
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# exit
```

IGP Link-State Database Distribution

A given BGP node may have connections to multiple, independent routing domains. IGP link-state database distribution into BGP-LS is supported for both OSPF and IS-IS protocols in order to distribute this information on to controllers or applications that desire to build paths spanning or including these multiple domains.

To distribute IS-IS link-state data using BGP LS, use the **distribute link-state** command in router configuration mode.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# router isis isp
RP/0/RP0/CPU0:router(config-isis)# distribute link-state instance-id 32 level 2 throttle 5
```

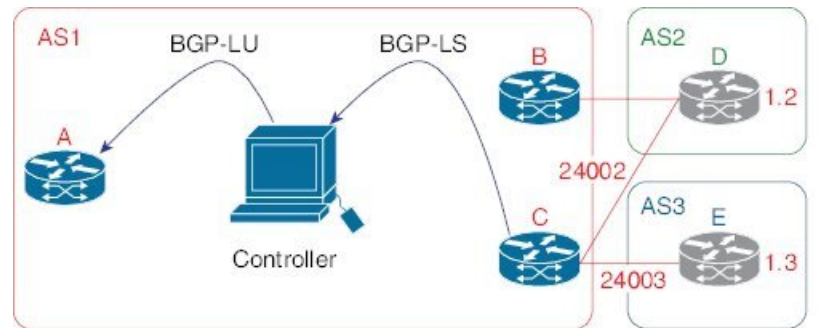
To distribute OSPFv2 link-state data using BGP LS, use the **distribute link-state** command in router configuration mode.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# router ospf 100
RP/0/RP0/CPU0:router(config-ospf)# distribute link-state instance-id 32 throttle 10
```

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 2: Topology



Procedure

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

Example:

```
RP/0/RP0/CPU0:router_C(config)# router bgp 1
RP/0/RP0/CPU0:router_C(config-bgp)# neighbor 192.168.1.3
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# remote-as 3
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# description to E
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# egress-engineering
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:router_C(config-bgp-nbr-af)# route-policy bgp_in in
RP/0/RP0/CPU0:router_C(config-bgp-nbr-af)# route-policy bgp_out out
RP/0/RP0/CPU0:router_C(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# exit
RP/0/RP0/CPU0:router_C(config-bgp)# neighbor 192.168.1.2
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# remote-as 2
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# description to D
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# egress-engineering
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:router_C(config-bgp-nbr-af)# route-policy bgp_in in
RP/0/RP0/CPU0:router_C(config-bgp-nbr-af)# route-policy bgp_out out
RP/0/RP0/CPU0:router_C(config-bgp-nbr-af)# exit
RP/0/RP0/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/RP0/CPU0:router_C(config-bgp)# neighbor 172.29.50.71
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# remote-as 1
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# description to EPE_controller
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# address-family link-state link-state
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# exit
RP/0/RP0/CPU0:router_C(config-bgp)# exit
```

Step 3 Commit the configuration.

Example:

```
RP/0/RP0/CPU0:router_C(config)# commit
```

Step 4 Verify the configuration.

Example:

```
RP/0/RP0/CPU0:router_C# 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/RP0/CPU0:router_C# show mpls forwarding labels 24002 24003
Local  Outgoing  Prefix      Outgoing  Next Hop  Bytes
Label  Label      or ID      Interface  _____  Switched
-----
24002  Unlabelled No ID      Te0/0/0/1  192.168.1.2  0
24003  Unlabelled No ID      Te0/0/0/2  192.168.1.3  0
```

The output shows that node C installed peer node SIDs in the Forwarding Information Base (FIB).



CHAPTER 6

SR-TE Policy Overview

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 following 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.

An SR-TE policy is identified as an ordered list (head-end, color, end-point):

- Head-end – Where the SR-TE policy is instantiated
- Color – A numerical value that distinguishes between two or more policies to the same node pairs (Head-end – End point)
- End-point – The destination of the SR-TE policy

Every SR-TE policy has a color value. Every policy between the same node pairs requires a unique color value.

An SR-TE policy uses one or more candidate paths. A candidate path is a single segment list (SID-list) or a set of weighted SID-lists (for weighted equal cost multi-path [WECMP]). A candidate path is either dynamic or explicit. See *SR-TE Policy Path Types* section for more information.

- [Auto-Route Announce for SR-TE, on page 37](#)
- [Instantiation of an SR Policy, on page 38](#)
- [SR-TE Policy Path Types, on page 38](#)
- [Protocols, on page 47](#)
- [Traffic Steering, on page 50](#)

Auto-Route Announce for SR-TE

Auto-route announce for SR-TE cannot handle LDP-over-SR-TE if the SR-TE terminates at an LDP mid-node.

Let us consider the following topology:

R1---R2---R3---R4---R5---R6

If there is an SR-TE route from R1 to R4, and an LDP prefix is learnt from R6, then auto-route announce will fail.

Instantiation of an SR Policy

An SR policy is instantiated, or implemented, at the head-end router.

The following sections provide details on the SR policy instantiation methods:

- [Manually Provisioned SR Policy, on page 38](#)

Manually Provisioned SR Policy

Manually provisioned SR policies are configured on the head-end router. These policies can use dynamic paths or explicit paths. See the [SR-TE Policy Path Types, on page 38](#) section for information on manually provisioning an SR policy using dynamic or explicit paths.

SR-TE Policy Path Types

A **dynamic** path is based on an optimization objective and a set of constraints. The head-end computes a solution, resulting in a SID-list or a set of SID-lists. When the topology changes, a new path is computed. If the head-end does not have enough information about the topology, the head-end might delegate the computation to a Segment Routing Path Computation Element (SR-PCE). For information on configuring SR-PCE, see *Configure Segment Routing Path Computation Element* chapter.

An **explicit** path is a specified SID-list or set of SID-lists.

An SR-TE policy initiates a single (selected) path in RIB/FIB. This is the preferred valid candidate path.

A candidate path has the following characteristics:

- It has a preference – If two policies have same {color, endpoint} but different preferences, the policy with the highest preference is selected.
- It is associated with a single binding SID (BSID) – A BSID conflict occurs when there are different SR policies with the same BSID. In this case, the policy that is installed first gets the BSID and is selected.
- It is valid if it is usable.

A path is selected when the path is valid and its preference is the best among all candidate paths for that policy.



Note The protocol of the source is not relevant in the path selection logic.

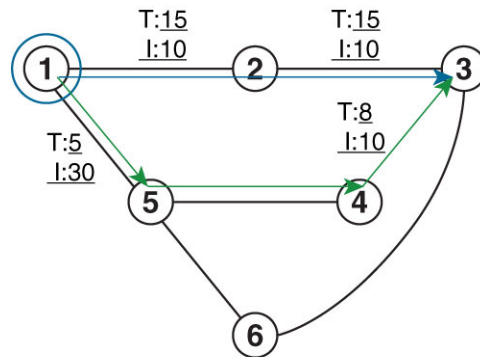
Dynamic Paths

Optimization Objectives

Optimization objectives allow the head-end router to compute a SID-list that expresses the shortest dynamic path according to the selected metric type:

- IGP metric — Refer to the "Implementing IS-IS" and "Implementing OSPF" chapters in the *Routing Configuration Guide for Series Routers*.
- TE metric — See the [Configure Interface TE Metrics, on page 39](#) section for information about configuring TE metrics.

This example shows a dynamic path from head-end router 1 to end-point router 3 that minimizes IGP or TE metric:



Default IGP link metric: I:10
Default TE link metric T:10

520018

- The blue path uses the minimum IGP metric: Min-Metric (1 → 3, IGP) = SID-list <16003>; cumulative IGP metric: 20
- The green path uses the minimum TE metric: Min-Metric (1 → 3, TE) = SID-list <16005, 16004, 16003>; cumulative TE metric: 23

Configure Interface TE Metrics

Use the **metric value** command in SR-TE interface submode to configure the TE metric for interfaces. The **value** range is from 0 to 2147483647.

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# interface type interface-path-id
Router(config-sr-te-if)# metric value
```

