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

Notwithstanding any other warranty herein, all document files and software of these suppliers are provided “as is” with all faults. Cisco and the above-named suppliers disclaim all warranties, expressed or implied, including, without limitation, those of merchantability, fitness for a particular purpose and noninfringement or arising from a course of dealing, usage, or trade practice.

In no event shall Cisco or its suppliers be liable for any indirect, special, consequential, or incidental damages, including, without limitation, lost profits or loss or damage to data arising out of the use or inability to use this manual, even if Cisco or its suppliers have been advised of the possibility of such damages.

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

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

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

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: 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)

© 2019–2020 Cisco Systems, Inc. All rights reserved.
CONTENTS

PREFACE

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

CHAPTER 1

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

CHAPTER 2

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

CHAPTER 3

Configure Segment Routing over IPv6 (SRv6) 7
Segment Routing over IPv6 Overview 7
Configuring SRv6 IS-IS 11
Configuring SRv6 IS-IS Flexible Algorithm 12
Configuring SRv6 IS-IS TI-LFA 14
Configuring SRv6 IS-IS Microloop Avoidance 17
SRv6 Services: IPv4 L3VPN 18
SRv6 Services: L3VPN VPNv4 Active-Standby Redundancy using Port-Active Mode 22
SRv6 Services for L3VPN Active-Standby Redundancy using Port-Active Mode: Operation 23
Configure SRv6 Services L3VPN Active-Standby Redundancy using Port-Active Mode 23
Configuration Example 23
Running Configuration 24
Verification 24
Configure Segment Routing Egress Peer Engineering 69
Configure BGP Link-State 70
Example: Configuring SR-EPE and BGP-LS 71
Configure BGP Proxy Prefix SID 73

CHAPTER 8

Configure SR-TE Policies 77
SR-TE Policy Overview 77
Instantiation of an SR Policy 78
On-Demand SR Policy – SR On-Demand Next-Hop 78
Configuring SR-ODN: Examples 83
Configuring SR-ODN for EVPN-VPWS: Use Case 91
Manually Provisioned SR Policy 112
PCE-Initiated SR Policy 112
SR-TE Policy Path Types 112
Dynamic Paths 113
Optimization Objectives 113
Constraints 114
Configure SR Policy with Dynamic Path 116
Anycast SID-Aware Path Computation 117
Explicit Paths 120
Configure SR-TE Policy with Explicit Path 120
Configuring Explicit Path with Affinity Constraint Validation 122
Explicit Path with Affinity Constraint Validation for Anycast SIDs 124
Protocols 127
Path Computation Element Protocol 127
Configure the Head-End Router as PCEP PCC 127
BGP SR-TE 131
Configure Explicit BGP SR-TE 131
Traffic Steering 133
Automated Steering 133
Color-Only Automated Steering 134
Setting CO Flag 135
Address-Family Agnostic Automated Steering 135
Using Binding Segments 136
CHAPTER 16

Using Segment Routing Traffic Matrix 219
  Segment Routing Traffic Matrix 219
  Traffic Collector Process 219
  Configuring Traffic Collector 220
  Displaying Traffic Information 222

CHAPTER 17

Using Segment Routing OAM 225
  MPLS Ping and Traceroute for BGP and IGP Prefix-SID 225
  Examples: MPLS Ping, Traceroute, and Tree Trace for Prefix-SID 226
  MPLS LSP Ping and Traceroute Nil FEC Target 228
  Examples: LSP Ping and Traceroute for Nil_FEC Target 228
  Segment Routing Ping and Traceroute 230
    Segment Routing Ping 230
    Segment Routing Traceroute 232
  Segment Routing Policy Nil-FEC Ping and Traceroute 234
  Segment Routing over IPv6 OAM 236
  Segment Routing Data Plane Monitoring 237
    Configure SR DPM 240
Preface

From Release 6.1.2 onwards, Cisco introduces support for the 64-bit Linux-based IOS XR operating system. Extensive feature parity is maintained between the 32-bit and 64-bit environments. Unless explicitly marked otherwise, the contents of this document are applicable for both the environments. For more details on Cisco IOS XR 64 bit, refer to the Release Notes for Cisco ASR 9000 Series Routers, Release 6.1.2 document.

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

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

Changes to This Document

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

<table>
<thead>
<tr>
<th>Date</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>August 2019</td>
<td>Initial release of this document</td>
</tr>
<tr>
<td>March 2020</td>
<td>Republished for Release 7.0.2</td>
</tr>
</tbody>
</table>

Communications, Services, and Additional Information

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

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.
This table summarizes the new and changed feature information for the *Segment Routing Configuration Guide for Cisco ASR 9000 Aggregation Services Routers*, and lists where they are documented.

- New and Changed Segment Routing Features, on page 1

### New and Changed Segment Routing Features

#### Segment Routing Features Added or Modified in IOS XR Release 7.0.x

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Introduced/Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRv6 TI-LFA with link, node, and Shared Risk Link Groups (SRLG) protection</td>
<td>This feature is introduced.</td>
<td>Release 7.0.2</td>
<td>Configuring SRv6 IS-IS TI-LFA, on page 14</td>
</tr>
<tr>
<td>SRv6 IS-IS Microloop Avoidance</td>
<td>This feature is introduced.</td>
<td>Release 7.0.2</td>
<td>Configuring SRv6 IS-IS Microloop Avoidance, on page 17</td>
</tr>
<tr>
<td>SRv6 IS-IS Flexible Algorithm</td>
<td>This feature is introduced.</td>
<td>Release 7.0.2</td>
<td>Configuring SRv6 IS-IS Flexible Algorithm, on page 12</td>
</tr>
<tr>
<td>SRv6 Services: SRv6 Services TLV Type 5 Support</td>
<td>This feature is introduced.</td>
<td>Release 7.0.2</td>
<td>SRv6 Services: SRv6 Services TLV Type 5 Support, on page 27</td>
</tr>
<tr>
<td>SRv6 SID Information in BGP-LS Reporting</td>
<td>This feature is introduced.</td>
<td>Release 7.0.2</td>
<td>SRv6 SID Information in BGP-LS Reporting, on page 32</td>
</tr>
<tr>
<td>SRv6 OAM — SID Verification</td>
<td>This feature is introduced.</td>
<td>Release 7.0.2</td>
<td>SRv6 OAM — SID Verification, on page 32</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Introduced/Changed in Release</td>
<td>Where Documented</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>-------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Anycast SID-Aware Path Computation</td>
<td>This feature is introduced.</td>
<td>Release 7.0.1</td>
<td>Anycast SID-Aware Path Computation, on page 117</td>
</tr>
<tr>
<td>Segment Routing Tree Segment Identifier (Tree-SID)</td>
<td>This feature is introduced.</td>
<td>Release 7.0.1</td>
<td>Segment Routing Tree Segment Identifier, on page 157</td>
</tr>
<tr>
<td>SR Policy End-to-End Delay Measurement</td>
<td>This feature is introduced.</td>
<td>Release 7.0.1</td>
<td>Configure Performance Measurement, on page 181</td>
</tr>
<tr>
<td>Segment Routing Data Plane Monitoring</td>
<td>This feature is introduced.</td>
<td>Release 7.0.1</td>
<td>Segment Routing Data Plane Monitoring, on page 237</td>
</tr>
<tr>
<td>OSPF Flexible Algorithm</td>
<td>This feature is introduced.</td>
<td>Release 7.0.1</td>
<td>Configuring Flexible Algorithm, on page 163</td>
</tr>
<tr>
<td>IS-IS Flexible Algorithm TI-LFA and Microloop Avoidance</td>
<td>This feature is introduced.</td>
<td>Release 7.0.1</td>
<td>Calculation of Flexible Algorithm Path, on page 162</td>
</tr>
<tr>
<td>IS-IS Flexible Algorithm Prefix-SID Redistribution</td>
<td>This feature is introduced.</td>
<td>Release 7.0.1</td>
<td>Flexible Algorithm Prefix-SID Redistribution, on page 163</td>
</tr>
<tr>
<td>SR-TE-Services: EVPN (EVPN &amp; EVPN-VPWS multi-homing) On-Demand Next Hop</td>
<td>This feature is introduced.</td>
<td>Release 7.0.1</td>
<td>On-Demand SR Policy – SR On-Demand Next-Hop, on page 78</td>
</tr>
<tr>
<td>SR-TE Policy tracking up to 256 Link Colors for Affinity</td>
<td>This enhancement increases the number of color names you can assign on the head-end router from 32 to 256.</td>
<td>Release 7.0.1</td>
<td>Configure SR-TE Policies, on page 77</td>
</tr>
</tbody>
</table>
About Segment Routing

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**

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 on the IGP
3. Configure Segment Routing on the BGP
4. Configure the SR-TE Policy
5. Configure TI-LFA
6. Configure the Segment Routing Mapping Server
7. Collect Traffic Statistics
Segment Routing over IPv6 Overview

Segment Routing (SR) can be applied on both MPLS and IPv6 data planes. This feature extends Segment Routing support with IPv6 data plane. In an SR-MPLS enabled network, an MPLS label is used as the Segment Identifier (SID) and the source router chooses a path to the destination and encodes the path in the packet header as a stack of labels. However, in a Segment Routing over IPv6 (SRv6) network, an IPv6 address serves as the SID. The source router encodes the path to destination as an ordered list of segments (list of IPv6 addresses) in the IPv6 packet. To encode an ordered list of IPv6 addresses in an IPv6 packet, a new routing header which is an extension header is used. This new header for SRv6 is called Segment Routing Header (SRH). In an SRv6 enabled network, the active segment is indicated by the destination address of the packet, and the next segment is indicated by a pointer in the SRH.

The following list explains the fields in SRH:

- **Next header**—Identifies the type of header immediately following the SRH.
- **Hdr Ext Len (header extension length)**—The length of the SRH in 8-octet units, not including the first 8 octets.
- **Segments left**—Specifies the number of route segments remaining. That means, the number of explicitly listed intermediate nodes still to be visited before reaching the final destination.
- **Last Entry**—Contains the index (zero based) of the last element of the segment list.
• Flags—Contains 8 bits of flags.

• Tag—Tag a packet as part of a class or group of packets like packets sharing the same set of properties.

• Segment list—128-bit IPv6 addresses representing the nth segment in the segment list. The segment list encoding starts from the last segment of the SR policy (path). That means the first element of the segment list (Segment list [0]) contains the last segment of the SR policy, the second element contains the penultimate segment of the SR policy and so on.

Each node along the SRv6 packet path has a different functionality:

• Source node—A node that can generate an IPv6 packet with an SRH (an SRv6 packet), or an ingress node that can impose an SRH on an IPv6 packet.

• Transit node—A node along the path of the SRv6 packet (IPv6 packet and SRH). The transit node does not inspect the SRH. The destination address of the IPv6 packet does not correspond to the transit node.

• End point node—A node in the SRv6 domain where the SRv6 segment is terminated. The destination address of the IPv6 packet with an SRH corresponds to the end point node. The segment endpoint node executes the function bound to the SID.

<table>
<thead>
<tr>
<th>Table 1: Example of a Segment Routing Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next Header</td>
</tr>
<tr>
<td>Last Entry</td>
</tr>
<tr>
<td>Segment List[0]</td>
</tr>
<tr>
<td>(128-bit IPv6 address)</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>Segment List[n]</td>
</tr>
<tr>
<td>(128-bit IPv6 address)</td>
</tr>
<tr>
<td>Optional Type Length Value objects (variable)</td>
</tr>
</tbody>
</table>

In SRv6, a SID represents a 128-bit value, consisting of the following three parts:

• Locator: This is the first part of the SID with most significant bits and represents an address of a specific SRv6 node.

• Function: This is the portion of the SID that is local to the owner node and designates a specific SRv6 function (network instruction) that is executed locally on a particular node, specified by the locator bits.

• Args: This field is optional and represents optional arguments to the function.

The locator part can be further divided into two parts:

• SID Block: This field is the SRv6 network designator and is a fixed or known address space for an SRv6 domain. This is the most significant bit (MSB) portion of a locator subnet.

• Node Id: This field is the node designator in an SRv6 network and is the least significant bit (LSB) portion of a locator subnet.
Configuring SRv6

To enable SRv6 globally, you should first configure a locator with its prefix. The IS-IS protocol announces the locator prefix in IPv6 network and SRv6 applications (like ISIS, BGP) use it to allocate SIDs.

The following usage guidelines and restrictions apply while configuring SRv6.

• All routers in the SRv6 domain should have the same SID block (network designator) in their locator.

• The locator length should be 64-bits long.
  • The SID block portion (MSBs) cannot exceed 40 bits. If this value is less than 40 bits, user should use a pattern of zeros as a filler.
  • The Node Id portion (LSBs) cannot exceed 24 bits.

• You can configure up to 8 locators to support SRv6 Flexible Algorithm. All locators prefix must share the same SID block (first 40-bits).

Enabling SRv6 with Locator

This example shows how to globally enable SRv6 and configure locator.

Router(config)# segment-routing srv6
Router(config-srv6)# locators
Router(config-srv6-locators)# locator myLoc1
Router(config-srv6-locator)# prefix 2001:db8:0:a2::/64

Optional: Configuring Encapsulation Parameters

This example shows how to configure encapsulation parameters when configuring SRv6. These optional parameters include:

• Source Address of outer encapsulating IPv6 header: The default source address for encapsulation is one of the loopback addresses.

• Hop Limit of outer encapsulating IPv6 header: The default value for hop-limit is 255.

Router(config)# segment-routing srv6
Router(config-srv6)# encapsulation source-address 1::1
Router(config-srv6)# hop-limit 60

Optional: Enabling Syslog Logging for Locator Status Changes

This example shows how to enable the logging of locator status.

Router(config)# segment-routing srv6
Router(config-srv6)# logging locator status

Verifying SRv6 Manager

This example shows how to verify the overall SRv6 state from SRv6 Manager point of view. The output displays parameters in use, summary information, and platform specific capabilities.

Router# show segment-routing srv6 manager
Parameters:
  SRv6 Enabled: Yes
  Encapsulation:
    Source Address:
Configured: 1::1
Default: 5::5
Hop-Limit: Default
Summary:
Number of Locators: 1 (1 operational)
Number of SIDs: 4 (0 stale)
Max SIDs: 64000
OOR:
  Thresholds: Green 3200, Warning 1920
  Status: Resource Available (0 cleared, 0 warnings, 0 full)
Platform Capabilities:
  SRv6: Yes
  TILFA: Yes
  Microloop-Avoidance: No
End Functions:
  End (PSP)
  End.X (PSP)
  End.DX4
  End.DT4
Transit Functions:
  T
  T.Insert.Red
  T.Encaps.Red
Security rules:
  SEC-1
  SEC-2
  SEC-3
  SEC-4
Counters:
  CNT-1
  CNT-3
Signaled Parameters:
  Max-EL: 4
  Max-End-Pop-SRH: 4
  Max-T-Insert: 4
  Max-T-Encap: 5
  Max-End-D: 5
Max SIDs: 64000
SID Holdtime: 30 mins

Verifying SRv6 Locator

This example shows how to verify the locator configuration and its operational status.

Router# show segment-routing srv6 locator myLoc1 detail
Name       ID    Prefix          Status
---------- ---   --------------- -------
myLoc1*    5    2001:db8:0:a2::/64 Up
(*) is default
Interface:
  Name: srv6-myLoc1
  IFH: 0x000000170
  IPv6 address: 2001:db8:0:a2::/64
  Chkpt Obj ID: 0x2fc8
  Created: Apr 25 06:21:57.077 (00:03:37 ago)

Verifying SRv6 SIDs

This example shows how to verify the allocation of SRv6 local SIDs off locator(s).
Router# show segment-routing srv6 locator myLoc1 sid

<table>
<thead>
<tr>
<th>SID</th>
<th>State</th>
<th>Function</th>
<th>Context</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001:db8:0:a2:1::</td>
<td></td>
<td>End (FSP)</td>
<td>'default':1</td>
<td>sidmgr</td>
</tr>
<tr>
<td></td>
<td>InUse</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001:db8:0:a2:40::</td>
<td></td>
<td>End.DT4</td>
<td>'VRF1'</td>
<td>bgp-100</td>
</tr>
<tr>
<td></td>
<td>InUse</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001:db8:0:a2:41::</td>
<td></td>
<td>End.X (PSP)</td>
<td>[Hu0/1/0/1, Link-Local]</td>
<td>isis-srv6</td>
</tr>
<tr>
<td></td>
<td>InUse</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following example shows how to display detail information regarding an allocated SRv6 local SID.

Router# show segment-routing srv6 locator myLoc1 sid 2001:db8:0:a2:40:: detail

<table>
<thead>
<tr>
<th>SID</th>
<th>State</th>
<th>Function</th>
<th>Context</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001:db8:0:a2:40::</td>
<td></td>
<td>End.DT4</td>
<td>'VRF1'</td>
<td>bgp-100</td>
</tr>
<tr>
<td></td>
<td>InUse</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SID context: { table-id=0xe0000011 ('VRF1':IPv4/Unicast) }
Locator: myLoc1'
Allocation type: Dynamic
Created: Feb 1 14:04:02.901 (3d00h ago)

Similarly, you can display SID information across locators by using the show segment-routing sid command.