Configuring TE Metric: Example

The following configuration example shows how to set the TE metric for various interfaces:

```
segment-routing
traffic-eng
interface TenGigE0/0/0/0
metric 100
!
interface TenGigE0/0/0/1
metric 1000
!
interface TenGigE0/0/2/0
metric 50
!
!
end
```

Constraints

Constraints allow the head-end router to compute a dynamic path according to the selected metric type:

- **TE affinity** — You can apply a color or name to links or interfaces by assigning affinity bit-maps to them. You can then specify an affinity (or relationship) between an SR policy path and link colors. SR-TE computes a path that includes or excludes links that have specific colors, or combinations of colors. See the [Named Interface Link Admin Groups and SR-TE Affinity Maps](#), on page 40 section for information on named interface link admin groups and SR-TE Affinity Maps.
- **Disjoint** — SR-TE computes a path that is disjoint from another path in the same disjoint-group. Disjoint paths do not share network resources. Path disjointness may be required for paths between the same pair of nodes, between different pairs of nodes, or a combination (only same head-end or only same end-point).
- **Flexible Algorithm** — Flexible Algorithm allows for user-defined algorithms where the IGP computes paths based on a user-defined combination of metric type and constraint.

Named Interface Link Admin Groups and SR-TE Affinity Maps

Named Interface Link Admin Groups and SR-TE Affinity Maps provide a simplified and more flexible means of configuring link attributes and path affinities to compute paths for SR-TE policies.

In the traditional TE scheme, links are configured with attribute-flags that are flooded with TE link-state parameters using Interior Gateway Protocols (IGPs), such as Open Shortest Path First (OSPF).

Named Interface Link Admin Groups and SR-TE Affinity Maps let you assign, or map, up to 32 color names for affinity and attribute-flag attributes instead of 32-bit hexadecimal numbers. After mappings are defined, the attributes can be referred to by the corresponding color name in the CLI. Furthermore, you can define constraints using *include-any*, *include-all*, and *exclude-any* arguments, where each statement can contain up to 10 colors.



Note You can configure affinity constraints using attribute flags or the Flexible Name Based Policy Constraints scheme; however, when configurations for both schemes exist, only the configuration pertaining to the new scheme is applied.

Configure Named Interface Link Admin Groups and SR-TE Affinity Maps

Use the **affinity name** *NAME* command in SR-TE interface submode to assign affinity to interfaces. Configure this on routers with interfaces that have an associated admin group attribute.

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# interface TenGigE0/0/1/2
Router(config-sr-if)# affinity
Router(config-sr-if-affinity)# name RED
```

Use the **affinity-map name** *NAME* **bit-position** *bit-position* command in SR-TE sub-mode to define affinity maps. The *bit-position* range is from 0 to 255.

Configure affinity maps on the following routers:

- Routers with interfaces that have an associated admin group attribute.
- Routers that act as SR-TE head-ends for SR policies that include affinity constraints.


```

Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# affinity-map
Router(config-sr-te-affinity-map)# name RED bit-position 23

```

Configuring Link Admin Group: Example

The following example shows how to assign affinity to interfaces and to define affinity maps. This configuration is applicable to any router (SR-TE head-end or transit node) with colored interfaces.

```

segment-routing
traffic-eng
interface TenGigE0/0/1/1
  affinity
  name CROSS
  name RED
  !
  !
interface TenGigE0/0/1/2
  affinity
  name RED
  !
  !
interface TenGigE0/0/2/0
  affinity
  name BLUE
  !
  !
affinity-map
  name RED bit-position 23
  name BLUE bit-position 24
  name CROSS bit-position 25
  !
end

```

Configure SR Policy with Dynamic Path

To configure a SR-TE policy with a dynamic path, optimization objectives, and affinity constraints, complete the following configurations:

1. Define the optimization objectives. See the [Optimization Objectives, on page 38](#) section.
2. Define the constraints. See the [Constraints, on page 40](#) section.
3. Create the policy.

Behaviors and Limitations

The following example shows a configuration of an SR policy at an SR-TE head-end router. The policy has a dynamic path with optimization objectives and affinity constraints computed by the head-end router.

```

segment-routing
traffic-eng
policy foo
  color 100 end-point ipv4 1.1.1.2
  candidate-paths
  preference 100
  dynamic

```


When configuring an explicit path using IP addresses of intermediate links, the SR-TE process selects either the protected or the unprotected Adj-SID of the link, depending on the order in which the Adj-SIDs were received.

Configure Local SR-TE Policy Using Explicit Paths

Create a segment list with IP addresses:

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# segment-list name SIDLIST1
Router(config-sr-te-sl)# index 10 address ipv4 1.1.1.2
Router(config-sr-te-sl)# index 20 address ipv4 1.1.1.3
Router(config-sr-te-sl)# index 30 address ipv4 1.1.1.4
Router(config-sr-te-sl)# exit
```

Create a segment list with MPLS labels:

```
Router(config-sr-te)# segment-list name SIDLIST2
Router(config-sr-te-sl)# index 10 mpls label 16002
Router(config-sr-te-sl)# index 20 mpls label 16003
Router(config-sr-te-sl)# index 30 mpls label 16004
Router(config-sr-te-sl)# exit
```

Create a segment list with IP addresses and MPLS labels:

```
Router(config-sr-te)# segment-list name SIDLIST3
Router(config-sr-te-sl)# index 10 address ipv4 1.1.1.2
Router(config-sr-te-sl)# index 20 mpls label 16003
Router(config-sr-te-sl)# index 30 mpls label 16004
Router(config-sr-te-sl)# exit
```

Create the SR-TE policy:

```
Router(config-sr-te)# policy POLICY1
Router(config-sr-te-policy)# color 10 end-point ipv4 1.1.1.4
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 100
Router(config-sr-te-policy-path-pref)# explicit segment-list SIDLIST1
Router(config-sr-te-policy-path-pref)# exit
Router(config-sr-te-pp-info)# exit
```

```
Router(config-sr-te)# policy POLICY2
Router(config-sr-te-policy)# color 20 end-point ipv4 1.1.1.4
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 100
Router(config-sr-te-policy-path-pref)# explicit segment-list SIDLIST2
Router(config-sr-te-policy-path-pref)# exit
Router(config-sr-te-pp-info)# exit
```

```
Router(config-sr-te)# policy POLICY3
Router(config-sr-te-policy)# color 30 end-point ipv4 1.1.1.4
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 100
Router(config-sr-te-policy-path-pref)# explicit segment-list SIDLIST3
Router(config-sr-te-policy-path-pref)# commit
```