**show Commands**

You can use the following `show` commands to verify the SRv6 global and locator configuration:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show segment-routing srv6 manager</td>
<td>Displays the summary information from SRv6 manager, including platform capabilities.</td>
</tr>
<tr>
<td>show segment-routing srv6 locator <code>locator-name</code> [detail]</td>
<td>Displays the SRv6 locator information on the router.</td>
</tr>
<tr>
<td>show segment-routing srv6 locator <code>locator-name</code> sid <code>[sid-ipv6-address] [detail]</code></td>
<td>Displays the information regarding SRv6 local SID(s) allocated from a given locator.</td>
</tr>
<tr>
<td>show segment-routing srv6 sid <code>[sid-ipv6-address] [stale] [all] [detail]</code></td>
<td>Displays SIDs information across locators. By default, only “active” (i.e. non-stale) SIDs are displayed.</td>
</tr>
<tr>
<td>show route ipv6 local-srv6</td>
<td>Displays all SRv6 local-SID prefixes in IPv6 RIB.</td>
</tr>
</tbody>
</table>

**Configuring SRv6 IS-IS**

Intermediate System-to-Intermediate System (IS-IS) protocol already supports segment routing with MPLS dataplane (SR-MPLS). This feature enables extensions in IS-IS to support Segment Routing with IPv6 data.
plane (SRv6). The extensions include advertising the SRv6 capabilities of nodes and node and adjacency segments as SRv6 SIDs.

SRv6 IS-IS performs the following functionalities:

1. Interacts with SID Manager to learn local locator prefixes and announces the locator prefixes in the IGP domain.
2. Learns remote locator prefixes from other IS-IS neighbor routers and installs the learned remote locator IPv6 prefix in RIB or FIB.
3. Allocate or learn prefix SID and adjacency SIDs, create local SID entries, and advertise them in the IGP domain.

Usage Guidelines and Restrictions

The following usage guidelines and restrictions apply for SRv6 IS-IS:

- An IS-IS address-family can support either SR-MPLS or SRv6, but both at the same time is not supported.

Configuring SRv6 IS-IS

To configure SRv6 IS-IS, enable SRv6 under the IS-IS IPv6 address-family. The following examples show how to configure SRv6 IS-IS.

```
Router(config)# router isis core
Router(config-isis)# address-family ipv6 unicast
Router(config-isis-af)# segment-routing srv6
Router(config-isis-srv6)# locator myLoc1
Router(config-isis-srv6-loc)# exit
```

Configuring SRv6 IS-IS Flexible Algorithm

This feature introduces support for implementing Flexible Algorithm using IS-IS SRv6.

SRv6 Flexible Algorithm allows operators to customize IGP shortest path computation according to their own needs. An operator can assign custom SR prefix-SIDs to realize forwarding beyond link-cost-based SPF. As a result, Flexible Algorithm provides a traffic engineered path automatically computed by the IGP to any destination reachable by the IGP.

Restrictions and Usage Guidelines

The following restrictions and usage guidelines apply:

- You can configure up to 8 locators to support SRv6 Flexible Algorithm:
  - All locators prefix must share the same SID block (first 40-bits).
  - The Locator Algorithm value range is 128 to 255.

Configuring SRv6 IS-IS Flexible Algorithm

The following example shows how to configure SRv6 IS-IS Flexible Algorithm.

```
Complete the Configuring SRv6 before performing these steps.

Router(config)# segment-routing srv6
Router(config-srv6)# locators
Router(config-srv6-locators)# locator Loc1-BE // best-effort
Router(config-srv6-locator)# prefix 2001:db8::a2::/64
Router(config-srv6-locators)# exit
Router(config-srv6-locators)# locator Loc1-LL // low latency
Router(config-srv6-locator)# prefix 2001:db8:1:a2::/64
Router(config-srv6-locator)# algorithm 128
Router(config-srv6-locators)# exit
Router(config-srv6)# exit

Configuring SRv6 IS-IS

The following example shows how to configure SRv6 IS-IS.

Router(config)# router isis test-igp
Router(config-isis)# flex-algo 128
Router(config-isis-flex-algo)# exit
Router(config-isis)# address-family ipv6 unicast
Router(config-isis-if-af)# segment-routing srv6
Router(config-srv6)# locator Loc1-BE
Router(config-srv6)# exit
Router(config-isis)# locator Loc1-LL
Router(config-srv6)# exit

Enable Flexible Algorithm for Low Latency

The following example shows how to enable Flexible Algorithm for low-latency:

• IS-IS: Configure Flexible Algorithm definition with delay objective

• Performance-measurement: Configure static delay per interface

Router(config)# router isis test-igp
Router(config-isis)# flex-algo 128
Router(config-isis-flex-algo)# metric-type delay
Router(config-isis-flex-algo)# exit
Router(config-isis)# interface GigabitEthernet0/0/0
Router(config-if)# address-family ipv6 unicast
Router(config-if)# root
Router(config-isis-perf-meas)# interface GigabitEthernet0/0/0
Router(config-if)# delay-measurement
Router(config-if)# advertise-delay 100
Router(config-if)# commit

Verification

SRv6-LF1# show segment-routing srv6 locator
Mon Aug 12 20:54:15.414 EDT
Configuring SRv6 IS-IS TI-LFA

This feature introduces support for implementing Topology-Independent Loop-Free Alternate (TI-LFA) using IS-IS SRv6.

TI-LFA provides link protection in topologies where other fast reroute techniques cannot provide protection. The goal of TI-LFA is to reduce the packet loss that results while routers converge after a topology change due to a link failure. TI-LFA leverages the post-convergence path which is planned to carry the traffic and ensures link and node protection within 50 milliseconds. TI-LFA with IS-IS SR-MPLS is already supported.

TI-LFA provides link, node, and Shared Risk Link Groups (SRLG) protection in any topology.

Usage Guidelines and Restrictions

The following usage guidelines and restrictions apply:

• TI-LFA provides link protection by default. Additional tiebreaker configuration is required to enable node or SRLG protection.

• Usage guidelines for node and SRLG protection:
  • TI-LFA node protection functionality provides protection from node failures. The neighbor node is excluded during the post convergence backup path calculation.
  • Shared Risk Link Groups (SRLG) refer to situations in which links in a network share a common fiber (or a common physical attribute). These links have a shared risk: when one link fails, other links in the group might also fail. TI-LFA SRLG protection attempts to find the post-convergence
backup path that excludes the SRLG of the protected link. All local links that share any SRLG with the protecting link are excluded.

- When you enable link protection, you can also enable node protection, SRLG protection, or both, and specify a tiebreaker priority in case there are multiple LFAs.
- Valid priority values are from 1 to 255. The lower the priority value, the higher the priority of the rule. Link protection always has a lower priority than node or SRLG protection.

Configuring SRv6 IS-IS TI-LFA

The following example shows how to configure SRv6 IS-IS TI-LFA.

Note

Complete the Configuring SRv6 before performing these steps.

```
Router(config)# router isis core
Router(config-isis)# address-family ipv6 unicast
Router(config-isis-af)# segment-routing srv6
Router(config-isis-srv6)# locator locator1
Router(config-isis-srv6-loc)# exit
Router(config-isis)# interface loopback 0
Router(config-isis-if)# passive
Router(config-isis-if)# address-family ipv6 unicast
Router(config-isis-if-af)# exit
Router(config-isis)# interface bundle-ether 1201
Router(config-isis-if)# address-family ipv6 unicast
Router(config-isis-if-af)# fast-reroute per-prefix
Router(config-isis-if-af)# fast-reroute per-prefix ti-lfa
Router(config-isis-if-af)# exit
Router(config-isis)# interface bundle-ether 1301
Router(config-isis-if)# address-family ipv6 unicast
Router(config-isis-if-af)# fast-reroute per-prefix
Router(config-isis-if-af)# fast-reroute per-prefix ti-lfa
Router(config-isis-if-af)# fast-reroute per-prefix tiebreaker node-protecting index 100
Router(config-isis-if-af)# fast-reroute per-prefix tiebreaker srlg-disjoint index 200
Router(config-isis-if-af)# exit
```

Verification

This example shows how to verify the SRv6 IS-IS TI-LFA configuration using the `show isis ipv6 fast-reroute ipv6-prefix detail` command.

```
Router# show isis ipv6 fast-reroute cafe:0:0:66::/64 detail
Thu Nov 22 16:12:51.983 EST
L1 cafe:0:0:66::/64 [11/115] low priority
    via fe80::2, TenGigE0/0/0/6, SRv6-HUB6, Weight: 0
    Backup path: TI-LFA (link), via fe80::1, Bundle-Ether1201 SRv6-LF1, Weight: 0, Metric: 51
    P node: SRv6-TP8.00 [8::8], SRv6 SID: cafe:0:0:88:1:: End (PSP)
    Backup src: SRv6-HUB6.00
    P: No, TM: 51, LC: No, NP: No, D: No, SRLG: Yes
    src SRv6-HUB6.00-00, 6::6
```
This example shows how to verify the SRv6 IS-IS TI-LFA configuration using the `show route ipv6 ipv6-prefix detail` command.

Router# show route ipv6 cafe:0:0:66::/64 detail
Thu Nov 22 16:14:07.385 EST

Routing entry for cafe:0:0:66::/64
Known via "isis srv6", distance 115, metric 11, type level-1
Installed Nov 22 09:24:05.160 for 06:50:02
Routing Descriptor Blocks
  fe80::2, from 6::6, via TenGigE0/0/0/6, Protected
    Route metric is 11
    Label: None
    Tunnel ID: None
    Binding Label: None
    Extended communities count: 0
    Path id:1  Path ref count:0
    NHID:0x2000a(Ref:11)
    NHID eid:0xffffffffffffffff
    Backup path id:65
  fe80::1, from 6::6, via Bundle-Ether1201, Backup (TI-LFA)
    Repair Node(s): 8::8
    Route metric is 51
    Label: None
    Tunnel ID: None
    Binding Label: None
    Extended communities count: 0
    Path id:65  Path ref count:1
    NHID:0x2000d(Ref:11)
    NHID eid:0xffffffffffffffff
    SRv6 Transit Type: T.Insert.Red
    SRv6 SID-list { cafe:0:0:88:1:: }

This example shows how to verify the SRv6 IS-IS TI-LFA configuration using the `show cef ipv6 ipv6-prefix detail location` command.

Router# show cef ipv6 cafe:0:0:66::/64 detail location 0/0/cpu0
Thu Nov 22 17:01:58.536 EST
cafe:0:0:66::/64, version 1356, SRv6 Transit, internal 0x1000001 0x2 (ptr 0x8a4a45cc) [1],
  0x0 (0x8a46ae20), 0x0 (0x8c8f31b0)
Updated Nov 22 09:24:05.166
local adjacency fe80::2
Prefix Len 64, traffic index 0, precedence n/a, priority 2
  gateway array (0x8a2dafa0) reference count 4, flags 0x50000000 source rib (7), 0 backups
    [5 type 3 flags 0x8401 (0x8a395d58) ext 0x0 (0x0)]
  LW-LDI[type=3, refc=1, ptr-0x8a46ae20, sh-ldi=0x8a395d58]
  gateway array update type-time 1 Nov 22 09:24:05.163
LDI Update time Nov 22 09:24:05.163
LW-LDI-LS Nov 22 09:24:05.166
via fe80::2/128, TenGigE0/0/0/6, 8 dependencies, weight 0, class 0, protected [flags
  0x400]
  path-id: 0 bkup-id: 1 NHID 0x2000a [0x8a2c2fd0 0x0]
  next hop fe80::2/128
via fe80::1/128, Bundle-Ether1201, 8 dependencies, weight 0, class 0, backup (TI-LFA)
[flags 0xb00]
  path-id: 1 NHID 0x2000d [0x8c267b0 0x0]
  next hop fe80::1/128, Repair Node(s): 8::8
  local adjacency
  SRv6 T.Insert.Red SID-list {cafe:0:0:88:1::}
This example shows how to verify the SRv6 IS-IS TI-LFA configuration using the `show cef ipv6 fast-reroute-db` command.

Router# show cef ipv6 fast-reroute-db
Sun Dec 9 20:23:08.111 EST

PROTECT-FRR: per-prefix [1, 0x0, 0x0, 0x98c83270]
protect-interface: Te0/0/0/6 (0x208)
protect-next-hop: fe80::2/128
ipv6 nhinfo [0x977397d0]
Update Time Dec 9 17:29:42.427

BACKUP-FRR: per-prefix [5, 0x0, 0x2, 0x98c83350]
backup-interface: TE1201 (0x800002c)
backup-next-hop: fe80::1/128
ipv6 nhinfo [0x977396a0 protect-frr: 0x98c83270]
Update Time Dec 9 17:29:42.428

PROTECT-FRR: per-prefix [1, 0x0, 0x0, 0x98c830b0]
protect-interface: BE1201 (0x800002c)
protect-next-hop: fe80::1/128
ipv6 nhinfo [0x977396a0]
Update Time Dec 9 17:29:42.429

BACKUP-FRR: per-prefix [5, 0x0, 0x1, 0x98c83190]
backup-interface: Te0/0/0/6 (0x208)
backup-next-hop: fe80::2/128
ipv6 nhinfo [0x977397d0 protect-frr: 0x98c830b0]
Update Time Dec 9 17:29:42.429

### Configuring SRv6 IS-IS Microloop Avoidance

This feature introduces support for implementing microloop avoidance using IS-IS SRv6. 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 SRv6 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.

### Restrictions and Usage Guidelines

The following restrictions and usage guidelines apply:
The Routing Information Base (RIB) update delay value specifies the amount of time the node uses the microloop avoidance policy before updating its forwarding table. The delay-time range is from 1 to 60000 milliseconds; the default value is 5000.

**Configuring SRv6 IS-IS Microloop Avoidance**

The following example shows how to configure SRv6 IS-IS Microloop Avoidance and set the Routing Information Base (RIB) update delay value.

```
Note Complete the Configuring SRv6 before performing these steps.
```

```
Router(config)# router isis test-igp
Router(config-isis)# address-family ipv6 unicast
Router(config-isis-af)# microloop avoidance segment-routing
Router(config-isis-af)# microloop avoidance rib-update-delay 2000
Router(config-isis-af)# commit
```

**SRv6 Services: IPv4 L3VPN**

The SRv6-based IPv4 L3VPN feature enables deployment of IPv4 L3VPN over a SRv6 data plane. Traditionally, it was done over an MPLS-based system. SRv6-based L3VPN uses SRv6 Segment IDs (SIDs) for service segments instead of labels. SRv6-based L3VPN functionality interconnects multiple sites to resemble a private network service over public infrastructure. To use this feature, you must configure SRv6-base.

For this feature, BGP allocates an SRv6 SID from the locator space, configured under SRv6-base and VPNv4 address family. For more information on this, refer Segment Routing over IPv6 Overview, on page 7. The BGP SID can be allocated in the following ways:

- Per-VRF mode that provides End.DT4 support. End.DT4 represents the Endpoint with decapsulation and IPv4 table lookup.
- Per-CE mode that provides End.DX4 cross connect support. End.DX4 represents the Endpoint with decapsulation and IPv4 cross-connect.

BGP encodes the SRv6 SID in the prefix-SID attribute of the IPv4 L3VPN Network Layer Reachability Information (NLRI) and advertises it to IPv6 peering over an SRv6 network. The Ingress PE (provider edge) router encapsulates the VRF IPv4 traffic with the SRv6 VPN SID and sends it over the SRv6 network.

**Restrictions and Usage Guidelines**

- Equal-Cost Multi-path (ECMP) and Unequal Cost Multipath (UCMP) are supported.
- BGP, OSPF, Static are supported as PE-CE protocol.

**Configuring SRv6 based IPv4 L3VPN**

To enable SRv6-based L3VPN, you need to configure SRv6 under BGP and SID allocation mode. The following example shows how to configure SRv6-based L3VPN:
/*Configure SRv6 locator name under BGP Global*/
RP/0/0/CPU0:Router(config)# router bgp 100
RP/0/0/CPU0:Router(config-bgp)# bgp router-id 10.6.6.6
RP/0/0/CPU0:Router(config-bgp)# segment-routing srv6
RP/0/0/CPU0:Router(config-bgp-srv6)# locator my_locator
RP/0/0/CPU0:Router(config-bgp-srv6)# exit

/*Configure SRv6 locator under VRF All under VPNv4 AFI*/
RP/0/0/CPU0:Router(config)# router bgp 100
RP/0/0/CPU0:Router(config-bgp)# bgp router-id 10.6.6.6
RP/0/0/CPU0:Router(config-bgp)# address-family vpnv4 unicast
RP/0/0/CPU0:Router(config-bgp-af)# vrf all
RP/0/0/CPU0:Router(config-bgp-af-vrfall)# segment-routing srv6
RP/0/0/CPU0:Router(config-bgp-af-vrfall-srv6)# locator my_locator
RP/0/0/CPU0:Router(config-bgp-af-vrfall-srv6)# exit

/*Configure a VRF with per-vrf label allocation mode*/
RP/0/0/CPU0:Router(config-bgp-af)# vrf vrf1
RP/0/0/CPU0:Router(config-bgp-af-vrf)# rd 106:1
RP/0/0/CPU0:Router(config-bgp-af-vrf)# address-family ipv4 unicast
RP/0/0/CPU0:Router(config-bgp-af-vrf-af)# segment-routing srv6
RP/0/0/CPU0:Router(config-bgp-af-vrf-af-srv6)# alloc mode per-vrf
RP/0/0/CPU0:Router(config-bgp-af-vrf-af-srv6)# exit
RP/0/0/CPU0:Router(config-bgp-af-vrf)# exit
RP/0/0/CPU0:Router(config-bgp-af-vrf)# neighbor 10.1.2.2
RP/0/0/CPU0:Router(config-bgp-af-vrf-nbr)# remote-as 100
RP/0/0/CPU0:Router(config-bgp-af-vrf-nbr)# address-family ipv4 unicast

/*Configure a VRF with per-ce label allocation mode*/
RP/0/0/CPU0:Router(config-bgp-af)# vrf vrf2
RP/0/0/CPU0:Router(config-bgp-af-vrf)# rd 106:2
RP/0/0/CPU0:Router(config-bgp-af-vrf)# address-family ipv4 unicast
RP/0/0/CPU0:Router(config-bgp-af-vrf-af)# segment-routing srv6
RP/0/0/CPU0:Router(config-bgp-af-vrf-af-srv6)# alloc mode per-ce
RP/0/0/CPU0:Router(config-bgp-af-vrf-af-srv6)# exit
RP/0/0/CPU0:Router(config-bgp-af-vrf-af)# exit
RP/0/0/CPU0:Router(config-bgp-af-vrf)# neighbor 10.1.2.2
RP/0/0/CPU0:Router(config-bgp-af-vrf-nbr)# remote-as 100
RP/0/0/CPU0:Router(config-bgp-af-vrf-nbr)# address-family ipv4 unicast

Verification

The following example shows how to verify the SRv6 based L3VPN configuration using the `show segment-routing srv6 sid` command.

In this example, End.X represents Endpoint function with Layer-3 cross-connect, End.DT4 represents Endpoint with decapsulation and IPv4 table lookup, and End.DX4 represents Endpoint with decapsulation and IPv4 cross-connect.