Running Configuration

```

Router# show running-configuration
segment-routing
traffic-eng
segment-list SIDLIST1
  index 10 address ipv4 1.1.1.2
  index 20 address ipv4 1.1.1.3
  index 30 address ipv4 1.1.1.4
!
segment-list SIDLIST2
  index 10 mpls label 16002
  index 20 mpls label 16003
  index 30 mpls label 16004
!
segment-list SIDLIST3
  index 10 address ipv4 1.1.1.2
  index 20 mpls label 16003
  index 30 mpls label 16004
!
policy POLICY1
  color 10 end-point ipv4 1.1.1.4
  candidate-paths
    preference 100
    explicit segment-list SIDLIST1
  !
!
!
policy POLICY2
  color 20 end-point ipv4 1.1.1.4
  candidate-paths
    preference 100
    explicit segment-list SIDLIST2
  !
!
!
policy POLICY3
  color 30 end-point ipv4 1.1.1.4
  candidate-paths
    preference 100
    explicit segment-list SIDLIST3
  !
!
!

```

Verification

```

Router# show segment-routing traffic-eng policy name srte_c_20_ep_1.1.1.4
Sat Jul  8 12:25:34.114 UTC
SR-TE policy database
-----
Name: P1 (Color: 20, End-point: 1.1.1.4)
Status:
  Admin: up Operational: up for 00:06:21 (since Jul  8 12:19:13.198)
Candidate-paths:
  Preference 10:
    Explicit: segment-list SIDLIST1 (active)
    Weight: 2
    400102 [Prefix-SID, 2.1.1.1]
    400106
    Explicit: segment-list SIDLIST2 (active)
    Weight: 2

```

```

400222 [Prefix-SID, 22.11.1.1]
400106
Attributes:
  Binding SID: 15001
  Allocation mode: explicit
  State: programmed
  Policy selected: yes
  Forward Class: 0

```

Configuring Explicit Path with Affinity Constraint Validation

To fully configure SR-TE flexible name-based policy constraints, you must complete these high-level tasks in order:

1. Assign Color Names to Numeric Values
2. Associate Affinity-Names with SR-TE Links
3. Associate Affinity Constraints for SR-TE Policies

```

/* Enter the global configuration mode and assign color names to numeric values
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# affinity-map
Router(config-sr-te-affinity-map)# blue bit-position 0
Router(config-sr-te-affinity-map)# green bit-position 1
Router(config-sr-te-affinity-map)# red bit-position 2
Router(config-sr-te-affinity-map)# exit

```

```

/* Associate affinity-names with SR-TE links
Router(config-sr-te)# interface Gi0/0/0/0
Router(config-sr-te-if)# affinity
Router(config-sr-te-if-affinity)# blue
Router(config-sr-te-if-affinity)# exit
Router(config-sr-te-if)# exit
Router(config-sr-te)# interface Gi0/0/0/1
Router(config-sr-te-if)# affinity
Router(config-sr-te-if-affinity)# blue
Router(config-sr-te-if-affinity)# green
Router(config-sr-te-if-affinity)# exit
Router(config-sr-te-if)# exit
Router(config-sr-te)#

```

```

/* Associate affinity constraints for SR-TE policies
Router(config-sr-te)# segment-list name SIDLIST1
Router(config-sr-te-sl)# index 10 address ipv4 1.1.1.2
Router(config-sr-te-sl)# index 20 address ipv4 2.2.2.23
Router(config-sr-te-sl)# index 30 address ipv4 1.1.1.4
Router(config-sr-te-sl)# exit
Router(config-sr-te)# segment-list name SIDLIST2
Router(config-sr-te-sl)# index 10 address ipv4 1.1.1.2
Router(config-sr-te-sl)# index 30 address ipv4 1.1.1.4
Router(config-sr-te-sl)# exit
Router(config-sr-te)# segment-list name SIDLIST3
Router(config-sr-te-sl)# index 10 address ipv4 1.1.1.5
Router(config-sr-te-sl)# index 30 address ipv4 1.1.1.4
Router(config-sr-te-sl)# exit

```

```

Router(config-sr-te)# policy POLICY1
Router(config-sr-te-policy)# color 20 end-point ipv4 1.1.1.4
Router(config-sr-te-policy)# binding-sid mpls 1000
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 200
Router(config-sr-te-policy-path-pref)# constraints affinity exclude-any red
Router(config-sr-te-policy-path-pref)# explicit segment-list SIDLIST1
Router(config-sr-te-pp-info)# exit
Router(config-sr-te-policy-path-pref)# explicit segment-list SIDLIST2
Router(config-sr-te-pp-info)# exit
Router(config-sr-te-policy-path-pref)# exit
Router(config-sr-te-policy-path)# preference 100
Router(config-sr-te-policy-path-pref)# explicit segment-list SIDLIST3

```

Running Configuration

```

Router# show running-configuration
segment-routing
 traffic-eng

 interface GigabitEthernet0/0/0/0
  affinity
   blue
  !
 !
 interface GigabitEthernet0/0/0/1
  affinity
   blue
   green
  !
 !

 segment-list name SIDLIST1
  index 10 address ipv4 1.1.1.2
  index 20 address ipv4 2.2.2.23
  index 30 address ipv4 1.1.1.4
 !
 segment-list name SIDLIST2
  index 10 address ipv4 1.1.1.2
  index 30 address ipv4 1.1.1.4
 !
 segment-list name SIDLIST3
  index 10 address ipv4 1.1.1.5
  index 30 address ipv4 1.1.1.4
 !
 policy POLICY1
  binding-sid mpls 1000
  color 20 end-point ipv4 1.1.1.4
  candidate-paths
  preference 100
   explicit segment-list SIDLIST3
  !
 !
  preference 200
   explicit segment-list SIDLIST1
  !
   explicit segment-list SIDLIST2
  !
  constraints
   affinity

```

```

        exclude-any
        red
        !
        !
        !
        !
        !
        affinity-map
        blue bit-position 0
        green bit-position 1
        red bit-position 2
        !
        !
        !

```

Protocols

Path Computation Element Protocol

The path computation element protocol (PCEP) describes a set of procedures by which a path computation client (PCC) can report and delegate control of head-end label switched paths (LSPs) sourced from the PCC to a PCE peer. The PCE can request the PCC to update and modify parameters of LSPs it controls. The stateful model also enables a PCC to allow the PCE to initiate computations allowing the PCE to perform network-wide orchestration.

BGP SR-TE

SR-TE can be used by data center (DC) operators to provide different levels of Service Level Assurance (SLA). Setting up SR-TE paths using BGP (BGP SR-TE) simplifies DC network operation without introducing a new protocol for this purpose.