```
RP/0/0/CPU0:SRv6-Hub6# show segment-routing srv6 sid
*** Locator: "my_locator" ***
SID  State   Function                Context                      Owner
-----  --------  ---------------------  --------------------------  --------
cafe:01:66:1:1: End (PSP) 'my_locator':1           sidmgr
        InUse   Y
cafe:01:66:40:1: End.X (PSP) [Te0/0/0/2, Link-Local] isis-srv6
        InUse   Y
```
The following examples show how to verify the SRv6 based L3VPN configuration using the `show segment-routing srv6 SID-prefix detail` command.

```
RP/0/RP0/CPU0:SRv6-Hub6# show segment-routing srv6 sid cafe:0:0:66:44:: detail
Sun Dec 9 16:52:54.015 EST
*** Locator: 'my_locator' ***
SID             Function  Context        Owner
----------------- ----------- ------------------------------ ------------------
cafe:0:0:66:44::  End.DT4    'VRF1'          bgp-100
InUse Y
SID context: { table-id=0xe0000001 ('VRF1':IPv4/Unicast) }
Locator: 'my_locator'
Allocation type: Dynamic
Created: Dec 8 16:34:32.506 (1d00h ago)

RP/0/RP0/CPU0:SRv6-Hub6# show segment-routing srv6 sid cafe:0:0:66:47:: detail
Sun Dec 9 16:54:26.073 EST
*** Locator: 'my_locator' ***
SID             Function  Context        Owner
----------------- ----------- ------------------------------ ------------------
cafe:0:0:66:47::  End.DX4    'VRF2':4          bgp-100
InUse Y
SID context: { table-id=0xe0000002 ('VRF2':IPv4/Unicast), nh-set-id=4 }
Locator: 'my_locator'
Allocation type: Dynamic
Created: Dec 9 16:49:44.714 (00:04:41 ago)
```

The following example shows how to verify the SRv6 based L3VPN configuration using the `show bgp vpnv4 unicast rd route-distinguisher/prefix` command on Egress PE.

```
RP/0/RP0/CPU0:SRv6-Hub6# sh bgp vpnv4 unicast rd 106:1 10.15.0.0/30
Wed Nov 21 16:08:44.765 EST
BGP routing table entry for 10.15.0.0/30, Route Distinguisher: 106:1
Versions:
  Process  bRIB/RIB  SendTblVer
  Speaker  2282449  2282449
  SRv6-VPN SID: cafe:0:0:66:44::/128
Last Modified: Nov 21 15:50:34.235 for 00:18:10
Paths: (2 available, best #1)
  Advertised to peers (in unique update groups):
    2::2
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    2::2
```
The following example shows how to verify the SRv6 based L3VPN configuration using the `show bgp vpnv4 unicast route-distinguisher prefix` command on Ingress PE.

```bash
RP/0/RP0/CPU0:SRv6-LF1# show bgp vpnv4 unicast rd 106:1 10.15.0.0/30
```

The following example shows how to verify the SRv6 based L3VPN configuration using the `show route vrf vrf-name/prefix detail` command.

```bash
RP/0/RP0/CPU0:SRv6-LF1# show route vrf VRF1 10.15.0.0/30 detail
```
Flow-tag: Not Set
Fwd-class: Not Set
Route Priority: RIB_PRIORITY_RECURSIVE (12) SVD Type RIB_SVD_TYPE_REMOTE
Download Priority 3, Download Version 3038384
No advertising protos.

The following example shows how to verify the SRv6 based L3VPN configuration for per-ce allocation mode using the `show bgp vrf vrf-name nexthop-set` command.

```
RP/0/RP0/CPU0:SRv6-Hub6# show bgp vrf VRF2 nexthop-set
Wed Nov 21 15:52:17.464 EST
Resilient per-CE nexthop set, ID 3
Number of nexthops 1, Label 0, Flags 0x2200
SRv6-VPN SID: cafe:0:0:66:46::/128
Nexthops:
10.1.2.2
Reference count 1,
Resilient per-CE nexthop set, ID 4
Number of nexthops 2, Label 0, Flags 0x2100
SRv6-VPN SID: cafe:0:0:66:47::/128
Nexthops:
10.1.2.2
10.2.2.2
Reference count 2,
```

The following example shows how to verify the SRv6 based L3VPN configuration using the `show cef vrf vrf-name prefix detail location line-card` command.

```
RP/0/RP0/CPU0:SRv6-LF1# show cef vrf VRF1 10.15.0.0/30 detail location 0/0/cpu0
Wed Nov 21 16:37:06.894 EST
151.1.0.0/30, version 3038384, SRv6 Transit, internal 0x5000001 0x0 (ptr 0x9ae6474c) [1],
0x0 (0x0), 0x0 (0x8c11b238)
Updated Nov 21 16:35:14.109
Prefix Len 30, traffic index 0, precedence n/a, priority 3
gateway array (0x8cd85190) reference count 1014, flags 0x2000, source rib (7), 0 backups
[1 type 3 flags 0x40441 (0x8a529798) ext 0x0 (0x0)]
LDI Update time Nov 21 14:52:59.073
Level 1 - Load distribution: 0
[0] via cafe:0:0:66::/128, recursive
via cafe:0:0:66::/128, 7 dependencies, recursive [flags 0x6000]
path-id 0 NHID 0x0 [0x8ac5b3cc 0x0]
next hop VRF - 'default', table - 0xe0800000
next hop cafe:0:0:66::/128 via cafe:0:0:66::/64
SRv6 T.Encaps.Red Sid-list [cafe:0:0:66:44::]
Load distribution: 0 (refcount 1)
Hash OK Interface Address
0 Y Bundle-Ether1201 fe80::2
```

**SRv6 Services: L3VPN VPNv4 Active-Standby Redundancy using Port-Active Mode**

The Segment Routing IPv6 (SRv6) Services: L3VPN VPNv4 Active-Standby Redundancy using Port-Active Mode feature provides all-active per-port load balancing for multihoming. The forwarding of traffic is determined based on a specific interface rather than per-flow across multiple Provider Edge routers. This feature enables efficient load-balancing and provides faster convergence. In an active-standby scenario, the
active PE router is detected using designated forwarder (DF) election by modulo calculation and the interface of the standby PE router brought down. For Modulo calculation, byte 10 of the Ethernet Segment Identifier (ESI) is used.

**Restrictions**

- This feature can only be configured for bundle interfaces.
- When an EVPN Ethernet Segment (ES) is configured with port-active load-balancing mode, you cannot configure ACs of that bundle on bridge-domains with a configured EVPN instance (EVI). EVPN Layer 2 bridging service is not compatible with port-active load-balancing.

**SRv6 Services for L3VPN Active-Standby Redundancy using Port-Active Mode: Operation**

Under port-active operational mode, EVPN Ethernet Segment (ES) routes are exchanged across BGP for the routers servicing the multihomed ES. Each PE router then builds an ordered list of the IP addresses of all PEs connected to the ES, including itself, and assigns itself an ordinal for its position in the list. The ordinals are used with the modulo calculation to determine which PE will be the Designated Forwarder (DF) for a given ES. All non-DF PEs will take the respective bundles out of service.

In the case of link or port failure, the active DF PE withdraws its ES route. This re-triggers DF election for all PEs that service the ES and a new PE is elected as DF.

**Configure SRv6 Services L3VPN Active-Standby Redundancy using Port-Active Mode**

This section describes how you can configure SRv6 services L3VPN active-standby redundancy using port-active mode under an Ethernet Segment (ES).

**Configuration Example**

```
/* Configure Ethernet Link Bundles */
Router# configure
Router(config)# interface Bundle-Ether10
Router(config-if)# ipv4 address 10.0.0.2 255.255.255.0
Router(config-if)# ipv6 address 2001:DB8::1
Router(config-if)# lacp period short
Router(config-if)# mac-address 1.2.3
Router(config-if)# bundle wait-while 0
Router(config-if)# exit

Router(config)# interface GigabitEthernet 0/2/0/5
Router(config-if)# bundle id 14 mode active
Router(config-if)# commit

/* Configure load balancing. */
Router# configure
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether10
Router(config-evpn-ac)# ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 11.11.11.11.11.11.11.14
Router(config-evpn-ac-es)# load-balancing-mode port-active
```
Running Configuration

interface Bundle-Ether14
ipv4 address 14.0.0.2 255.255.255.0
ipv6 address 14::2/64
lacp period short
mac-address 1.2.3
bundle wait-while 0

interface GigabitEthernet0/2/0/5
bundle id 14 mode active

evpn
interface Bundle-Ether14
   ethernet-segment
      identifier type 0 11.11.11.11.11.11.11.11.14
      load-balancing-mode port-active

Verification

Verify the SRv6 services L3VPN active-standby redundancy using port-active mode configuration.

/* Verify ethernet-segment details on active DF router */
Router# show evpn ethernet-segment interface Bundle-Ether14 detail
Ethernet Segment Id Interface Nexthops
---------------------------------------- ---------------------
0011.1111.1111.1111.1114 BE14 192.168.0.2 192.168.0.3

   ES to BGP Gates : Ready
   ES to L2FIB Gates : Ready
   Main port :
      Interface name : Bundle-Ether14
      Interface MAC : 0001.0002.0003
      IfHandle : 0x0000041d0
State: Up
Redundancy: Not Defined
ESI type: 0
Value: 11.1111.1111.1111.1114
ES Import RT: 1111.1111.1111 (from ESI)
Source MAC: 0000.0000.0000 (N/A)
Topology:
  Operational: MH
  Configured: Port-Active
Service Carving: Auto-selection
Multicast: Disabled
Peering Details:
  192.168.0.2 [MOD:P:00]
  192.168.0.3 [MOD:P:00]
Service Carving Results:
  Forwarders: 0
  Permanent: 0
  Elected: 0
  Not Elected: 0
MAC Flushing mode: STP-TCN
Peering timer: 3 sec [not running]
Recovery timer: 30 sec [not running]
Carving timer: 0 sec [not running]
Local SHG label: None
Remote SHG labels: 0

/* Verify bundle Ethernet configuration on active DF router */
Router# show bundle bundle-ether 14
Bundle-Ether14
Status: Up
Local links (active/standby/configured): 1 / 0 / 1
Local bandwidth (effective/available): 1000000 (1000000) kbps
MAC address (source): 0001.0002.0003 (Configured)
Inter-chassis link: No
Minimum active links / bandwidth: 1 / 1 kbps
Maximum active links: 64
Wait while timer: Off
Load balancing:
  Link order signaling: Not configured
  Hash type: Default
  Locality threshold: None
  LACP: Operational
  Flap suppression timer: Off
  Cisco extensions: Disabled
  Non-revertive: Disabled
mLACP: Not configured
IPv4 BFD: Not configured
IPv6 BFD: Not configured

<table>
<thead>
<tr>
<th>Port</th>
<th>Device</th>
<th>State</th>
<th>Port ID</th>
<th>B/W, kbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gi0/2/0/5</td>
<td>Local</td>
<td>Active</td>
<td>0x8000, 0x0003</td>
<td>1000000</td>
</tr>
</tbody>
</table>

Link is Active

/* Verify ethernet-segment details on standby DF router */
Router# show evpn ethernet-segment interface bundle-ether 10 detail

Router# show evpn ethernet-segment interface Bundle-Ether24 detail
Ethernet Segment Id Interface Nexthops
----------------------------------------------------------------------
Segment Routing Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 7.0.x

Verification

0011.1111.1111.1111.1114 BE24

ES to BGP Gates : Ready
ES to L2FIB Gates : Ready
Main port:
   Interface name : Bundle-Ether24
   Interface MAC : 0001.0002.0003
   IfHandle : 0x0000041b0
   State : Standby
   Redundancy : Not Defined
ESI type : 0
   Value : 11.1111.1111.1111.1114
ES Import RT : 1111.1111.1111 (from ESI)
Source MAC : 0000.0000.0000 (N/A)
Toplogy:
   Operational : MH
   Configured : Port-Active
Service Carving : Auto-selection
Multicast : Disabled
Peering Details:
   192.168.0.2 [MOD:P:00]
   192.168.0.3 [MOD:P:00]
Service Carving Results:
   Forwarders : 0
   Permanent : 0
   Elected : 0
   Not Elected : 0
MAC Flushing mode : STP-TCN
Peering timer : 3 sec [not running]
Recovery timer : 30 sec [not running]
Carving timer : 0 sec [not running]
Local SHG label : None
Remote SHG labels : 0

/* Verify bundle configuration on standby DF router */
Router# show bundle bundle-ether 24

Bundle-Ether24
Status: LACP OOS (out of service)
Local links <active/standby/configured>: 0 / 1 / 1
Local bandwidth <effective/available>: 0 (0) kbps
MAC address (source): 0001.0002.0003 (Configured)
Inter-chassis link: No
Minimum active links / bandwidth: 1 / 1 kbps
Maximum active links: 64
Wait while timer: Off
Load balancing:
   Link order signaling: Not configured
   Hash type: Default
   Locality threshold: None
LACP:
   Flap suppression timer: Off
   Cisco extensions: Disabled
   Non-revertive: Disabled
mLACP: Not configured
IPv4 BFD: Not configured
IPv6 BFD: Not configured

<table>
<thead>
<tr>
<th>Port</th>
<th>Device State</th>
<th>Port ID</th>
<th>B/W, kbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gi0/0/0/4</td>
<td>Local Standby</td>
<td>0x8000, 0x0002</td>
<td>100000</td>
</tr>
</tbody>
</table>

Link is in standby due to bundle out of service state
SRv6 Services: SRv6 Services TLV Type 5 Support

IOS XR 6.6.1 supports IETF draft draft-dawra-idr-srv6-vpn-04, in which the SRv6-VPN SID TLV (TLV Type 4) carries the SRv6 Service SID information. This SID TLV is inconsistent with the SRv6 SID Structure.

In IOS XR 7.0.2 and later releases, the implementation is compliant with draft-ietf-bess-srv6-services-00, which defines a new SRv6 Services TLV (TLV Type 5/6) and SRv6 SID Structure Sub-Sub-TLV to address this inconsistency.

SRv6 SID Structure Sub-Sub-TLV describes mechanisms for signaling of the SRv6 Service SID by transposing a variable part of the SRv6 SID value (function and/or the argument parts) and carrying them in existing label fields to achieve more efficient packing of VPN and EVPNN prefix NLRI s in BGP update messages.

In order to allow backward compatibility between the newer software and the older software, use the segment-routing srv6 prefix-sid-type4 command in Router BGP Neighbor VPNv4 Address-Family configuration mode to advertise BGP VPNv4 NLRI s in TLV Type 4 format. The newer software can receive either TLV Type 4 or TLV Type 5 formats.

The following configuration shows how to enable the advertisement of BGP VPNv4 NLRI s in TLV Type 4 format:

```
RP/0/RSP0/CPU0:Rtr-a(config)# router bgp 65000
RP/0/RSP0/CPU0:Rtr-a(config-bgp)# neighbor 6::6
RP/0/RSP0/CPU0:Rtr-a(config-bgp-nbr)# address-family vpnv4 unicast
RP/0/RSP0/CPU0:Rtr-a(config-bgp-nbr-af)# segment-routing srv6 prefix-sid-type4
RP/0/RSP0/CPU0:Rtr-a(config-bgp-nbr-af)#
```

SRv6/MPLS L3 Service Interworking Gateway

SRv6/MPLS L3 Service Interworking Gateway enables you to extend L3 services between MPLS and SRv6 domains by providing service continuity on the control plane and data plane.

This feature allows for SRv6 L3VPN domains to interwork with existing MPLS L3VPN domains. The feature also allows a way to migrate from MPLS L3VPN to SRv6 L3VPN.

The SRv6/MPLS L3 Service Interworking Gateway provides both transport and service termination at the gateway node. The gateway generates both SRv6 VPN SIDs and MPLS VPN labels for all prefixes under the VRF configured for re-origination. The gateway supports traffic forwarding from MPLS domain to SRv6 domain by popping the MPLS VPN label, looking up the destination prefix, and pushing the appropriate SRv6 encapsulation. From SRv6 domain to MPLS domain, the gateway removes the outer IPv6 header, looks up the destination prefix, and pushes the VPN and next-hop MPLS labels.

VRFs on the gateway node are configured with 2 sets of route targets (RTs):

- MPLS L3VPN RTs
- SRv6 L3VPN RTs (called stitching RTs)

The gateway performs the following actions:

- Imports service routes received from one domain (MPLS or SRv6)
- Re-advertises exported service routes to the other domain (next-hop-self)
• Stitches the service on the data plane (End.DT4/H.Encaps.Red ↔ service label)

SRv6/MPLS L3 Service Interworking Gateway Scenarios

The following scenario is used to describe the gateway functionality:

• Node 1 is an L3VPN PE in the MPLS domain with an SR prefix SID label of 16001 for its Loopback interface 1.1.1.1/32.

• Node 2 is the SRv6/MPLS L3 Service Interworking Gateway. In the MPLS domain, it has an SR prefix SID label of 16002 for its Loopback interface 1.1.1.2/32. In the SRv6 domain, it has an SRv6 locator of B:0::2::64 and Loopback interface B:0:0:2::/128.

• Node 3 is an L3VPN PE in the SRv6 domain with SRv6 locator of B:0:0:3::64 and Loopback interface B:0:0:3::1/128.

Scenario 1: SRv6-to-MPLS Control-Plane Direction/MPLS-to-SRv6 Data-Plane Direction

The figure below describes the associated control-plane behaviors in the SRv6-to-MPLS direction for traffic in the MPLS-to-SRv6 data-plane direction.

A. Node 3 advertises a BGP L3VPN update for prefix B.0.0.0/8 with RD corresponding to VRFA, including the SRv6 VPN SID (B:0:0:3::V9:) assigned to this VRF, in the SRv6 domain.
SRv6 End.DT4 function value "V9" is not a valid hex number, however it is used for illustration purposes to remind you of its connection to a VRF.

B. Node 2 (gateway) imports the BGP L3VPN update and programs its FIB:
   • MPLS label 24010 is allocated for VRFA
   • Prefix B.0.0.0/8 is programmed with an "SR Headend Behavior with Reduced Encapsulation in an SR Policy" function (H.Encaps.Red) of B:0:0:3:V9::

The gateway follows per-VRF label and per-VRF SID allocation methods.

C. Node 2 re-originates a BGP L3VPN update for the same prefix, including the MPLS VPN label (24010) allocated for the VRF, in the MPLS domain.

D. Site A sends traffic to an IPv4 prefix (B.B.B.B) of Site B

E. Node 1 encapsulates incoming traffic with the MPLS VPN label (24010) and the prefix SID MPLS label (16002) of the BGP next-hop (Node 2).

F. Node 2 performs the following actions:
   • Pops the MPLS VPN label and looks up the destination prefix
   • Encapsulates the payload in an outer IPv6 header with destination address (DA) equal to the H.Encaps.Red function (B:0:0:3:V9::)

G. Node 3 removes the outer IPv6 header, looks up the payload destination address (B.B.B.B), and forwards to Site B.

Scenario 2: MPLS-to-SRv6 Control-Plane Direction/SRv6-to-MPLS Data-Plane Direction

The figure below describes the associated control-plane behaviors in the MPLS-to-SRv6 direction for traffic in the SRv6-to-MPLS data-plane direction.
A. Node 1 advertises a BGP L3VPN update for prefix A.0.0.0/8 with RD corresponding to VRFA, including the MPLS VPN label (24055) assigned to this VRF, in the MPLS domain.

B. Node 2 (gateway) imports the BGP L3VPN update and programs its FIB:
   - Prefix A.0.0.0/8 is programmed to impose an MPLS VPN label (24055) and the prefix SID MPLS label (16001) of the BGP next-hop (Node 1)
   - "Endpoint with decapsulation and IPv4 table lookup" function (End.DT4) of B:0:0:2:V8:: is allocated to VRFA

   **Note**
   SRv6 End.DT4 function value "V8" is not a valid hex number, however it is used for illustration purposes to remind you of its connection to a VRF.

   **Note**
   The gateway follows per-VRF label and per-VRF SID allocation methods.

C. Node 2 re-originates a BGP L3VPN update for the same prefix, including the End.DT4 function (B:0:0:2:V8::) allocated for the VRF, in the SRv6 domain.

D. Site B sends traffic to an IPv4 prefix (A.A.A.A) of Site A.

E. Node 3 Encapsulates the payload in an outer IPv6 header with destination address (DA) equal to the End.DT4 function (B:0:0:2:V8::).

F. Node 2 performs the following actions:
   - Removes the outer IPv6 header and looks up the destination prefix
   - Pushes the MPLS VPN label (24055) and the prefix SID MPLS label (16001) of the BGP next-hop (Node 1)
G. Node 1 pops the MPLS VPN label, looks up the payload destination address (A.A.A.A), and forwards to Site A.

Example

Leveraging the topology described in the above use-case, this example shows the SRv6/MPLS L3 Service Interworking Gateway configuration required at Node 2.

The following configuration shows how to enable SRv6 with locator and configure encapsulation parameters:

```bash
segment-routing
srv6
encapsulation
   source-address b:0:0:2::1
locators
   locator LOC1
      prefix b:0:0:2::/64
```

The following configuration shows how to configure a VPNv4 VRF with the following route targets (RTs):

- 1111:1, RT used for MPLS L3VPN
- 2222:1, RT used for SRv6 L3VPN (stitching RT)

```bash
vrf ACME
   address-family ipv4 unicast
      import route-target
         1111:1
         2222:1 stitching
      export route-target
         1111:1
         2222:21 stitching
```

The following configuration shows how to configure SRv6/SRv6 VPNs under BGP:

```bash
router bgp 100
   segment-routing srv6
      locator LOC1
   neighbor 1.1.1.1
      address-family vpnv4 unicast
         import re-originate stitching-rt
         route-reflector-client
         advertise vpnv4 unicast re-originated
   neighbor b:0:0:3::1
      address-family vpnv4 unicast
         import stitching-rt re-originate
         route-reflector-client
         encapsulation-type srv6
         advertise vpnv4 unicast re-originated stitching-rt
   vrf ACME
      address-family ipv4 unicast
```

SRv6 SID Information in BGP-LS Reporting

BGP Link-State (BGP-LS) is used to report the topology of the domain using nodes, links, and prefixes. This feature adds the capability to report SRv6 Segment Identifier (SID) Network Layer Reachability Information (NLRI).

The following NLRI has been added to the BGP-LS protocol to support SRv6:

- Node NLRI: SRv6 Capabilities, SRv6 MSD types
- Link NLRI: End.X, LAN End.X, and SRv6 MSD types
- Prefix NLRI: SRv6 Locator
- SRv6 SID NLRI (for SIDs associated with the node): Endpoint Function, BGP-EPE Peer Node/Set

This example shows how to distribute IS-IS SRv6 link-state data using BGP-LS:

```
Router(config)# router isis 200
Router(config-isis)# distribute link-state instance-id 200
```

When you distribute IS-IS SRv6 link-state data, SRv6 ping or traceroute operations using the `use-srv6-op-sid` keyword can retrieve and use the corresponding END.OP SID automatically. See SRv6 OAM — SID Verification, on page 32 for details about SRv6 ping and traceroute.

---

SRv6 OAM — SID Verification

This feature provides enhanced Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6).

Existing OAM mechanisms to ping and trace a remote IPv6 prefix, along the shortest path, continue to work without any modification in an SRv6 network.

However, classic IPv6 OAM cannot be used to ping or trace a remote SRv6 SID function. This feature augments ping and traceroute operations to target remote SRv6 SIDs. An SRv6-enabled router now allocates a new SRv6 OAM SID known as END.OP (OAM Endpoint with Punt).

Use the following commands to performs SRv6 ping and traceroute:

```
ping B:k:F:: [use-srv6-op-sid [ end.op-sid-value]]
traceroute B:k:F:: [use-srv6-op-sid [ end.op-sid-value]]
```

Where `B:k:F::` is the target SID at node `k` with locator block `B` and function `F`. 
Ping/Traceroute to SID Without OAM SID

The user can issue ping or traceroute to an SRv6 SID using the classic CLI. A ping or traceroute to an SRv6 SID does not require the user to enter the “use-end-op” keyword when pinging or tracing a SID function. In this case, and as usual, the packet is pre-routed as an ICMP echo request or UDP packet.

Ping/Traceroute to SID With OAM SID

When ping or traceroute operations include the `use-srv6-op-sid` keyword, the packet is pre-routed with END.OP SID as Destination Address (DA) and the target SID in the SRH.

Note

The END.OP SID value is an optional 128 bit value in IPv6 address format. See END.OP SID Derivation below for details on this value.

At the target node, the END.OP SID forces the punt of the packet to the OAM process, which verifies that the next SID is local and valid. If the next SID received by the target node is a local valid address that is not a SID, the target node still replies to indicate ping success. The ping reply contains a subtype to indicate the target was a SID or a local address.

A target remote SID include the following:

- END
- END.DT4/END.DX4 (used by L3 Services over SRv6)

END.OP SID Derivation

The ingress node can automatically derive the END.OP SID associated with a specified target SID by leveraging the IGP topology database in that node. The database will contain END.OP SIDs from remote nodes.

An END.OP SID associated with a locator will be advertised by IS-IS within an IGP domain in an area/level, which is added to the topology database. However, END.OP SIDs are not redistributed by IS-IS across IGP domains or across different area/level within an IGP domain. In this case, the topology database in a node contains END.OP SIDs only from the nodes within the same IGP domain in an area/level. An END.OP SID cannot be determined automatically if the specified target SID is external to the domain. For target SIDs across IGP domains or across different area/level within an IGP domain, the `end.op-sid-value` must be explicitly provided.

If `end.op-sid-value` is not provided and the END.OP SID cannot be automatically derived, an error is displayed prompting the user to provide the `end.op-sid-value`.

Configuration Examples

The following example shows using ping to a SID without OAM SID.

```
Router# ping cafe:0:0:a3:40::
Wed Jul 24 19:24:50.812 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to cafe:0:0:a3:40::, timeout is 2 seconds:
!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
```

The following example shows using ping to a SID with an OAM SID. Note that the output shows "S" to indicate that this is a response from a SID target.

```
```
The following example shows using ping to a SID with an explicit OAM SID. Note that the output shows "S" to indicate that this is a response from a SID target.

Router# ping cafe:0:0:a3:40:: use-srv6-op-sid cafe:0:0:a3:11::
Wed Jul 24 19:24:50.812 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to cafe:0:0:a3:40::, timeout is 2 seconds:
SSSSS
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms

The following example shows using traceroute to a SID without OAM SID.

Router# traceroute cafe:0:0:a3:1::
Wed Jul 24 19:40:19.192 UTC
Type escape sequence to abort.
Tracing the route to cafe:0:0:a3:1::
1 2001::1:1:1:1 2 msec 2 msec 2 msec
2 2001:10:10:13::2 2 msec 2 msec 2 msec

The following example shows using traceroute to a SID with OAM SID.

Router# traceroute cafe:0:0:a3:40:: use-srv6-op-sid
Type escape sequence to abort.
Tracing the route to cafe:0:0:a3:40::
1 2001::1:1:1:1 [IP tunnel: DA=cafe:0:0:a3:11:: SRH Stack 0 =(cafe:0:0:a3:40:: ,SL=1) ] 2 msec
2 2001::33:33:33:33 [IP tunnel: DA=cafe:0:0:a3:11:: SRH Stack 0 =(cafe:0:0:a3:40:: ,SL=1) ] 3 msec

The following example shows using traceroute to a SID with an explicit OAM SID.

Router# traceroute cafe:0:0:a3:40:: use-srv6-op-sid cafe:0:0:a3:11::
Type escape sequence to abort.
Tracing the route to cafe:0:0:a3:40::
1 2001::1:1:1:1 [IP tunnel: DA=cafe:0:0:a3:11:: SRH Stack 0 =(cafe:0:0:a3:40:: ,SL=1) ] 2 msec
2 2001::33:33:33:33 [IP tunnel: DA=cafe:0:0:a3:11:: SRH Stack 0 =(cafe:0:0:a3:40:: ,SL=1) ] 3 msec
Chapter 4

Configure Segment Routing Global Block and Segment Routing Local Block

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

- About the Segment Routing Global Block, on page 35
- About the Segment Routing Local Block, on page 36
- Setup a Non-Default Segment Routing Global Block Range, on page 37
- Setup a Non-Default Segment Routing Local Block Range, on page 38

About the Segment Routing Global Block

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

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

The default SRGB range is from 16000 to 23999.

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

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

To keep the segment routing configuration simple and to make it easier to troubleshoot segment routing issues, we recommend that you use the default SRGB range on each node in the domain. However, there are instances when you might need to define a different range. For example:
The nodes of another vendor support a label range that is different from the default SRGB, and you want to use the same SRGB on all nodes.

- The default range is too small.

- To specify separate SRGBs for IS-IS and OSPF protocols, as long as the ranges do not overlap.

Restrictions:

- In Cisco IOS XR release 6.2.x and earlier, LSD label values 0-15999 are reserved. In Cisco IOS XR release 6.3.1 and later, LSD label values 0-14999 are reserved.

- In Cisco IOS XR release 6.2.x and earlier, the maximum SRGB size is 65536. In Cisco IOS XR release 6.3.1 and later, the maximum SRGB size is 262,143.

- The SRGB upper bound cannot exceed the platform's capability.

Note

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

The SRGB can be disabled if SR is not used.

About the Segment Routing Local Block

The Segment Routing Local Block (SRLB) is a range of label values preserved for the manual allocation of adjacency segment identifiers (adj-SIDs), Layer 2 adj-SIDs, and binding SIDs (BSIDs). These labels are locally significant and are only valid on the nodes that allocate the labels. The default SRLB range is from 15000 to 15999.

Note

Adjacency SIDs, Layer 2 adjacency SIDs, and binding SIDs (BSIDs) that are not manually allocated using the SRLB will be dynamically allocated from the dynamic label range.

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

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

- The default range is too small.

When you define a new 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). The previous SRLB range (active) will continue to be in use until one of the following occurs:

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

- Use the `clear segment-routing local-block discrepancy all` command to clear the label conflicts.
Restrictions:

- LSD label values 0-14999 are reserved.
- The SRLB size cannot be more than 262,143.
- The SRLB upper bound cannot exceed the platform's capability.

Note

The SRLB (Segment Routing Local Block) inconsistency and allocation failure error is observed when a non-default values of SRLB and SRGB (Segment Routing Global Block) are configured and a commit-replace is followed by configuration re-application. This issue impacts data forwarding as the SR labels are not properly programmed.

To prevent the issue, use the `clear segment-routing local-block discrepancy all` command to clear the label conflicts.

The SRLB can be disabled if SR is not used.

Setup a Non-Default Segment Routing Global Block Range

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

SUMMARY STEPS

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

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RSP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>`[router {isis instance-id</td>
<td>ospf process_name}]`</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RSP0/CPU0:router(config)# router isis 1</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>segment-routing global-block starting_value ending_value</code></td>
<td>Enter the lowest value that you want the SRGB range to include as the starting value. Enter the highest value that you want the SRGB range to include as the ending value.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RSP0/CPU0:router(config-isis)# segment-routing</code></td>
<td></td>
</tr>
</tbody>
</table>
Configure Segment Routing Global Block and Segment Routing Local Block

Setup a Non-Default Segment Routing Local Block Range

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

**SUMMARY STEPS**

1. configure
2. segment-routing local-block *starting_value* ending_value
3. Use the commit or end command.

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
</tr>
</tbody>
</table>

Verify the SRGB configuration:

```
RP/0/RSP0/CPU0:router# show mpls label table detail
Table Label Owner State Rewrite
------ ------- ------------------------------- ------ -------
0 18000 ISIS(A):1 InUse No
Lbl-blk SRGB, vers:0, (start_label=18000, size=2000)
0 24000 ISIS(A):1 InUse Yes
(SR Adj Segment IPv4, vers:0, index=1, type=0, intf=G10/0/0/0, nh=10.0.0.2)
```

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.

**SUMMARY STEPS**

1. configure
2. segment-routing local-block *starting_value* ending_value
3. Use the commit or end command.

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
</tr>
</tbody>
</table>
Configure Segment Routing Global Block and Segment Routing Local Block

Setup a Non-Default Segment Routing Local Block Range

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>configure</code></td>
<td></td>
</tr>
</tbody>
</table>

**Step 2**

**segment-routing local-block** *starting_value ending_value*

**Example:**

```
RP/0/RSP0/CPU0:router(config)# segment-routing local-block 30000 30999
```

**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:

- **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 SRLB configuration:

```
RP/0/RSP0/CPU0:router# show mpls label table detail
Table Label Owner State Rewrite
-------- ------ ------------------------------- ------ -------
<...snip...>
0 13 LSD(A) InUse Yes
0 30000 LSD(A) InUse No
(Lbl-blk SRLB, vers:0, (start_label=30000, size=1000, app_notify=0)
0 3002 Static(A) InUse Yes
```

Display and resolve any SRLB inconsistencies:

```
RP/0/RSP0/CPU0:router# show segment-routing local-block inconsistencies
Tue Aug 15 13:53:30.555 EDT
SRLB inconsistencies range: Start/End: 30000/30009

RP/0/RSP0/CPU0:router# show mpls lsd private | i SRLB
Tue Aug 15 13:53:50.874 EDT
SRLB Lbl Mgr:
  Current Active SRLB block = [15000, 15999]
  Configured Pending SRLB block = [30000, 30009]

RP/0/RSP0/CPU0:router# clear segment-routing local-block discrepancy all
Tue Aug 15 13:59:46.897 EDT

RP/0/RSP0/CPU0:router# show mpls lsd private | i SRLB
Tue Aug 15 13:59:55.370 EDT
SRLB Lbl Mgr:
```
Current Active SRLB block = [30000, 30009]
Configured Pending SRLB block = [0, 0]

RP/0/RSP0/CPU0:router# show mpls label table detail private
Tue Aug 15 14:00:26.023 EDT
Table Label Owner State Rewrite
----- ------- ------------------------------- ------ -------
0 0 LSD(A) InUse Yes
0 1 LSD(A) InUse Yes
0 2 LSD(A) InUse Yes
0 13 LSD(A) InUse Yes
0 30000 LSD(A) InUse No
(Lbl-blk SRLB, vers:0, (start_label=30000, size=1000, app_notify=0)

What to do next

Configure adjacency SIDs and enable segment routing.
CHAPTER 5

Configure Segment Routing for IS-IS Protocol


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

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

Note
- Enabling Segment Routing for IS-IS Protocol, on page 41
- Configuring a Prefix-SID on the IS-IS Enabled Loopback Interface, on page 44
- Configuring an Adjacency SID, on page 47
- Configuring Bandwidth-Based Local UCMP, on page 53
- IS-IS Prefix Attributes for Extended IPv4 and IPv6 Reachability, on page 54
- IS-IS Multi-Domain Prefix SID and Domain Stitching: Example, on page 57

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
- Multiple IS-IS instances on the same loopback interface for domain border nodes
- 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.

SUMMARY STEPS

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

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> router isis instance-id</td>
<td>Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.</td>
</tr>
<tr>
<td><strong>Note</strong> You can change the level of routing to be performed by a particular routing instance by using the is-type router configuration command.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> address-family { ipv4</td>
<td>ipv6 } [ unicast ]</td>
</tr>
<tr>
<td><strong>Step 4</strong> metric-style wide [ level { 1</td>
<td>2 } ]</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Enables RSVP traffic engineering functionality.</td>
</tr>
<tr>
<td><em>mpls traffic-eng level</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-isis-af)# mpls traffic-eng level-2-only</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Sets the traffic engineering loopback interface.</td>
</tr>
<tr>
<td><em>mpls traffic-eng router-id interface</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-isis-af)# mpls traffic-eng router-id Loopback0</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Configures router ID for each address-family (ipv4/ipv6).</td>
</tr>
<tr>
<td><em>router-id loopback loopback interface used for prefix-sid</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/(config-isis-af)#router-id Loopback0</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Segment routing is enabled by the following actions:</td>
</tr>
<tr>
<td><em>segment-routing mpls</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-isis-af)# segment-routing mpls</code></td>
<td></td>
</tr>
<tr>
<td>Segment routing is enabled by the following actions:</td>
<td></td>
</tr>
<tr>
<td>• MPLS forwarding is enabled on all interfaces where IS-IS is active.</td>
<td></td>
</tr>
<tr>
<td>• All known prefix-SIDs in the forwarding plain are programmed, with the prefix-SIDs advertised by remote routers or learned through local or remote mapping server.</td>
<td></td>
</tr>
<tr>
<td>• The prefix-SIDs locally configured are advertised.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>Enables traffic engineering functionality on the node. The node advertises the traffic engineering link attributes in IGP which populates the traffic engineering database (TED) on the head-end. The RSVP-TE head-end requires the TED to calculate and validate the path of the RSVP-TE policy.</td>
</tr>
<tr>
<td><em>exit</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-isis-af)# exit</code></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-isis-af)# exit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td></td>
</tr>
<tr>
<td><em>mpls traffic-eng</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td></td>
</tr>
<tr>
<td>Use the <em>commit</em> or <em>end</em> command.</td>
<td></td>
</tr>
<tr>
<td><em>commit</em> — Saves the configuration changes and remains within the configuration session.</td>
<td></td>
</tr>
<tr>
<td><em>end</em> — Prompts user to take one of these actions:</td>
<td></td>
</tr>
<tr>
<td>• <em>Yes</em> — Saves configuration changes and exits the configuration session.</td>
<td></td>
</tr>
<tr>
<td>• <em>No</em> — Exits the configuration session without committing the configuration changes.</td>
<td></td>
</tr>
</tbody>
</table>
Configure the prefix SID.

### 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.

Strict-SPF SIDs are used to forward traffic strictly along the SPF path. Strict-SPF SIDs are not forwarded to SR-TE policies. IS-IS advertises the SR Algorithm sub Type Length Value (TLV) (in the SRRouter Capability SubTLV) to include both algorithm 0 (SPF) and algorithm 1 (Strict-SPF). When the IS-IS area or level is Strict-SPF TE-capable, Strict-SPF SIDs are used to build the SR-TE Strict-SPF policies. Strict-SPF SIDs are also used to program the backup paths for prefixes, node SIDs, and adjacency SIDs.

The same SRGB is used for both regular SIDs and strict-SPF SIDs.

---

### Note

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

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

### Before you begin

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

### SUMMARY STEPS

1. configure
2. router isis instance-id
3. interface Loopback instance
4. address-family [ ipv4 | ipv6 ] [ unicast ]
5. prefix-sid [strict-spf | algorithm algorithm-number] [index SID-index | absolute SID-value] [n-flag-clear] [explicit-null]
6. Use the `commit` or `end` command.

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | `configure`  
Example:  
RP/0/RSP0/CPU0:router# configure | Enters global configuration mode. |
| **Step 2** | `router isis instance-id`  
Example:  
RP/0/RSP0/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 instance`  
Example:  
RP/0/RSP0/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/RSP0/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 [strict-spf | algorithm algorithm-number]`  
Example:  
RP/0/RSP0/CPU0:router(config-isis-if-af)# prefix-sid index 1001  
RP/0/RSP0/CPU0:router(config-isis-if-af)# prefix-sid strict-spf index 101  
RP/0/RSP0/CPU0:router(config-isis-if-af)# prefix-sid absolute 17001 | Configures the prefix-SID index or absolute value for the interface.  
Specify `strict-spf` to configure the prefix-SID to use the SPF path instead of the SR-TE policy.  
Specify `algorithm algorithm-number` to configure SR Flexible Algorithm. See Enabling Segment Routing Flexible Algorithm, on page 161.  
Specify `index SID-index` for each node to create a prefix SID based on the lower boundary of the SRGB + the index.  
Specify `absolute SID-value` for each node to create a specific prefix SID within the SRGB.  
By default, the n-flag is set on the prefix-SID, indicating that it is a node SID. For specific prefix-SID (for example, Anycast prefix-SID), enter the `n-flag-clear` keyword.  
IS-IS does not set the N flag in the prefix-SID sub Type Length Value (TLV). |
Configure a Prefix-SID on the IS-IS Enabled Loopback Interface

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> IS-IS does not advertise separate explicit-NULL or flags for regular SIDs and strict-SPF SIDs. The settings in the regular SID are used if the settings are different.</td>
</tr>
</tbody>
</table>

**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:

- **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/RSP0/CPU0:router# show isis database verbose
IS-IS 1 (Level-2) Link State Database
LSPID LSP Seq Num LSP Checksum LSP Holdtime ATT/P/OL
router.00-00 * 0x0000039b 0xfc27 1079 0/0/0
Area Address: 49.0001
NLPID: 0xcc
NLPID: 0x8e
MT: Standard (IPv4 Unicast)
MT: IPv6 Unicast 0/0/0
Hostname: router
IP Address: 10.0.0.1
IPv6 Address: 2001:0db8:1234::0a00:0001
Router Cap: 10.0.0.1, D:0, S:0
Segment Routing: I:1 V:1, SRGB Base: 16000 Range: 8000
SR Algorithm:
  Algorithm: 0
  Algorithm: 1
<...>
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
Prefix-SID Index: 101, Algorithm:1, 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.

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.
SUMMARY STEPS

1. configure
2. router isis instance-id
3. interface type interface-path-id
4. point-to-point
5. address-family { ipv4 | ipv6 } [ unicast ]
6. adjacency-sid {index adj-SID-index | absolute adj-SID-value } [protected ]
7. Use the commit or end command.

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> router isis instance-id</td>
<td>Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# router isis 1</td>
<td>• You can change the level of routing to be performed by a particular routing instance by using the is-type router configuration command.</td>
</tr>
<tr>
<td><strong>Step 3</strong> interface type interface-path-id</td>
<td>Specifies the interface and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-isis)# interface GigabitEthernet0/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> point-to-point</td>
<td>Specifies the interface is a point-to-point interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-isis-if)# point-to-point</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> address-family { ipv4</td>
<td>ipv6 } [ unicast ]</td>
</tr>
<tr>
<td><strong>Example:</strong> The following is an example for ipv4 address family: RP/0/RSP0/CPU0:router(config-isis-if)# address-family ipv4 unicast</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> adjacency-sid {index adj-SID-index</td>
<td>absolute adj-SID-value } [protected ]</td>
</tr>
<tr>
<td><strong>Example:</strong> Specify index adj-SID-index for each link to create an Adj-SID based on the lower boundary of the SRLB + the index.</td>
<td>RP/0/RSP0/CPU0:router(config-isis-if-af)#</td>
</tr>
</tbody>
</table>
**Configure Segment Routing for IS-IS Protocol**

### Configuring an Adjacency SID

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>adjacency-sid index 10</td>
<td>Specify absolute adj-SID-value for each link to create a specific Adj-SID within the SRLB. 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.</td>
</tr>
<tr>
<td>adjacency-sid absolute 15010</td>
<td></td>
</tr>
</tbody>
</table>

**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:
  - **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:

```plaintext
RP/0/RSP0/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):

```plaintext
RP/0/RSP0/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 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.
Manually Configure a Layer 2 Adjacency SID

Typically, an adjacency SID (Adj-SID) is associated with a Layer 3 adjacency to a neighboring node, to steer the traffic to a specific adjacency. If you have Layer 2 bundle interfaces, where multiple physical interfaces form a bundle interface, the individual Layer 2 bundle members are not visible to IGP; only the bundle interface is visible.

You can configure a Layer 2 Adj-SID for the individual Layer 2 bundle interfaces. This configuration allows you to track the availability of individual bundle member links and to verify the segment routing forwarding over the individual bundle member links, for Operational Administration and Maintenance (OAM) purposes.

A Layer 2 Adj-SID can be allocated dynamically or configured manually.

- IGP dynamically allocates Layer 2 Adj-SIDs from the dynamic label range for each Layer 2 bundle member. A dynamic Layer 2 Adj-SID is not persistent and can be reallocated as the Layer 2 bundle link goes up and down.

- Manually configured Layer 2 Adj-SIDs are persistent if the Layer 2 bundle link goes up and down. Layer 2 Adj-SIDs are allocated from the Segment Routing Local Block (SRLB) range of labels. However, if the configured value of Layer 2 Adj-SID does not fall within the available SRLB, a Layer 2 Adj-SID will not be programmed into forwarding information base (FIB).

Restrictions

- Adj-SID forwarding requires a next-hop, which can be either an IPv4 address or an IPv6 address, but not both. Therefore, manually configured Layer 2 Adj-SIDs are configured per address-family.

- Manually configured Layer 2 Adj-SID can be associated with only one Layer 2 bundle member link.

- A SID value used for Layer 2 Adj-SID cannot be shared with Layer 3 Adj-SID.

- SR-TE using Layer 2 Adj-SID is not supported.

This task explains how to configure a Layer 2 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.

SUMMARY STEPS

1. `configure`
2. `segment-routing`
3. `adjacency-sid`
4. `interface type interface-path-id`
5. `address-family { ipv4 | ipv6 } [ unicast ]`
6. `l2-adjacency sid {index adj-SID-index | absolute adj-SID-value} [next-hop {ipv4_address | ipv6_address}]

7. Use the `commit` or `end` command.
8. `end`
9. `router isis instance-id`
10. `address-family { ipv4 | ipv6 } [ unicast ]`
11. **segment-routing bundle-member-adj-sid**

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>segment-routing</td>
<td>Enters segment routing configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config)# segment-routing</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>adjacency-sid</td>
<td>Enters adjacency SID configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sr)# adjacency-sid</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>interface type interface-path-id</td>
<td>Specifies the interface and enters interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sr-adj)# interface GigabitEthernet0/0/0/3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>address-family {ipv4</td>
<td>ipv6} [unicast]</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sr-adj-intf)# address-family ipv4 unicast</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>l2-adjacency sid {index adj-SID-index</td>
<td>absolute adj-SID-value} [next-hop {ipv4_address</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sr-adj-intf-af)# l2-adjacency sid absolute 15015 next-hop 10.1.1.4</td>
<td>Specify <strong>index adj-SID-index</strong> for each link to create an Adj-SID based on the lower boundary of the SRLB + the index. Specify <strong>absolute adj-SID-value</strong> for each link to create a specific Adj-SID within the SRLB. For point-to-point interfaces, you are not required to specify a next-hop. However, if you do specify the next-hop, the Layer 2 Adj-SID will be used only if the specified next-hop matches the neighbor address. For LAN interfaces, you must configure the next-hop IPv4 or IPv6 address. If you do not configure the next-hop, the Layer 2 Adj-SID will not be used for LAN interface.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Use the <strong>commit</strong> or <strong>end</strong> command.</td>
<td><strong>commit</strong> — Saves the configuration changes and remains within the configuration session.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
</tbody>
</table>
| **end** | Prompts user to take one of these actions:  
  - **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. |

**Step 8**  
End

**Step 9**  
router isis instance-id  
Example:  
Router(config)# router isis isp

**Step 10**  
address-family { ipv4 | ipv6 } [ unicast ]  
Example:  
Router(config-isis)# address-family ipv4 unicast

**Step 11**  
segment-routing bundle-member-adj-sid  
Example:  
Router(config-isis-af)# segment-routing bundle-member-adj-sid

Verify the configuration:

Router# show mpls forwarding detail | i "Pop|Outgoing Interface|Physical Interface"  
Tue Jun 20 06:53:51.876 PDT

15001  
Pop  
Pop SRLB (idx 1)  
BE1  
10.1.1.4  
0  
Outgoing Interface: Bundle-Ether1 (ifhandle 0x000000b0)  
Physical Interface: GigabitEthernet0/0/0/3 (ifhandle 0x000000b0)

Router# show running-config segment-routing  
Tue Jun 20 07:14:25.815 PDT  
segment-routing  
adjacency-sid  
interface GigabitEthernet0/0/0/3  
address-family ipv4 unicast  
12-adjacency-sid absolute 15001  
!  
!  
!
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**

**SUMMARY STEPS**

1. configure
2. router isis instance-id
3. address-family { ipv4 | ipv6 } [ unicast ]
4. apply-weight ecmp-only bandwidth
5. Use the commit or end command.

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>router isis instance-id</td>
<td>Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# router isis 1</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>address-family { ipv4</td>
<td>ipv6 } [ unicast ]</td>
</tr>
<tr>
<td>Example:</td>
<td>The following is an example for ipv4 address family:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-isis)# address-family ipv4 unicast</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>apply-weight ecmp-only bandwidth</td>
<td>Enables UCMP functionality locally between ECMP paths based on the bandwidth of the local links.</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis-af)# apply-weight ecmp-only bandwidth</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use the <strong>commit</strong> or <strong>end</strong> command.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>commit — Saves the configuration changes and remains within the configuration session.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>end — Prompts user to take one of these actions:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• <strong>Yes</strong> — Saves configuration changes and exits the configuration session.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• <strong>No</strong> — Exits the configuration session without committing the configuration changes.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• <strong>Cancel</strong> — Remains in the configuration session, without committing the configuration changes.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**IS-IS Prefix Attributes for Extended IPv4 and IPv6 Reachability**

The following sub-TLVs support the advertisement of IPv4 and IPv6 prefix attribute flags and the source router ID of the router that originated a prefix advertisement, as described in RFC 7794.

- Prefix Attribute Flags
- IPv4 and IPv6 Source Router ID

**Prefix Attribute Flags**

The Prefix Attribute Flag sub-TLV supports the advertisement of attribute flags associated with prefix advertisements. Knowing if an advertised prefix is directly connected to the advertising router helps to determine how labels that are associated with an incoming packet should be processed.

This section describes the behavior of each flag when a prefix advertisement is learned from one level to another.

**Note**

Prefix attributes are only added when wide metric is used.

**Prefix Attribute Flags Sub-TLV Format**