Explicit BGP SR-TE

Explicit BGP SR-TE uses an SR-TE policy (identified by a unique color ID) that contains a list of explicit paths with SIDs that correspond to each explicit path. A BGP speaker signals an explicit SR-TE policy to a remote peer, which triggers the setup of an SR-TE policy with specific characteristics and explicit paths. On the receiver side, an SR-TE policy that corresponds to the explicit path is setup by BGP. The packets for the destination mentioned in the BGP update follow the explicit path described by the policy. Each policy can include multiple explicit paths, and TE will create a policy for each path.



Note For more information on routing policies and routing policy language (RPL), refer to the "Implementing Routing Policy" chapter in the *Routing Configuration Guide for Cisco NCS 540 Series Routers*.

IPv4 and IPv6 SR policies can be advertised over BGPv4 or BGPv6 sessions between the SR-TE controller and the SR-TE headend. The Cisco IOS-XR implementation supports the following combinations:

- IPv4 SR policy advertised over BGPv4 session

- IPv6 SR policy advertised over BGPv4 session
- IPv6 SR policy advertised over BGPv6 session

Configure Explicit BGP SR-TE

Perform this task to configure explicit BGP SR-TE:

Procedure

	Command or Action	Purpose
Step 1	configure	
Step 2	router bgp <i>as-number</i> Example: RP/0/RP0/CPU0:router(config)# router bgp 65000	Specifies the BGP AS number and enters the BGP configuration mode, allowing you to configure the BGP routing process.
Step 3	bgp router-id <i>ip-address</i> Example: RP/0/RP0/CPU0:router(config-bgp)# bgp router-id 1.1.1.1	Configures the local router with a specified router ID.
Step 4	address-family { ipv4 ipv6 } sr-policy Example: RP/0/RP0/CPU0:router(config-bgp)# address-family ipv4 sr-policy	Specifies either the IPv4 or IPv6 address family and enters address family configuration submode.
Step 5	exit	
Step 6	neighbor <i>ip-address</i> Example: RP/0/RP0/CPU0:router(config-bgp)# neighbor 10.10.0.1	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.
Step 7	remote-as <i>as-number</i> Example: RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 1	Creates a neighbor and assigns a remote autonomous system number to it.

	Command or Action	Purpose
Step 8	address-family {ipv4 ipv6} sr-policy Example: RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family ipv4 sr-policy	Specifies either the IPv4 or IPv6 address family and enters address family configuration submode.
Step 9	route-policy route-policy-name {in out} Example: RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass out	Applies the specified policy to IPv4 or IPv6 unicast routes.

Example: BGP SR-TE with BGPv4 Neighbor to BGP SR-TE Controller

The following configuration shows the an SR-TE head-end with a BGPv4 session towards a BGP SR-TE controller. This BGP session is used to signal both IPv4 and IPv6 SR policies.

```
router bgp 65000
  bgp router-id 1.1.1.1
  !
  address-family ipv4 sr-policy
  !
  address-family ipv6 sr-policy
  !
  neighbor 10.1.3.1
    remote-as 10
    description *** eBGP session to BGP SRTE controller ***
    address-family ipv4 sr-policy
      route-policy pass in
      route-policy pass out
    !
    address-family ipv6 sr-policy
      route-policy pass in
      route-policy pass out
    !
  !
  !
```

Example: BGP SR-TE with BGPv6 Neighbor to BGP SR-TE Controller

The following configuration shows an SR-TE head-end with a BGPv6 session towards a BGP SR-TE controller. This BGP session is used to signal IPv6 SR policies.

```
router bgp 65000
  bgp router-id 1.1.1.1
  address-family ipv6 sr-policy
  !
  neighbor 3001::10:1:3:1
    remote-as 10
    description *** eBGP session to BGP SRTE controller ***
    address-family ipv6 sr-policy
      route-policy pass in
      route-policy pass out
    !
  !
  !
```

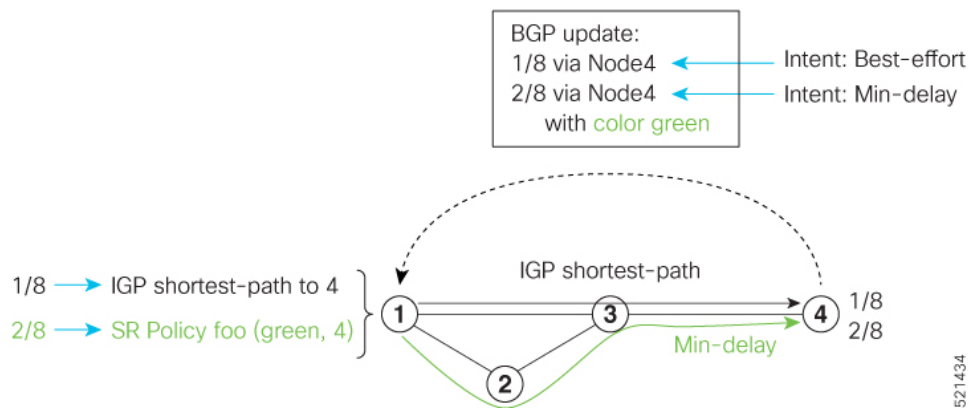
Traffic Steering

Automated Steering

Automated steering (AS) allows service traffic to be automatically steered onto the required transport SLA path programmed by an SR policy.

With AS, BGP automatically steers traffic onto an SR Policy based on the next-hop and color of a BGP service route. The color of a BGP service route is specified by a color extended community attribute. This color is used as a transport SLA indicator, such as min-delay or min-cost.

When the next-hop and color of a BGP service route matches the end-point and color of an SR Policy, BGP automatically installs the route resolving onto the BSID of the matching SR Policy. Recall that an SR Policy on a head-end is uniquely identified by an end-point and color.



When a BGP route has multiple extended-color communities, each with a valid SR Policy, the BGP process installs the route on the SR Policy giving preference to the color with the highest numerical value.

The granularity of AS behaviors can be applied at multiple levels, for example:

- At a service level—When traffic destined to all prefixes in a given service is associated to the same transport path type. All prefixes share the same color.
- At a destination/prefix level—When traffic destined to a prefix in a given service is associated to a specific transport path type. Each prefix could be assigned a different color.
- At a flow level—When flows destined to the same prefix are associated with different transport path types

AS behaviors apply regardless of the instantiation method of the SR policy, including:

- On-demand SR policy
- Manually provisioned SR policy
- PCE-initiated SR policy

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.

Explicit Binding SID

Use the **binding-sid mpls label** command in SR-TE policy configuration mode to specify the explicit BSID. Explicit BSIDs are allocated from the segment routing local block (SRLB) or the dynamic range of labels. A best-effort is made to request and obtain the BSID for the SR-TE policy. If requested BSID is not available (if it does not fall within the available SRLB or is already used by another application or SR-TE policy), the policy stays down.

Use the **binding-sid explicit {fallback-dynamic | enforce-srlb}** command to specify how the BSID allocation behaves if the BSID value is not available.

- Fallback to dynamic allocation – If the BSID is not available, the BSID is allocated dynamically and the policy comes up:

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# binding-sid explicit fallback-dynamic
```

- Strict SRLB enforcement – If the BSID is not within the SRLB, the policy stays down:

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
```

```
Router(config-sr-te)# binding-sid explicit enforce-srlb
```

This example shows how to configure an SR policy to use an explicit BSID of 1000. If the BSID is not available, the BSID is allocated dynamically and the policy comes up.

```
segment-routing
traffic-eng
binding-sid explicit fallback-dynamic
policy goo
binding-sid mpls 1000
!
!
!
```

Stitching SR-TE Polices Using Binding SID: Example

In this example, three SR-TE policies are stitched together to form a seamless end-to-end path from node 1 to node 10. The path is a chain of SR-TE policies stitched together using the binding-SIDs of intermediate policies, providing a seamless end-to-end path.

Figure 3: Stitching SR-TE Polices Using Binding SID

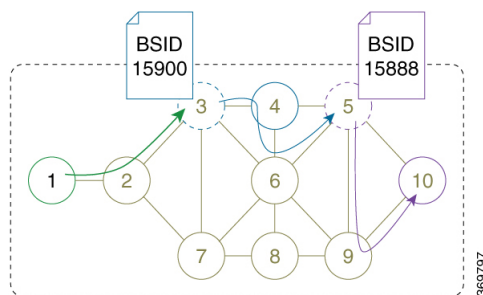


Table 2: Router IP Address

Router	Prefix Address	Prefix SID/Adj-SID
3	Loopback0 - 1.1.1.3	Prefix SID - 16003
4	Loopback0 - 1.1.1.4 Link node 4 to node 6 - 10.4.6.4	Prefix SID - 16004 Adjacency SID - dynamic
5	Loopback0 - 1.1.1.5	Prefix SID - 16005
6	Loopback0 - 1.1.1.6 Link node 4 to node 6 - 10.4.6.6	Prefix SID - 16006 Adjacency SID - dynamic
9	Loopback0 - 1.1.1.9	Prefix SID - 16009
10	Loopback0 - 1.1.1.10	Prefix SID - 16010

Procedure

Step 1

On node 5, do the following:

- a) Define an SR-TE policy with an explicit path configured using the loopback interface IP addresses of node 9 and node 10.
- b) Define an explicit binding-SID (**mpls label 15888**) allocated from SRLB for the SR-TE policy.

Example:

Node 5

```

segment-routing
traffic-eng
segment-list PATH-9_10
index 10 address ipv4 1.1.1.9
index 20 address ipv4 1.1.1.10
!
policy foo
binding-sid mpls 15888
color 777 end-point ipv4 1.1.1.10
candidate-paths
preference 100
explicit segment-list PATH5-9_10
!
!
!
!
!
!
RP/0/RSP0/CPU0:Node-5# show segment-routing traffic-eng policy color 777

SR-TE policy database
-----

Color: 777, End-point: 1.1.1.10
Name: srte_c_777_ep_1.1.1.10
Status:
Admin: up Operational: up for 00:00:52 (since Aug 19 07:40:12.662)
Candidate-paths:
Preference: 100 (configuration) (active)
Name: foo
Requested BSID: 15888
PCC info:
Symbolic name: cfg_foo_discr_100
PLSP-ID: 70
Explicit: segment-list PATH-9_10 (valid)
Weight: 1, Metric Type: TE
16009 [Prefix-SID, 1.1.1.9]
16010 [Prefix-SID, 1.1.1.10]
Attributes:
Binding SID: 15888 (SRLB)
Forward Class: 0
Steering BGP disabled: no
IPv6 caps enable: yes

```

Step 2

On node 3, do the following:

- a) Define an SR-TE policy with an explicit path configured using the following:
 - Loopback interface IP address of node 4

- Interface IP address of link between node 4 and node 6
- Loopback interface IP address of node 5
- Binding-SID of the SR-TE policy defined in Step 1 (**mpls label 15888**)

Note This last segment allows the stitching of these policies.

- b) Define an explicit binding-SID (**mpls label 15900**) allocated from SRLB for the SR-TE policy.

Example:

Node 3

```

segment-routing
traffic-eng
segment-list PATH-4_4-6_5_BSID
index 10 address ipv4 1.1.1.4
index 20 address ipv4 10.4.6.6
index 30 address ipv4 1.1.1.5
index 40 mpls label 15888
!
policy baa
binding-sid mpls 15900
color 777 end-point ipv4 1.1.1.5
candidate-paths
preference 100
explicit segment-list PATH-4_4-6_5_BSID
!
!
!
!
!
!
RP/0/RSP0/CPU0:Node-3# show segment-routing traffic-eng policy color 777

SR-TE policy database
-----

Color: 777, End-point: 1.1.1.5
Name: srte_c_777_ep_1.1.1.5
Status:
Admin: up Operational: up for 00:00:32 (since Aug 19 07:40:32.662)
Candidate-paths:
Preference: 100 (configuration) (active)
Name: baa
Requested BSID: 15900
PCC info:
Symbolic name: cfg_baa_discr_100
PLSP-ID: 70
Explicit: segment-list PATH-4_4-6_5_BSID (valid)
Weight: 1, Metric Type: TE
16004 [Prefix-SID, 1.1.1.4]
80005 [Adjacency-SID, 10.4.6.4 - 10.4.6.6]
16005 [Prefix-SID, 1.1.1.5]
15888
Attributes:
Binding SID: 15900 (SRLB)
Forward Class: 0
Steering BGP disabled: no
IPv6 caps enable: yes

```

- Step 3** On node 1, define an SR-TE policy with an explicit path configured using the loopback interface IP address of node 3 and the binding-SID of the SR-TE policy defined in step 2 (**mpls label 15900**). This last segment allows the stitching of these policies.

Example:

Node 1

```
segment-routing
traffic-eng
segment-list PATH-3_BSID
index 10 address ipv4 1.1.1.3
index 20 mpls label 15900
!
policy bar
color 777 end-point ipv4 1.1.1.3
candidate-paths
preference 100
explicit segment-list PATH-3_BSID
!
!
!
!
!
```

```
RP/0/RSP0/CPU0:Node-1# show segment-routing traffic-eng policy color 777
```

```
SR-TE policy database
```

```
-----
Color: 777, End-point: 1.1.1.3
Name: srte_c_777_ep_1.1.1.3
Status:
  Admin: up Operational: up for 00:00:12 (since Aug 19 07:40:52.662)
Candidate-paths:
  Preference: 100 (configuration) (active)
  Name: bar
  Requested BSID: dynamic
  PCC info:
    Symbolic name: cfg_bar_discr_100
    PLSP-ID: 70
  Explicit: segment-list PATH-3_BSID (valid)
    Weight: 1, Metric Type: TE
    16003 [Prefix-SID, 1.1.1.3]
    15900
Attributes:
  Binding SID: 80021
  Forward Class: 0
  Steering BGP disabled: no
  IPv6 caps enable: yes
```

L2VPN Preferred Path

EVPN VPWS Preferred Path over SR-TE Policy feature allows you to set the preferred path between the two end-points for EVPN VPWS pseudowire (PW) using SR-TE policy.