```
0 1 2 3 4 5 6 7 ...
+-------------------...
|X|R|N|...
+-------------------...
```
Prefix Attribute Flags Sub-TLV Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X (External Prefix Flag)</td>
<td>This flag is set if the prefix has been redistributed from another protocol. The value of the flag is preserved when the prefix is propagated to another level.</td>
</tr>
<tr>
<td>R (Re-advertisement Flag)</td>
<td>This flag is set to 1 by the Level 1-2 router when the prefix is propagated between IS-IS levels (from Level 1 to Level 2, or from Level 2 to Level 1). This flag is set to 0 when the prefix is connected locally to an IS-IS-enabled interface (regardless of the level configured on the interface).</td>
</tr>
<tr>
<td>N (Node Flag)</td>
<td>For prefixes that are propagated from another level:</td>
</tr>
<tr>
<td></td>
<td>1. Copy the N-flag from the prefix attribute sub-TLV, if present in the source level.</td>
</tr>
<tr>
<td></td>
<td>2. Copy the N-flag from the prefix-SID sub-TLV, if present in the source level.</td>
</tr>
<tr>
<td></td>
<td>3. Otherwise, set to 0.</td>
</tr>
<tr>
<td></td>
<td>For connected prefixes:</td>
</tr>
<tr>
<td></td>
<td>1. Set to 0 if prefix-attributes n-flag-clear is configured (see Configuring Prefix Attribute N-flag-clear).</td>
</tr>
<tr>
<td></td>
<td>2. Set to 0 if n-flag-clear SID-index</td>
</tr>
<tr>
<td></td>
<td>3. Otherwise, set to 1 when the prefix is a host prefix (/32 for IPv4, /128 for IPv6) that is associated with a loopback address.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> If the flag is set and the prefix length is not a host prefix, then the flag must be ignored.</td>
</tr>
</tbody>
</table>

**IPv4 and IPv6 Source Router ID**

The Source Router ID sub-TLV identifies the source of the prefix advertisement. The IPv4 and IPv6 source router ID is displayed in the output of the `show isis database verbose` command.

The Source Router ID sub-TLV is added when the following conditions are met:

1. The prefix is locally connected.
2. The N-flag is set to 1 (when it's a host prefix and the `n-flag-clear` configuration is not used).
3. The router ID is configured in the corresponding address family.

The source router ID is propagated between levels.
Table 3: Source Router Sub-TLV Format

<table>
<thead>
<tr>
<th></th>
<th>Type: 11</th>
<th>Length: 4</th>
<th>Value: IPv4 Router ID of the source of the prefix advertisement</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 Source Router ID</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Type: 12</th>
<th>Length: 16</th>
<th>Value: IPv6 Router ID of the source of the prefix advertisement</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv6 Source Router ID</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Configuring Prefix Attribute N-flag-clear**

The N-flag is set to 1 when the prefix is a host prefix (/32 for IPv4, /128 for IPv6) that is associated with a loopback address. The advertising router can be configured to not set this flag. This task explains how to clear the N-flag.

**SUMMARY STEPS**

1. `configure`
2. `router isis instance-id`
3. `interface Loopback instance`
4. `prefix-attributes n-flag-clear [Level-1 | Level-2]`
5. Use the `commit` or `end` command.

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/O/RSP0/CPU0:router# configure</td>
</tr>
</tbody>
</table>

| Step 2             |         |
| `router isis instance-id` |         |
| Example:            | RP/O/RSP0/CPU0:router(config)# router isis 1 |

| Step 3             |         |
| `interface Loopback instance` | Specifies the loopback interface. |
| Example:            | RP/O/RSP0/CPU0:router(config)# interface Loopback0 |

| Step 4             |         |
| `prefix-attributes n-flag-clear [Level-1 | Level-2]` | Clears the prefix attribute N-flag explicitly. |
| Example:            |         |
Configure Segment Routing for IS-IS Protocol

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| `RP/0/RSP0/CPU0:router(config-if)# isis prefix-attributes n-flag-clear` | **commit** — Saves the configuration changes and remains within the configuration session. **end** — Prompts user to take one of these actions:  
  • 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. |

### Step 5
Use the **commit** or **end** command.

Verify the prefix attribute configuration:

```
RP/0/RSP0/CPU0:router# show isis database verbose
IS-IS 1 (Level-2) Link State Database
LSPID   LSP Seq Num LSP Checksum LSP Holdtime ATT/P/OL
router.00-00  * 0x0000039b 0xfc27 1079 0/0/0
Area Address: 49.0001
NLPIID: 0xcc
NLPIID: 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: 0
Algorithm: 1
Metric: 0
  IP-Extended 10.0.0.1/32
  Prefix-SID Index: 1001, Algorithm:0, R:1 N:0 P:1 E:0 V:0 L:0
Prefix Attribute Flags: X:0 R:1 N:0
Metric: 10
  IP-Extended 10.0.0.2/32
  Prefix-SID Index: 1002, Algorithm:0, R:0 N:1 P:0 E:0 V:0 L:0
Prefix Attribute Flags: X:0 R:0 N:1
Source Router ID: 10.0.0.2
```

### 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) (see Configure Segment Routing Path Computation Element).

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 6

Configure Segment Routing for OSPF Protocol

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

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

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

Note

- Enabling Segment Routing for OSPF Protocol, on page 61
- Configuring a Prefix-SID on the OSPF-Enabled Loopback Interface, on page 63

Enabling Segment Routing for OSPF Protocol

Segment routing on the OSPF control plane supports the following:

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

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

Before you begin

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

**SUMMARY STEPS**

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

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</td>
</tr>
<tr>
<td><code>router ospf process-name</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# router ospf 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enables segment routing using the MPLS data plane on the routing process and all areas and interfaces in the routing process. Enables segment routing forwarding on all interfaces in the routing process and installs the SIDs received by OSPF in the forwarding table.</td>
</tr>
<tr>
<td><code>segment-routing mpls</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# segment-routing mpls</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Enters area configuration mode.</td>
</tr>
<tr>
<td><code>area 0</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# area 0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Enables IGP traffic engineering functionality.</td>
</tr>
<tr>
<td><code>mpls traffic-eng area</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar)# mpls traffic-eng area 0</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 6</td>
<td><code>mpls traffic-eng router-id interface</code></td>
<td>Sets the traffic engineering loopback interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>mpls traffic-eng router-id Loopback0</code></td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td><code>segment-routing mpls</code></td>
<td>(Optional) Enables segment routing using the MPLS data plane on the area and all interfaces in the area. Enables segment routing forwarding on all interfaces in the area and installs the SIDs received by OSPF in the forwarding table.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>segment-routing mpls</code></td>
<td></td>
</tr>
<tr>
<td>Step 8</td>
<td><code>exit</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>exit</code></td>
<td></td>
</tr>
<tr>
<td>Step 9</td>
<td><code>mpls traffic-eng</code></td>
<td>Enables traffic engineering functionality on the node. The node advertises the traffic engineering link attributes in IGP which populates the traffic engineering database (TED) on the head-end. The SR-TE head-end requires the TED to calculate and validate the path of the SR-TE policy.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>mpls traffic-eng</code></td>
<td></td>
</tr>
<tr>
<td>Step 10</td>
<td>Use the <code>commit</code> or <code>end</code> command.</td>
<td><code>commit</code> — Saves the configuration changes and remains within the configuration session. <code>end</code> — Prompts user to take one of these actions: • 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.</td>
</tr>
</tbody>
</table>

**What to do next**

Configure the prefix SID.

## 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.

**SUMMARY STEPS**

1. `configure`
2. `router ospf  process-name`
3. `area  value`
4. `interface Loopback  interface-instance`
5. `prefix-sid [strict-spf | algorithm  algorithm-number] [index  SID-index | absolute  SID-value ] [n-flag-clear] [explicit-null]`
6. Use the `commit` or `end` command.

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enables OSPF routing for the specified routing process, and places the router in router configuration mode.</td>
</tr>
<tr>
<td><code>router ospf  process-name</code></td>
<td></td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CPU0:router(config)# router ospf 1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enters area configuration mode.</td>
</tr>
<tr>
<td><code>area  value</code></td>
<td></td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CPU0:router(config-ospf)# area 0</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Specifies the loopback interface and instance.</td>
</tr>
<tr>
<td><code>interface Loopback  interface-instance</code></td>
<td></td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CPU0:router(config-ospf-ar)# interface Loopback0 passive</code></td>
<td></td>
</tr>
</tbody>
</table>
Configure a Prefix-SID on the OSPF-Enabled Loopback Interface

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>prefix-sid [strict-spf</td>
<td>algorithm algorithm-number] [index SID-index</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Verify the prefix-SID configuration:

```
RP/0/RSP0/CPU0:router# show ospf database opaque-area 7.0.0.1 self-originate
OSPF Router with ID {10.0.0.1} (Process ID 1)
   Type-10 Opaque Link Area Link States (Area 0)

<...>
   Extended Prefix TLV: Length: 20
      Route-type: 1
      AF : 0
      Flags : 0x40
      Prefix : 10.0.0.1/32

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

What to do next

Configure SR-TE Policies
Configure Segment Routing for BGP

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

This module provides the configuration information used to enable Segment Routing for BGP.

---

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

- Segment Routing for BGP, on page 67
- Configure BGP Prefix Segment Identifiers, on page 68
- Segment Routing Egress Peer Engineering, on page 69
- Configure BGP Link-State, on page 70
- Example: Configuring SR-EPE and BGP-LS, on page 71
- Configure BGP Proxy Prefix SID, on page 73

---

Segment Routing for BGP

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

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

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

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

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

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

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

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

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

```plaintext
RP/0/RSP0/CPU0:router(config)# segment-routing global-block 16000 23999
RP/0/RSP0/CPU0:router(config)# route-policy SID($SID)
RP/0/RSP0/CPU0:router(config-rpl)# set label-index $SID
RP/0/RSP0/CPU0:router(config-rpl)# end policy
RP/0/RSP0/CPU0:router(config)# route bgp 1
RP/0/RSP0/CPU0:router(config-bgp)# bgp router-id 1.1.1.1
RP/0/RSP0/CPU0:router(config-bgp)# address-family ipv4 unicast
RP/0/RSP0/CPU0:router(config-bgp-af)# network 1.1.1.3/32 route-policy SID(3)
RP/0/RSP0/CPU0:router(config-bgp-af)# allocate-label all
RP/0/RSP0/CPU0:router(config-bgp-af)# commit
RP/0/RSP0/CPU0:router(config-bgp-af)# end

RP/0/RSP0/CPU0:router# show bgp 1.1.1.3/32
BGP routing table entry for 1.1.1.3/32
Versions:
  Process bRIB/RIB SendTblVer
  Speaker 74 74
  Local Label=16003
Last Modified: Sep 29 19:52:18.155 for 00:07:22
Paths: (1 available, best #1)
  Advertised to update-groups (with more than one peer):
    0.2
```
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.

SUMMARY STEPS

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

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 router bgp as-number</td>
<td>Specifies the BGP AS number and enters the BGP configuration mode, allowing you to configure the BGP routing process.</td>
</tr>
</tbody>
</table>

Example:

RP/0/RSP0/CPU0:router(config)# router bgp 1

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
### Purpose

Command or Action | Purpose
--- | ---
**Step 2** neighbor ip-address | Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.

**Example:**

RP/0/RSP0/CPU0:router(config-bgp)# neighbor 192.168.1.3

**Step 3** remote-as as-number | Creates a neighbor and assigns a remote autonomous system number to it.

**Example:**

RP/0/RSP0/CPU0:router(config-bgp-nbr)# remote-as 3

**Step 4** egress-engineering | Configures the egress node with EPE for the eBGP peer.

**Example:**

RP/0/RSP0/CPU0:router(config-bgp-nbr)# egress-engineering

### Configure BGP Link-State

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

**Note**

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

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

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

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

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

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

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

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# router isis isp
RP/0/RSP0/CPU0:router(config-isis)# distribute link-state instance-id 32 level 2 throttle 5
```

To distribute OSPFv2 and OSPFv3 link-state data using BGP LS, use the `distribute link-state` command in router configuration mode.

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# router ospf 100
RP/0/RSP0/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](image)

Step 1

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

Example:

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

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

Example:

RP/0/RSP0/CPU0:router_C(config-bgp)## neighbor 172.29.50.71
RP/0/RSP0/CPU0:router_C(config-bgp-nbr)## remote-as 1
RP/0/RSP0/CPU0:router_C(config-bgp-nbr)## description to EPE_controller
RP/0/RSP0/CPU0:router_C(config-bgp-nbr)## address-family link-state link-state
RP/0/RSP0/CPU0:router_C(config-bgp-nbr-sf)## route-policy bgp_in in
RP/0/RSP0/CPU0:router_C(config-bgp-nbr-sf)## route-policy bgp_out out
RP/0/RSP0/CPU0:router_C(config-bgp-nbr-sf)## exit
RP/0/RSP0/CPU0:router_C(config-bgp)## exit

Step 3 Commit the configuration.

Example:

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

Step 4 Verify the configuration.

Example:

RP/0/RSP0/CPU0:router_C## 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.
The output shows that node C installed peer node SIDs in the Forwarding Information Base (FIB).

Configure BGP Proxy Prefix SID

To support segment routing, Border Gateway Protocol (BGP) requires the ability to advertise a segment identifier (SID) for a BGP prefix. A BGP-Prefix-SID is the segment identifier of the BGP prefix segment in a segment routing network. BGP prefix SID attribute is a BGP extension to signal BGP prefix-SIDs. However, there may be routers which do not support BGP extension for segment routing. Hence, those routers also do not support BGP prefix SID attribute and an alternate approach is required.

BGP proxy prefix SID feature allows you to attach BGP prefix SID attributes for remote prefixes learnt from BGP labeled unicast (LU) neighbours which are not SR-capable and propagate them as SR prefixes. This allows an LSP towards non SR endpoints to use segment routing global block in a SR domain. Since BGP proxy prefix SID uses global label values it minimizes the use of limited resources such as ECMP-FEC and provides more scalability for the networks.

BGP proxy prefix SID feature is implemented using the segment routing mapping server (SRMS). SRMS allows the user to configure SID mapping entries to specify the prefix-SIDs for the prefixes. The mapping server advertises the local SID-mapping policy to the mapping clients. BGP acts as a client of the SRMS and uses the mapping policy to calculate the prefix-SIDs.

Configuration Example:

This example shows how to configure the BGP proxy prefix SID feature for the segment routing mapping server.

```
RP/0/RSP0/CPU0:router(config)# segment-routing
RP/0/RSP0/CPU0:router(config-ar)# mapping-server
RP/0/RSP0/CPU0:router(config-ar-ms)# prefix-sid-map
RP/0/RSP0/CPU0:router(config-ar-ms-map)# address-family ipv4
RP/0/RSP0/CPU0:router(config-ar-ms-map-af)# 1.1.1.1/32 10 range 200
RP/0/RSP0/CPU0:router(config-ar-ms-map-af)# 192.168.64.1/32 400 range 300
```

This example shows how to configure the BGP proxy prefix SID feature for the segment-routing mapping client.

```
RP/0/RSP0/CPU0:router(config)# router bgp 1
RP/0/RSP0/CPU0:router(config-bgp)# address-family ip4 unicast
RP/0/RSP0/CPU0:router(config-bgp-af)# segment-routing prefix-sid-map
```

Verification

These examples show how to verify the BGP proxy prefix SID feature.
Configure BGP Proxy Prefix SID

show segment-routing mapping-server prefix-sid-map ipv4
detail
Prefix
1.1.1.1/32
   SID Index: 10
   Range: 200
   Last Prefix: 1.1.1.200/32
   Last SID Index: 209
   Flags: 
   Number of mapping entries: 1

show bgp ipv4 labeled-unicast 192.168.64.1/32

BGP routing table entry for 192.168.64.1/32
Versions:
   Process bRIB/RIB SendTblVer
   Speaker 117 117
   Local Label: 16400
Last Modified: Oct 25 01:02:28.562 for 00:11:45
Paths: (2 available, best #1)
   Advertised to peers (in unique update groups):
      201.1.1.1
Path #1: Received by speaker 0  Advertised to peers (in unique update groups):
      201.1.1.1
Local
   20.0.101.1 from 20.0.101.1 (20.0.101.1) Received Label 61
   Origin IGP, localpref 100, valid, internal, best, group-best, multipath, labeled-unicast
   Received Path ID 0, Local Path ID 0, version 117
   Prefix SID Attribute Size: 7
   Label Index: 1

show route ipv4 unicast 192.68.64.1/32 detail

Routing entry for 192.168.64.1/32
   Known via "bgp 65000", distance 200, metric 0, [ei]-bgp, labeled SR, type internal
   Installed Oct 25 01:02:28.583 for 00:20:09
Routing Descriptor Blocks
   20.0.101.1, from 20.0.101.1, BGP multi path
      Route metric is 0
      Label: 0x3d (61)
      Tunnel ID: None
      Binding Label: None
      Extended communities count: 0
      NHID:0x0(Ref:0)
      Route version is 0x6 (6)
      Local Label: 0x3e81 (16400)
      IP Precedence: Not Set
      QoS Group ID: Not Set
      Flow-tag: Not Set
      Fwd-class: Not Set
      Route Priority: RIB_PRIORITY_RECURSIVE (12) SVD Type RIB_SVD_TYPE_LOCAL
      Download Priority 4, Download Version 242
      No advertising protos.

show cef ipv4 192.168.64.1/32 detail
192.168.64.1/32, version 476, labeled SR, drop adjacency, internal 0x5000001 0x80 (ptr 0x71c42b40) [1], 0x0 (0x71c11590), 0x808 (0x722b91e0)
   Updated Oct 31 23:23:48.733
   Prefix Len 32, traffic index 0, precedence n/a, priority 4
   Extensions: context-label:16400
gateway array (0x71ae7e78) reference count 3, flags 0x7a, source rib (7), 0 backups
   [2 type 5 flags 0x88401 (0x722eb450) ext 0x0 (0x0)]
LW-LDI[type=5, refc=3, ptr=0x71c11590, sh-ldi=0x722eb450]
LDI Update time Oct 31 23:23:48.733
via 20.0.101.1/32, 0 dependencies, recursive, bgp-ext [flags 0x6020]
  path-idx 0 NHID 0x0 [0x7129a294 0x0]
  recursion-via-/32
unresolved
  local label 16400
  labels imposed {ExpNullv6}

RP/0/RSP0/CPU0:router# show bgp labels
BGP router identifier 2.1.1.1, local AS number 65000
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000 RD version: 245
BGP main routing table version 245
BGP NSR Initial initsync version 16 (Reached)
BGP NSR/ISSU Sync-Group versions 245/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
  i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Rcvd Label</th>
<th>Local Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>*1.1.1.1/32</td>
<td>1.1.1.1</td>
<td>3</td>
<td>16010</td>
</tr>
<tr>
<td>*2.1.1.1/32</td>
<td>0.0.0.0</td>
<td>nolabel</td>
<td>3</td>
</tr>
<tr>
<td>*192.68.64.1/32</td>
<td>20.0.101.1</td>
<td>2</td>
<td>16400</td>
</tr>
<tr>
<td>*192.68.64.2/32</td>
<td>20.0.101.1</td>
<td>2</td>
<td>16401</td>
</tr>
</tbody>
</table>
Configure BGP Proxy Prefix SID
CHAPTER 8

Configure SR-TE Policies

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

- SR-TE Policy Overview, on page 77
- Instantiation of an SR Policy, on page 78
- SR-TE Policy Path Types, on page 112
- Protocols, on page 127
- Traffic Steering, on page 133
- Miscellaneous, on page 144

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.
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:

- On-Demand SR Policy – SR On-Demand Next-Hop, on page 78
- Manually Provisioned SR Policy, on page 112
- PCE-Initiated SR Policy, on page 112

On-Demand SR Policy – SR On-Demand Next-Hop

Segment Routing On-Demand Next Hop (SR-ODN) allows a service head-end router to automatically instantiate an SR policy to a BGP next-hop when required (on-demand). Its key benefits include:

- **SLA-aware BGP service** – Provides per-destination steering behaviors where a prefix, a set of prefixes, or all prefixes from a service can be associated with a desired underlay SLA. The functionality applies equally to single-domain and multi-domain networks.
- **Simplicity** – No prior SR Policy configuration needs to be configured and maintained. Instead, operator simply configures a small set of common intent-based optimization templates throughout the network.
- **Scalability** – Device resources at the head-end router are used only when required, based on service or SLA connectivity needs.

The following example shows how SR-ODN works:

![SR-ODN Diagram]

---

*Configure SR-TE Policies*

*Instantiation of an SR Policy*
1. An egress PE (node H) advertises a BGP route for prefix T/t. This advertisement includes an SLA intent encoded with a BGP color extended community. In this example, the operator assigns color purple (example value = 100) to prefixes that should traverse the network over the delay-optimized path.

2. The route reflector receives the advertised route and advertises it to other PE nodes.

3. Ingress PEs in the network (such as node F) are pre-configured with an ODN template for color purple that provides the node with the steps to follow in case a route with the intended color appears, for example:
   - Contact SR-PCE and request computation for a path toward node H that does not share any nodes with another LSP in the same disjointness group.
   - At the head-end router, compute a path towards node H that minimizes cumulative delay.

4. In this example, the head-end router contacts the SR-PCE and requests computation for a path toward node H that minimizes cumulative delay.

5. After SR-PCE provides the compute path, an intent-driven SR policy is instantiated at the head-end router. Other prefixes with the same intent (color) and destined to the same egress PE can share the same on-demand SR policy. When the last prefix associated with a given [intent, egress PE] pair is withdrawn, the on-demand SR policy is deleted, and resources are freed from the head-end router.

An on-demand SR policy is created dynamically for BGP global or VPN (service) routes. The following services are supported with SR-ODN:

- IPv4 BGP global routes
- IPv6 BGP global routes (6PE)
- VPNv4
-VPNv6 (6vPE)
- EVPN-VPWS (single-homing)
- EVPN-VPWS (multi-homing)
- EVPN (single-homing/multi-homing)

Note: Colored per-ESI/per-EVI EVPN Ethernet Auto-Discovery route (route-type 1) and Inclusive Multicast Route (route-type 3) are used to trigger instantiation of ODN SR-TE policies.

Note: The following scenarios involving virtual Ethernet Segments (vES) are also supported with EVPN ODN:

- VPLS VFI as vES for single-active Multi-Homing to EVPN
- Active/backup Pseudo-wire (PW) as vES for Single-Homing to EVPN
- Static Pseudo-wire (PW) as vES for active-active Multi-Homing to EVPN
Configuration Steps

To configure SR-ODN, complete the following configurations:

1. Define the SR-ODN template on the SR-TE head-end router.
   (Optional) If using Segment Routing Path Computation Element (SR-PCE) for path computation:
   a. Configure SR-PCE. For detailed SR-PCE configuration information, see Configure SR-PCE, on page 172.
   b. Configure the head-end router as Path Computation Element Protocol (PCEP) Path Computation Client (PCC). For detailed PCEP PCC configuration information, see Configure the Head-End Router as PCEP PCC, on page 127.


   The following RPL attach-points for setting/matching BGP color extended communities are supported:

   - VRF export
   - VRF import
   - EVI export
   - EVI import
   - Neighbor-in
   - Neighbor-out
   - Inter-AFI export
   - Inter-AFI import
   - Default-originate

   The following table shows the supported RPL match operations; however, routing policies are required primarily to set BGP color extended community. Matching based on BGP color extended communities is performed automatically by ODN's on-demand color template.

<table>
<thead>
<tr>
<th>Attach Point</th>
<th>Set</th>
<th>Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>VRF export</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>VRF import</td>
<td>–</td>
<td>X</td>
</tr>
<tr>
<td>EVI export</td>
<td>X</td>
<td>–</td>
</tr>
<tr>
<td>EVI import</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Neighbor-in</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Neighbor-out</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Inter-AFI export</td>
<td>–</td>
<td>X</td>
</tr>
<tr>
<td>Inter-AFI import</td>
<td>–</td>
<td>X</td>
</tr>
<tr>
<td>Default-originate</td>
<td>X</td>
<td>–</td>
</tr>
</tbody>
</table>

Configure On-Demand Color Template

• Use the `on-demand color color` command to create an ODN template for the specified color value. The head-end router automatically follows the actions defined in the template upon arrival of BGP global or VPN routes with a BGP color extended community that matches the color value specified in the template.

The color range is from 1 to 4294967295.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# on-demand color 10
```

**Note**
Matching based on BGP color extended communities is performed automatically via ODN's on-demand color template. RPL routing policies are not required.

• Use the `on-demand color color dynamic` command to associate the template with on-demand SR policies with a locally computed dynamic path (by SR-TE head-end router utilizing its TE topology database) or centrally (by SR-PCE). The head-end router will first attempt to install the locally computed path; otherwise, it will use the path computed by the SR-PCE.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# on-demand color 10 dynamic
```

• Use the `on-demand color color dynamic pcep` command to indicate that only the path computed by SR-PCE should be associated with the on-demand SR policy. With this configuration, local path computation is not attempted; instead the head-end router will only instantiate the path computed by the SR-PCE.

```
Router(config-sr-te)# on-demand color 10 dynamic pcep
```

Configure Dynamic Path Optimization Objectives

• Use the `metric type {igp | te | latency}` command to configure the metric for use in path computation.

```
Router(config-sr-te-color-dyn)# metric type te
```

• Use the `metric margin {absolute value | relative percent}` command to configure the On-Demand dynamic path metric margin. The range for `value` and `percent` is from 0 to 2147483647.

```
Router(config-sr-te-color-dyn)# metric margin absolute 5
```

Configure Dynamic Path Constraints

• Use the `disjoint-path group-id group-id type {link | node | srlg | srlg-node} [sub-id sub-id]` command to configure the disjoint-path constraints. The `group-id` and `sub-id` range is from 1 to 65535.

```
Router(config-sr-te-color-dyn)# disjoint-path group-id 775 type link
```

• Use the `affinity {include-any | include-all | exclude-any} {name WORD}` command to configure the affinity constraints.

```
Router(config-sr-te-color-dyn)# affinity exclude-any name CROSS
```
• Use the `sid-algorithm algorithm-number` command to configure the SR Flexible Algorithm constraints. The `algorithm-number` range is from 128 to 255.

```
Router(config-sr-te-color-dyn) # sid-algorithm 128
```

• Use the `maximum-sid-depth value` command to customize the maximum SID depth (MSD) constraints advertised by the router.

The default MSD `value` is equal to the maximum MSD supported by the platform (10).

```
Router(config-sr-te-color) # maximum-sid-depth 5
```

For cases with path computation at PCE, a PCC can signal its MSD to the PCE in the following ways:

• During PCEP session establishment – The signaled MSD is treated as a node-wide property.
  
  • MSD is configured under `segment-routing traffic-eng maximum-sid-depth value` command

• During PCEP LSP path request – The signaled MSD is treated as an LSP property.
  
  • On-demand (ODN) SR Policy: MSD is configured using the `segment-routing traffic-eng on-demand color color maximum-sid-depth value` command

  • Local SR Policy: MSD is configured using the `segment-routing traffic-eng policy WORD candidate-paths preference preference dynamic metric sid-limit value` command.

---

**Note**

If the configured MSD values are different, the per-LSP MSD takes precedence over the per-node MSD.

---

After path computation, the resulting label stack size is verified against the MSD requirement.

• If the label stack size is larger than the MSD and path computation is performed by PCE, then the PCE returns a "no path" response to the PCC.

• If the label stack size is larger than the MSD and path computation is performed by PCC, then the PCC will not install the path.

---

**Note**

A sub-optimal path (if one exists) that satisfies the MSD constraint could be computed in the following cases:

• For a dynamic path with TE metric, when the PCE is configured with the `pce segment-routing te-latency` command or the PCC is configured with the `segment-routing traffic-eng te-latency` command.

  • For a dynamic path with LATENCY metric

  • For a dynamic path with affinity constraints

For example, if the PCC MSD is 4 and the optimal path (with an accumulated metric of 100) requires 5 labels, but a sub-optimal path exists (with accumulated metric of 110) requiring 4 labels, then the sub-optimal path is installed.
Configuring SR-ODN: Examples

Configuring SR-ODN: Layer-3 Services Examples

The following examples show end-to-end configurations used in implementing SR-ODN on the head-end router.

**Configuring ODN Color Templates: Example**

Configure ODN color templates on routers acting as SR-TE head-end nodes. The following example shows various ODN color templates:

- color 10: minimization objective = te-metric
- color 20: minimization objective = igp-metric
- color 21: minimization objective = igp-metric; constraints = affinity
- color 22: minimization objective = te-metric; path computation at SR-PCE; constraints = affinity
- color 30: minimization objective = delay-metric
- color 128: constraints = flex-algo

```plaintext
segment-routing
traffic-eng
on-demand color 10
dynamic
  metric
type te
!
!
!

on-demand color 20
dynamic
  metric
type igp
!
!
!

on-demand color 21
dynamic
  metric
type igp
  affinity exclude-any
    name CROSS
!
!
!

on-demand color 22
dynamic
  pcep
!
  metric
type te
  affinity exclude-any
    name CROSS
!
!
!
```
on-demand color 30
dynamic
metric
type latency
!
!
on-demand color 128
dynamic
sid-algorithm 128
!
!
end

**Configuring BGP Color Extended Community Set: Example**

The following example shows how to configure BGP color extended communities that are later applied to BGP service routes via route-policies.

> Note
>
> In most common scenarios, egress PE routers that advertise BGP service routes apply (set) BGP color extended communities. However, color can also be set at the ingress PE router.

```
extcommunity-set opaque color10-te
  10
end-set
!
estcommunity-set opaque color20-igp
  20
end-set
!
estcommunity-set opaque color21-igp-excl-cross
  21
end-set
!
estcommunity-set opaque color30-delay
  30
end-set
!
estcommunity-set opaque color128-fa128
  128
end-set
!
```

**Configuring RPL to Set BGP Color (Layer-3 Services): Examples**

The following example shows various representative RPL definitions that set BGP color community.

The first 4 RPL examples include the set color action only. The last RPL example performs the set color action for selected destinations based on a prefix-set.

```
route-policy SET_COLOR_LOW_LATENCY_TE
  set extcommunity color color10-te
  pass
end-policy
!
route-policy SET_COLOR_HI_BW
  set extcommunity color color20-igp
  pass
end-policy
```
!  
route-policy SET_COLOR_LOW_LATENCY  
   set extcommunity color color30-delay  
pass  
end-policy  
!  
route-policy SET_COLOR_FA_128  
   set extcommunity color color128-fa128  
pass  
end-policy  
!  
prefix-set sample-set  
   88.1.0.0/24  
end-set  
!  
route-policy SET_COLOR_GLOBAL 
   if destination in sample-set then 
      set extcommunity color color10-te 
   else 
      pass  
   endif 
end-policy  

Applying RPL to BGP Services (Layer-3 Services): Example  
The following example shows various RPLs that set BGP color community being applied to BGP Layer-3 VPN services (VPNv4/VPNv6) and BGP global.  

• The L3VPN examples show the RPL applied at the VRF export attach-point.  
• The BGP global example shows the RPL applied at the BGP neighbor-out attach-point.  

vrf vrf_cust1  
   address-family ipv4 unicast  
      export route-policy SET_COLOR_LOW_LATENCY_TE  
   !  
   address-family ipv6 unicast  
      export route-policy SET_COLOR_LOW_LATENCY_TE  
   !  
!  
vrf vrf_cust2  
   address-family ipv4 unicast  
      export route-policy SET_COLOR_HI_BW  
   !  
   address-family ipv6 unicast  
      export route-policy SET_COLOR_HI_BW  
   !  
!  
vrf vrf_cust3  
   address-family ipv4 unicast  
      export route-policy SET_COLOR_LOW_LATENCY  
   !  
   address-family ipv6 unicast  
      export route-policy SET_COLOR_LOW_LATENCY  
   !  
!  
vrf vrf_cust4  
   address-family ipv4 unicast  
      export route-policy SET_COLOR_FA_128  
   !  
   address-family ipv6 unicast
Verifying BGP VRF Information

Use the `show bgp vrf` command to display BGP prefix information for VRF instances. The following output shows the BGP VRF table including a prefix (88.1.1.0/24) with color 10 advertised by router 1.1.1.8.

```
RP/0/RP0/CP00:R4# show bgp vrf vrf_cust1
BGP VRF vrf_cust1, state: Active
BGP Route Distinguisher: 1.1.1.4:101
VRF ID: 0x60000007
BGP router identifier 1.1.1.4, local AS number 100
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000007 RD version: 282
BGP main routing table version 287
BGP NSR Initial init-sync version 31 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0

Status codes: s suppressed, d damped, h history, * valid, > best
  i - internal, r RIB-failure, S stale, N Next-hop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
Network       Next Hop     Metric LocPrf Weight Path
Route Distinguisher: 1.1.1.4:101 (default for vrf vrf_cust1)
*> 44.1.1.0/24 40.4.101.11 0 400 {1} i
*>155.1.1.0/24 1.1.1.5 100 0 500 {1} i
*>i88.1.1.0/24 1.1.1.8 C:10 100 0 800 {1} i
*>199.1.1.0/24 1.1.1.9 100 0 800 {1} i

Processed 4 prefixes, 4 paths
```

The following output displays the details for prefix 88.1.1.0/24. Note the presence of BGP extended color community 10, and that the prefix is associated with an SR policy with color 10 and BSID value of 24036.