L2VPN VPLS or VPWS Preferred Path over SR-TE Policy feature allows you to set the preferred path between the two end-points for L2VPN Virtual Private LAN Service (VPLS) or Virtual Private Wire Service (VPWS) using SR-TE policy.

Refer to the [EVPN VPWS Preferred Path over SR-TE Policy](#) and [L2VPN VPLS or VPWS Preferred Path over SR-TE Policy](#) sections in the "L2VPN Services over Segment Routing for Traffic Engineering Policy" chapter of the *L2VPN and Ethernet Services Configuration Guide*.



CHAPTER 7

About SR-PCE

The path computation element protocol (PCEP) describes a set of procedures by which a path computation client (PCC) can report and delegate control of head-end label switched paths (LSPs) sourced from the PCC to a PCE peer. The PCE can request the PCC to update and modify parameters of LSPs it controls. The stateful model also enables a PCC to allow the PCE to initiate computations allowing the PCE to perform network-wide orchestration.

SR-PCE learns topology information by way of IGP (OSPF or IS-IS) or through BGP Link-State (BGP-LS).

SR-PCE is capable of computing paths using the following methods:

- TE metric—SR-PCE uses the TE metric in its path calculations to optimize cumulative TE metric.
- IGP metric—SR-PCE uses the IGP metric in its path calculations to optimize reachability.
- LSP Disjointness—SR-PCE uses the path computation algorithms to compute a pair of disjoint LSPs. The disjoint paths can originate from the same head-end or different head-ends. Disjoint level refers to the type of resources that should not be shared by the two computed paths. SR-PCE supports the following disjoint path computations:
 - Link – Specifies that links are not shared on the computed paths.
 - Node – Specifies that nodes are not shared on the computed paths.
 - SRLG – Specifies that links with the same SRLG value are not shared on the computed paths.
 - SRLG-node – Specifies that SRLG and nodes are not shared on the computed paths.

When the first request is received with a given disjoint-group ID, the first LSP is computed, encoding the shortest path from the first source to the first destination. When the second LSP request is received with the same disjoint-group ID, information received in both requests is used to compute two disjoint paths: one path from the first source to the first destination, and another path from the second source to the second destination. Both paths are computed at the same time.

- [Configure SR-PCE, on page 57](#)
- [Configure the Disjoint Policy \(Optional\), on page 59](#)

Configure SR-PCE

This task explains how to configure SR-PCE.

Before you begin

The Cisco IOS XRv 9000 is the recommended platform to act as the SR-PCE.

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# <code>configure</code>	Enters mode.
Step 2	pce Example: RP/0/RP0/CPU0:router(config)# <code>pce</code>	Enables PCE and enters PCE configuration mode.
Step 3	address ipv4 address Example: RP/0/RP0/CPU0:router(config-pce)# address ipv4 192.168.0.1	Configures a PCE IPv4 address.
Step 4	state-sync ipv4 address Example: RP/0/RP0/CPU0:router(config-pce)# state-sync ipv4 192.168.0.3	Configures the remote peer for state synchronization.
Step 5	tcp-buffer size size Example: RP/0/RP0/CPU0:router(config-pce)# tcp-buffer size 1024000	Configures the transmit and receive TCP buffer size for each PCEP session, in bytes. The default buffer size is 256000. The valid range is from 204800 to 1024000.
Step 6	password {clear encrypted} password Example: RP/0/RP0/CPU0:router(config-pce)# password encrypted pwd1	Enables TCP MD5 authentication for all PCEP peers. Any TCP segment coming from the PCC that does not contain a MAC matching the configured password will be rejected. Specify if the password is encrypted or clear text.
Step 7	segment-routing {strict-sid-only te-latency} Example: RP/0/RP0/CPU0:router(config-pce)#	Configures the segment routing algorithm to use strict SID or TE latency. Note This setting is global and applies to all LSPs that request a path from this controller.

	Command or Action	Purpose
	<code>segment-routing strict-sid-only</code>	
Step 8	timers Example: RP/0/RP0/CPU0:router(config-pce) # timers	Enters timer configuration mode.
Step 9	keepalive time Example: RP/0/RP0/CPU0:router(config-pce-timers) # keepalive 60	Configures the timer value for locally generated keep-alive messages. The default time is 30 seconds.
Step 10	minimum-peer-keepalive time Example: RP/0/RP0/CPU0:router(config-pce-timers) # minimum-peer-keepalive 30	Configures the minimum acceptable keep-alive timer that the remote peer may propose in the PCEP OPEN message during session establishment. The default time is 20 seconds.
Step 11	reoptimization time Example: RP/0/RP0/CPU0:router(config-pce-timers) # reoptimization 600	Configures the re-optimization timer. The default timer is 1800 seconds.
Step 12	exit Example: RP/0/RP0/CPU0:router(config-pce-timers) # exit	Exits timer configuration mode and returns to PCE configuration mode.

Configure the Disjoint Policy (Optional)

This task explains how to configure the SR-PCE to compute disjointness for a pair of LSPs signaled by PCCs that do not include the PCEP association group-ID object in their PCEP request. This can be beneficial for deployments where PCCs do not support this PCEP object or when the network operator prefers to manage the LSP disjoint configuration centrally.

Procedure

	Command or Action	Purpose
Step 1	disjoint-path Example: <pre>RP/0/RP0/CPU0:router(config-pce)# disjoint-path</pre>	Enters disjoint configuration mode.
Step 2	group-id value type {link node srlg srlg-node} [sub-id value] Example: <pre>RP/0/RP0/CPU0:router(config-pce-disjoint)# group-id 1 type node sub-id 1</pre>	<p>Configures the disjoint group ID and defines the preferred level of disjointness (the type of resources that should not be shared by the two paths):</p> <ul style="list-style-type: none"> • link—Specifies that links are not shared on the computed paths. • node—Specifies that nodes are not shared on the computed paths. • srlg—Specifies that links with the same SRLG value are not shared on the computed paths. • srlg-node—Specifies that SRLG and nodes are not shared on the computed paths. <p>If a pair of paths that meet the requested disjointness level cannot be found, then the paths will automatically fallback to a lower level:</p> <ul style="list-style-type: none"> • If the requested disjointness level is SRLG or node, then link-disjoint paths will be computed. • If the requested disjointness level was link, or if the first fallback from SRLG or node disjointness failed, then the lists of segments encoding two shortest paths, without any disjointness constraint, will be computed.
Step 3	strict Example: <pre>RP/0/RP0/CPU0:router(config-pce-disjoint)# strict</pre>	<p>(Optional) Prevents the automatic fallback behavior of the preferred level of disjointness. If a pair of paths that meet the requested disjointness level cannot be found, the disjoint calculation terminates and no new path is provided. The existing path is not modified.</p>

	Command or Action	Purpose
Step 4	<p>lsp {1 2} pcc ipv4 address lsp-name <i>lsp_name</i> [shortest-path]</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-pce-disjoint)# lsp 1 pcc ipv4 192.168.0.1 lsp-name rtrA_t1 shortest-path RP/0/RP0/CPU0:router(config-pce-disjoint)# lsp 2 pcc ipv4 192.168.0.5 lsp-name rtrE_t2</pre>	<p>Adds LSPs to the disjoint group.</p> <p>The shortest-path keyword forces one of the disjoint paths to follow the shortest path from the source to the destination. This option can only be applied to the the first LSP specified.</p>



CHAPTER 8

Limitations

Only two backup labels are supported.

- [Usage Guidelines and Limitations](#), on page 63
- [Configuring TI-LFA for IS-IS](#), on page 64
- [Configuring TI-LFA for OSPF](#), on page 66
- [TI-LFA Node and SRLG Protection: Examples](#), on page 67

Usage Guidelines and Limitations

The TI-LFA guidelines and limitations are listed below:

TI-LFA Functionality	IS-IS ¹	OSPFv2
<i>Protected Traffic Types</i>		
Protection for SR labeled traffic	Supported	Supported
Protection of IPv4 unlabeled traffic	Supported (IS-ISv4)	Supported
Protection of IPv6 unlabeled traffic	Unsupported	N/A
<i>Protection Types</i>		
Link Protection	Supported	Supported
Node Protection	Supported	Supported
Local SRLG Protection	Supported	Supported
Weighted Remote SRLG Protection	Unsupported	Unsupported
Line Card Disjoint Protection	Unsupported	Unsupported
<i>Interface Types</i>		
Ethernet Interfaces	Supported	Supported
Ethernet Bundle Interfaces	Unsupported	Unsupported
TI-LFA over GRE Tunnel as Protecting Interface	Unsupported	Unsupported

TI-LFA Functionality	IS-IS ¹	OSPFv2
<i>Additional Functionality</i>		
BFD-triggered	Unsupported	Unsupported
BFDv6-triggered	Unsupported	N/A
Prefer backup path with lowest total metric	Unsupported	Unsupported
Prefer backup path from ECMP set	Supported	Supported
Prefer backup path from non-ECMP set	Supported	Supported
Load share prefixes across multiple backups paths	Unsupported	Unsupported
Limit backup computation up to the prefix priority	Supported	Supported

¹ Unless specified, IS-IS support is IS-ISv4 and IS-ISv6

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 13](#).
- 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)#
```

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	router isis instance-id Example: RP/0/RP0/CPU0:router(config)# router isis	Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.

	Command or Action	Purpose
	1	Note You can change the level of routing to be performed by a particular routing instance by using the is-type router configuration command.
Step 3	interface <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-isis)# interface GigabitEthernet0/0/0/1	Enters interface configuration mode.
Step 4	address-family ipv4 [unicast] Example: RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast	Specifies the IPv4 address family, and enters router address family configuration mode.
Step 5	fast-reroute per-prefix Example: RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix	Enables per-prefix fast reroute.
Step 6	fast-reroute per-prefix ti-lfa Example: RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix ti-lfa	Enables per-prefix TI-LFA fast reroute link protection.
Step 7	fast-reroute per-prefix tiebreaker {node-protecting srlg-disjoint} index <i>priority</i> Example: RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix tie-breaker srlg-disjoint index 100	Enables TI-LFA node or SRLG protection and specifies the tiebreaker priority. Valid <i>priority</i> values are from 1 to 255. The lower the <i>priority</i> value, the higher the priority of the rule. Link protection always has a lower priority than node or SRLG protection. Note The same attribute cannot be configured more than once on an interface. Note For IS-IS, TI-LFA node protection and SRLG protection can be configured on the interface or the instance.

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 25](#).
- 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)#
```

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	router ospf <i>process-name</i> Example: RP/0/RP0/CPU0:router(config)# router ospf 1	Enables OSPF routing for the specified routing process, and places the router in router configuration mode.
Step 3	area <i>area-id</i> Example: RP/0/RP0/CPU0:router(config-ospf)# area 1	Enters area configuration mode.
Step 4	interface <i>type interface-path-id</i> Example:	Enters interface configuration mode.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router (config-ospf-ar) # interface GigabitEthernet0/0/0/1	
Step 5	fast-reroute per-prefix Example: RP/0/RP0/CPU0:router (config-ospf-ar-if) # fast-reroute per-prefix	Enables per-prefix fast reroute.
Step 6	fast-reroute per-prefix ti-lfa Example: RP/0/RP0/CPU0:router (config-ospf-ar-if) # fast-reroute per-prefix ti-lfa	Enables per-prefix TI-LFA fast reroute link protection.
Step 7	fast-reroute per-prefix tiebreaker { node-protecting srlg-disjoint } index priority Example: RP/0/RP0/CPU0:router (config-ospf-ar-if) # fast-reroute per-prefix tie-breaker srlg-disjoint index 100	Enables TI-LFA node or SRLG protection and specifies the tiebreaker priority. Valid <i>priority</i> values are from 1 to 255. The lower the <i>priority</i> value, the higher the priority of the rule. Link protection always has a lower priority than node or SRLG protection. Note The same attribute cannot be configured more than once on an interface.

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.



CHAPTER 9

About Segment Routing Microloop Avoidance

Microloops are brief packet loops that occur in the network following a topology change (link down, link up, or metric change events). Microloops are caused by the non-simultaneous convergence of different nodes in the network. If nodes converge and send traffic to a neighbor node that has not converged yet, traffic may be looped between these two nodes, resulting in packet loss, jitter, and out-of-order packets.

The Segment Routing Microloop Avoidance feature detects if microloops are possible following a topology change. If a node computes that a microloop could occur on the new topology, the node creates a loop-free SR-TE policy path to the destination using a list of segments. After the RIB update delay timer expires, the SR-TE policy is replaced with regular forwarding paths.

- [Segment Routing Microloop Avoidance Limitations, on page 69](#)
- [Configure Segment Routing Microloop Avoidance for IS-IS, on page 69](#)

Segment Routing Microloop Avoidance Limitations

For IS-IS, Segment Routing Microloop Avoidance is not supported when incremental shortest path first (ISPF) is configured.

Configure Segment Routing Microloop Avoidance for IS-IS

This task describes how to enable Segment Routing Microloop Avoidance and set the Routing Information Base (RIB) update delay value for IS-IS.

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 13](#).
- 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)#
```

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	router isis <i>instance-id</i> Example: RP/0/RP0/CPU0:router(config)# router isis 1	<p>Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.</p> <p>You can change the level of routing to be performed by a particular routing instance by using the is-type router configuration command.</p>
Step 3	address-family ipv4 [unicast] Example: RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast	Specifies the IPv4 address family and enters router address family configuration mode.
Step 4	microloop avoidance segment-routing Example: RP/0/RP0/CPU0:router(config-isis-af)# microloop avoidance segment-routing	Enables Segment Routing Microloop Avoidance.
Step 5	microloop avoidance rib-update-delay <i>delay-time</i> Example: RP/0/RP0/CPU0:router(config-isis-af)# microloop avoidance rib-update-delay 3000	Specifies the amount of time the node uses the microloop avoidance policy before updating its forwarding table. The <i>delay-time</i> is in milliseconds. The range is from 1-60000. The default value is 5000.



CHAPTER 10

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.
- [Usage Guidelines and Restrictions, on page 71](#)
- [Segment Routing and LDP Interoperability, on page 72](#)
- [Configuring Mapping Server, on page 74](#)
- [Enable Mapping Advertisement, on page 76](#)
- [Enable Mapping Client, on page 78](#)

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

Segment Routing and LDP Interoperability

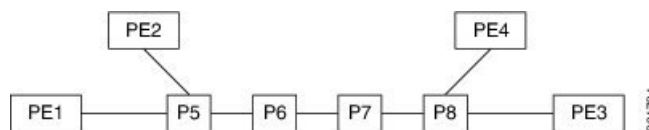
IGP provides mechanisms through which segment routing (SR) interoperate with label distribution protocol (LDP). The control plane of segment routing co-exists with LDP.

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 103.) P6 swaps 103 for 103 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.

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	segment-routing Example: RP/0/RP0/CPU0:router(config)# segment-routing	Enables segment routing.
Step 3	mapping-server Example: RP/0/RP0/CPU0:router(config-sr)# mapping-server	Enables mapping server configuration mode.
Step 4	prefix-sid-map Example: RP/0/RP0/CPU0:router(config-sr-ms)# prefix-sid-map	Enables prefix-SID mapping configuration mode. Note Two-way prefix SID can be enabled directly under IS-IS or through a mapping server.
Step 5	address-family ipv4 ipv6 Example: This example shows the address-family for ipv4: RP/0/RP0/CPU0:router(config-sr-ms-map)# address-family ipv4 This example shows the address-family for ipv6: RP/0/RP0/CPU0:router(config-sr-ms-map)# address-family ipv6	Configures address-family for IS-IS.
Step 6	<i>ip-address/prefix-length first-SID-value</i> range range	Adds SID-mapping entries in the active local mapping policy. In the configured example:

	Command or Action	Purpose
	<p>Example:</p> <pre>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</pre>	<ul style="list-style-type: none"> • Prefix 10.1.1.1/32 is assigned prefix-SID 10, prefix 10.1.1.2/32 is assigned prefix-SID 11,..., prefix 10.1.1.199/32 is assigned prefix-SID 200 • Prefix 20.1.0.0/16 is assigned prefix-SID 400, prefix 20.2.0.0/16 is assigned prefix-SID 401,..., and so on.
Step 7	Use the commit or end command.	<p>commit—Saves the configuration changes and remains within the configuration session.</p> <p>end—Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Verify information about the locally configured prefix-to-SID mappings.



Note Specify the address family for IS-IS.

```
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

Procedure

	Command or Action	Purpose
Step 1	router isis <i>instance-id</i> Example: RP/0/RP0/CPU0:router(config)# router isis 1	Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode. <ul style="list-style-type: none"> You can change the level of routing to be performed by a particular routing instance by using the is-type router configuration command.
Step 2	address-family { ipv4 ipv6 } [unicast] Example: The following is an example for ipv4 address family: 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 3	segment-routing prefix-sid-map advertise-local Example: RP/0/RP0/CPU0:router(config-isis-af)# segment-routing prefix-sid-map advertise-local	Configures IS-IS to advertise locally configured prefix-SID mappings.
Step 4	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions:

	Command or Action	Purpose
		<ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No — Exits the configuration session without committing the configuration changes. • Cancel — Remains in the configuration session, without committing the configuration changes.

Verify IS-IS prefix-SID mapping advertisement and TLV.

```
RP/0/RP0/CPU0:router# show isis database verbose
```

```
<...removed...>
```

```
SID Binding: 10.1.1.1/32 F:0 M:0 S:0 D:0 A:0 Weight:0 Range:200
SID: Start:10, Algorithm:0, R:0 N:0 P:0 E:0 V:0 L:0
SID Binding: 20.1.1.0/24 F:0 M:0 S:0 D:0 A:0 Weight:0 Range:300
SID: Start:400, Algorithm:0, R:0 N:0 P:0 E:0 V:0 L:0
```

Configure Mapping Advertisement for OSPF

Procedure

	Command or Action	Purpose
Step 1	<p>router ospf <i>process-name</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# router ospf 1</pre>	Enables OSPF routing for the specified routing instance, and places the router in router configuration mode.
Step 2	<p>segment-routing prefix-sid-map advertise-local</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-ospf)# segment-routing prefix-sid-map advertise-local</pre>	Configures OSPF to advertise locally configured prefix-SID mappings.
Step 3	Use the commit or end command.	<p>commit — Saves the configuration changes and remains within the configuration session.</p> <p>end — Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

Verify OSP prefix-SID mapping advertisement and TLV.

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

```
<...removed...>
```

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

```
SID sub-TLV: Length: 8
  Flags    : 0x60
  MTID     : 0
  Algo     : 0
  SID Index : 10
```

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 disable
RP/0/RP0/CPU0:router(config-ospf)# commit
```



CHAPTER 11

Traffic Collector Process

The Traffic Collector collects packet and byte statistics from router components such as prefix counters, tunnel counters, and the TM counter. The TM counter 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

Feature Support and Limitations

- Pcounters for IPv4 SR Prefix SIDs are supported.
- Pcounters for IPv6 SR Prefix SIDs are not supported.
- TM Pcounters increment for incoming SR-labeled and IP traffic destined for an SR Prefix SID.
- External interface support can be enabled on all Ethernet interfaces except Management, Bundle, and sub interfaces. Tunnels may not be set as external interfaces.
- Default VRF is supported. Non-default VRF is not supported.
- [Configuring Traffic Collector, on page 80](#)

- [Displaying Traffic Information, on page 81](#)

Configuring Traffic Collector

Perform these tasks to configure the traffic collector.

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
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 10	(Optional) Specifies the number of entries kept in the history database. Valid values are from 1 to 10. The default is 5. Note 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.
Step 5	statistics history-timeout value Example: RP/0/RP0/CPU0:router(config-tc)# statistics history-timeout 24	(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. Note Enter 0 to disable the history timeout. (No history is retained.)

	Command or Action	Purpose
Step 6	interface <i>type l3-interface-address</i> Example: RP/0/RP0/CPU0:router (config-tc) # interface TenGigE 0/0/0/3	Identifies interfaces that handle external traffic. Only L3 interfaces are supported for external traffic.
Step 7	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. end —Prompts user to take one of these actions: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes. • Cancel —Remains in the configuration session, without committing the configuration changes.

This completes the configuration for the traffic collector.

Displaying Traffic Information

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



Note 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
-----
Te0/0/0/3          Enabled
Te0/0/0/4          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
```

```

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
 22:58 - 22:59: Packets 9845278, Bytes: 9707444108
 22:57 - 22:58: Packets 9033554, Bytes: 8907084244
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
 22:58 - 22:59: Packets 9911734, Bytes: 9733322788
 22:57 - 22:58: Packets 9051879, Bytes: 8888945178

```

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 policy:

```

RP/0/RP0/CPU0:router# show traffic-collector counters tunnels srte_c_12_ep_6.6.6.2
detail
Tunnel: srte_c_12_ep_6.6.6.2 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 policy.