```
RP/0/RP0/CP00:R4# show bgp vrf vrf_cust1 88.1.1.0/24
BGP routing table entry for 88.1.1.0/24, Route Distinguisher: 1.1.1.4:101
Versions:
  Process bRIB/RIB SendTblVer
Speaker   282 282
Last Modified: May 20 09:23:34.112 for 00:06:03
Paths: (1 available, best #1)
  Advertised to CE peers (in unique update groups):
  40.4.101.11
  Path #1: Received by speaker 0
  Advertised to CE peers (in unique update groups):
  40.4.101.11
  800 {1}
  1.1.1.8 C:10 (bsid:24036) (metric 20) from 1.1.1.55 (1.1.1.8)
    Received Label 24012
    Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate, imported
```
Received Path ID 0, Local Path ID 1, version 273
Extended community: Color:10 RT:100:1
Originator: 1.1.1.8, Cluster list: 1.1.1.55
SR policy color 10, up, registered, bsid 24036, if-handle 0x08000024
Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 1.1.1.8:101

Verifying Forwarding (CEF) Table

Use the `show cef vrf` command to display the contents of the CEF table for the VRF instance. Note that prefix 88.1.1.0/24 points to the BSID label corresponding to an SR policy. Other non-colored prefixes, such as 55.1.1.0/24, point to BGP next-hop.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Next Hop</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0/0</td>
<td>drop</td>
<td>default handler</td>
</tr>
<tr>
<td>0.0.0.0/32</td>
<td>broadcast</td>
<td></td>
</tr>
<tr>
<td>40.4.101.0/24</td>
<td>attached</td>
<td>TenGigE0/0/0/0.101</td>
</tr>
<tr>
<td>40.4.101.0/32</td>
<td>broadcast</td>
<td>TenGigE0/0/0/0.101</td>
</tr>
<tr>
<td>40.4.101.11/32</td>
<td>receive</td>
<td>TenGigE0/0/0/0.101</td>
</tr>
<tr>
<td>40.4.101.11/32</td>
<td>40.4.101.11/32</td>
<td>TenGigE0/0/0/0.101</td>
</tr>
<tr>
<td>40.4.101.255/32</td>
<td>broadcast</td>
<td>TenGigE0/0/0/0.101</td>
</tr>
<tr>
<td>44.1.1.0/24</td>
<td>40.4.101.11/32</td>
<td>&lt;recursive&gt;</td>
</tr>
<tr>
<td>55.1.1.0/24</td>
<td>1.1.1.5/32</td>
<td>&lt;recursive&gt;</td>
</tr>
<tr>
<td><strong>88.1.1.0/24</strong></td>
<td><strong>24036 (via-label)</strong></td>
<td><strong>&lt;recursive&gt;</strong></td>
</tr>
<tr>
<td>99.1.1.0/24</td>
<td>1.1.1.9/32</td>
<td>&lt;recursive&gt;</td>
</tr>
<tr>
<td>224.0.0.0/4</td>
<td>0.0.0.0/32</td>
<td></td>
</tr>
<tr>
<td>224.0.0.0/24</td>
<td>receive</td>
<td></td>
</tr>
<tr>
<td>255.255.255.255/32</td>
<td>broadcast</td>
<td></td>
</tr>
</tbody>
</table>

The following output displays CEF details for prefix 88.1.1.0/24. Note that the prefix is associated with an SR policy with BSID value of 24036.

```
RP/0/RP0/CPU0:R4# show cef vrf vrf_cust1 88.1.1.0/24
88.1.1.0/24, version 51, internal 0x50000001 0x0 (ptr 0x98c60ddc) [1], 0x0 (0x0), 0x208 (0x98425268)
Updated May 20 09:23:34.216
Prefix Len 24, traffic index 0, precedence n/a, priority 3
via local-label 24036, 5 dependencies, recursive [flags 0x6000]
path-idx 0 NHID 0x0 [0x97091ec0 0x0]
recursion-via-label
next hop VRF - 'default', table - 0xe0000000
next hop via 24036/0/21
next hop srte_c_10_ep labels imposed {ImplNull 24012}
```

Verifying SR Policy

Use the `show segment-routing traffic-eng policy` command to display SR policy information.

The following outputs show the details of an on-demand SR policy that was triggered by prefixes with color 10 advertised by node 1.1.1.8.

```
RP/0/RP0/CPU0:R4# show segment-routing traffic-eng policy color 10 tabular
```

Segment Routing Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 7.0.x
The following outputs show the details of the on-demand SR policy for BSID 24036.

There are 2 candidate paths associated with this SR policy: the path that is computed by the head-end router (with preference 200), and the path that is computed by the SR-PCE (with preference 100). The candidate path with the highest preference is the active candidate path (highlighted below) and is installed in forwarding.

```
RP/0/RP0/CPU0:R4# show segment-routing traffic-eng policy binding-sid 24036
SR-TE policy database
---------------------
Color: 10, End-point: 1.1.1.8
Name: srte_c_10_ep_1.1.1.8
Status:
  Admin: up  Operational: up for 4d14h (since Jul 3 20:28:57.840)
Candidate-paths:
  Preference: 200 (BGP ODN) (active)
    Requested BSID: dynamic
    PCC info:
      Symbolic name: bgp_c_10_ep_1.1.1.8_discr_200
      PLSF-ID: 12
      Dynamic (valid)
    Metric Type: TE,  Path Accumulated Metric: 30
      16009 [Prefix-SID, 1.1.1.9]
      16008 [Prefix-SID, 1.1.1.8]
  Preference: 100 (BGP ODN)
    Requested BSID: dynamic
    PCC info:
      Symbolic name: bgp_c_10_ep_1.1.1.8_discr_100
      PLSF-ID: 11
      Dynamic (pce 1.1.1.57) (valid)
    Metric Type: TE,  Path Accumulated Metric: 30
      16009 [Prefix-SID, 1.1.1.9]
      16008 [Prefix-SID, 1.1.1.8]
Attributes:
  Binding SID: 24036
  Forward Class: 0
  Steering BGP disabled: no
  IPv6 caps enable: yes
```

**Verifying SR Policy Forwarding**

Use the `show segment-routing traffic-eng forwarding policy` command to display the SR policy forwarding information.

The following outputs show the forwarding details for an on-demand SR policy that was triggered by prefixes with color 10 advertised by node 1.1.1.8.

```
RP/0/RP0/CPU0:R4# show segment-routing traffic-eng forwarding policy binding-sid 24036
tabular
<table>
<thead>
<tr>
<th>Color</th>
<th>Endpoint</th>
<th>Segment List</th>
<th>Outgoing Label</th>
<th>Outgoing Interface</th>
<th>Next Hop</th>
<th>Bytes</th>
<th>Switched</th>
<th>Pure</th>
<th>Backup</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1.1.1.8</td>
<td>dynamic</td>
<td>16009</td>
<td>Gi0/0/0/4</td>
<td>10.4.5.5</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>16001</td>
<td>Gi0/0/0/5</td>
<td>11.4.8.8</td>
<td>0</td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>
```
RP/0/RP0/CPU0:R4# show segment-routing traffic-eng forwarding policy binding-sid 24036 detail
Mon Jul 8 11:56:46.887 PST

SR-TE Policy Forwarding database
--------------------------------
Color: 10, End-point: 1.1.1.8
Name: srte_c_10_ep_1.1.1.8
Binding SID: 24036
Segment Lists:
   SL[0]:
      Name: dynamic
      Paths:
         Path[0]:
            Outgoing Label: 16009
            Outgoing Interface: GigabitEthernet0/0/0/4
            Next Hop: 10.4.5.5
            Switched Packets/Bytes: 0/0
            FRR Pure Backup: No
            Label Stack (Top -> Bottom): { 16009, 16008 }
            Path-id: 1 (Protected), Backup-path-id: 2, Weight: 64
         Path[1]:
            Outgoing Label: 16001
            Outgoing Interface: GigabitEthernet0/0/0/5
            Next Hop: 11.4.8.8
            Switched Packets/Bytes: 0/0
            FRR Pure Backup: Yes
            Label Stack (Top -> Bottom): { 16001, 16009, 16008 }
            Path-id: 2 (Pure-Backup), Weight: 64
      Policy Packets/Bytes Switched: 0/0
      Local label: 80013

Configuring SR-ODN: EVPN Services Examples

Configuring BGP Color Extended Community Set: Example

The following example shows how to configure BGP color extended communities that are later applied to BGP service routes via route-policies.

```plaintext
extcommunity-set opaque color-44
  44
end-set

extcommunity-set opaque color-55
  55
end-set

extcommunity-set opaque color-77
  77
end-set

extcommunity-set opaque color-88
  88
end-set
```

Configuring RPL to Set BGP Color (EVPN Services): Examples

The following examples shows various representative RPL definitions that set BGP color community.

The following RPL examples match on EVPN route-types and then set the BGP color extended community.
route-policy sample-export-rpl
if evpn-route-type is 1 then
    set extcommunity color color-44
endif
if evpn-route-type is 3 then
    set extcommunity color color-55
endif
end-policy

route-policy sample-import-rpl
if evpn-route-type is 1 then
    set extcommunity color color-77
elseif evpn-route-type is 3 then
    set extcommunity color color-88
else
    pass
endif
end-policy

The following RPL example sets BGP color extended community while matching on the following:

- Route Distinguisher (RD)
- Ethernet Segment Identifier (ESI)
- Ethernet Tag (ETAG)
- EVPN route-types

route-policy sample-bgpneighbor-rpl
if rd in (1.1.1.1:3504) then
    set extcommunity color color3504
elseif rd in (1.1.1.1:3505) then
    set extcommunity color color3505
elseif rd in (1.1.1.1:3506) then
    set extcommunity color color99996
elseif esi in (0010.0000.0000.0000.1201) and rd in (1.1.1.1:3508) then
    set extcommunity color color3508
elseif etag in (30509) and rd in (1.1.1.1:3509) then
    set extcommunity color color3509
elseif etag in (0) and rd in (1.1.1.1:2001) and evpn-route-type is 1 then
    set extcommunity color color82001
elseif etag in (0) and rd in (1.1.1.1:2001) and evpn-route-type is 3 then
    set extcommunity color color92001
endif
pass
end-policy

Applying RPL to BGP Services (EVPN Services): Example
The following examples show various RPLs that set BGP color community being applied to EVPN services. The following 2 examples show the RPL applied at the EVI export and import attach-points.

Note
RPLs applied under EVI import or export attach-point also support matching on the following:

- Ethernet Segment Identifier (ESI)
- Ethernet Tag (ETAG)
- EVPN-Originator
evpn
ev1 101
bgp
  route-target 101:1
  route-target import 100:1
  route-target export 101:1
  route-policy import sample-import-rpl
  !
  advertise-mac
  !

! 


evi 102
bgp
  route-target 102:1
  route-target import 100:2
  route-target export 102:1
  route-policy export sample-export-rpl
  !
  advertise-mac
  !

The following example shows the RPL applied at the BGP neighbor-out attach-point.

---

**Note**

RPLs defined under BGP neighbor-out attach-point also support matching on the following:

- **EVPN-Originator**

---

```
router bgp 100
  bgp router-id 1.1.1.1
  address-family 12vpn evpn
  !
  neighbor-group evpn-rr
  remote-as 100
  update-source Loopback0
  address-family 12vpn evpn
  !
  neighbor 10.10.10.10
  use neighbor-group evpn-rr
  address-family 12vpn evpn
  route-policy sample-bgpneighbor-rpl out
```

---

**Configuring SR-ODN for EVPN-VPWS: Use Case**

This use case shows how to set up a pair of ELINE services using EVPN-VPWS between two sites. Services are carried over SR policies that must not share any common links along their paths (link-disjoint). The SR policies are triggered on-demand based on ODN principles. An SR-PCE computes the disjoint paths.

This use case uses the following topology with 2 sites: Site 1 with nodes A and B, and Site 2 with nodes C and D.
Figure 3: Topology for Use Case: SR-ODN for EVPN-VPWS

Table 4: Use Case Parameters

<table>
<thead>
<tr>
<th>IP Addresses of Loopback0 (Lo0) Interfaces</th>
<th>SR-PCE Lo0: 1.1.1.207</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site 1:</td>
<td></td>
</tr>
<tr>
<td>• Node A Lo0: 1.1.1.5</td>
<td></td>
</tr>
<tr>
<td>• Node B Lo0: 1.1.1.6</td>
<td></td>
</tr>
<tr>
<td>Site 2:</td>
<td></td>
</tr>
<tr>
<td>• Node C Lo0: 1.1.1.2</td>
<td></td>
</tr>
<tr>
<td>• Node D Lo0: 1.1.1.4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EVPN-VPWS Service Parameters</th>
<th>ELINE-1:</th>
<th>ELINE-2:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• EVPN-VPWS EVI 100</td>
<td>• EVPN-VPWS EVI 101</td>
</tr>
<tr>
<td></td>
<td>• Node A: AC-ID = 11</td>
<td>• Node B: AC-ID = 12</td>
</tr>
<tr>
<td></td>
<td>• Node C: AC-ID = 21</td>
<td>• Node D: AC-ID = 22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ODN BGP Color Extended Communities</th>
<th>Site 1 routers (Nodes A and B):</th>
<th>Site 2 routers (Nodes C and D):</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• set color 10000</td>
<td>• set color 11000</td>
</tr>
<tr>
<td></td>
<td>• match color 11000</td>
<td>• match color 10000</td>
</tr>
</tbody>
</table>

Note: These colors are associated with the EVPN route-type 1 routes of the EVPN-VPWS services.

<table>
<thead>
<tr>
<th>PCEP LSP Disjoint-Path Association Group ID</th>
<th>Site 1 to Site 2 LSPs (from Node A to Node C/from Node B to Node D):</th>
<th>Site 2 to Site 1 LSPs (from Node C to Node A/from Node D to Node B):</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• group-id = 775</td>
<td>• group-id = 776</td>
</tr>
</tbody>
</table>

The use case provides configuration and verification outputs for all devices.
Configuration: SR-PCE

For cases when PCC nodes support, or signal, PCEP association-group object to indicate the pair of LSPs in a disjoint set, there is no extra configuration required at the SR-PCE to trigger disjoint-path computation.

SR-PCE also supports disjoint-path computation for cases when PCC nodes do not support PCEP association-group object. See Configure the Disjoint Policy (Optional), on page 174 for more information.

Configuration: Site 1 Node A

This section depicts relevant configuration of Node A at Site 1. It includes service configuration, BGP color extended community, and RPL. It also includes the corresponding ODN template required to achieve the disjointness SLA.

Nodes in Site 1 are configured to set color 10000 on originating EVPN routes, while matching color 11000 on incoming EVPN routes from routers located at Site 2.

Since both nodes in Site 1 request path computation from SR-PCE using the same disjoint-path group-id (775), the PCE will attempt to compute disjointness for the pair of LSPs originating from Site 1 toward Site 2.

/* EVPN-VPWS configuration */

interface GigabitEthernet0/0/0/3.2500 12transport
  encapsulation dot1q 2500
  rewrite ingress tag pop 1 symmetric
! l2vpn
  xconnect group evpn_vpws_group
  p2p evpn_vpws_100
    interface GigabitEthernet0/0/0/3.2500
    neighbor evpn evi 100 target 21 source 11
! !
! /* BGP color community and RPL configuration */

  extcommunity-set opaque color=10000
    10000
  end-set
!
  route-policy SET_COLOR_EVPN_VPWS
if evpn-route-type is 1 and rd in (ios-regex '.*.*.*.*:*'(100)') then
    set extcommunity color color-10000
endif
pass
end-policy
!
router bgp 65000
neighbor 1.1.1.253
    address-family l2vpn evpn
        route-policy SET_COLOR_EVPN_VPWS out
    !
!
/* OND template configuration */

segment-routing
    traffic-eng
        on-demand color 11000
            dynamic
                pcep
                !
                metric
                type igp
                !
                disjoint-path group-id 775 type link
                !
                !

Configuration: Site 1 Node B

This section depicts relevant configuration of Node B at Site 1.

/* EVPN-VPWS configuration */

interface TenGigE0/3/0/0/8.2500 12transport
    encapsulation dot1q 2500
    rewrite ingress tag pop 1 symmetric
!
    l2vpn
    xconnect group evpn_vpws_group
    p2p evpn_vpws_101
        interface TenGigE0/3/0/0/8.2500
            neighbor evpn evi 101 target 22 source 12
        !
        !
!
/* BGP color community and RPL configuration */

extcommunity-set opaque color-10000
    10000
end-set

route-policy SET_COLOR_EVPN_VPWS
    if evpn-route-Type is 1 and rd in (ios-regex '.*.*.*.*:*'(101)') then
        set extcommunity color color-10000
    endif
    pass
end-policy
!
router bgp 65000
neighbor 1.1.1.253
  address-family l2vpn evpn
    route-policy SET_COLOR_EVPN_VPWS out
    !
    !
/* ODN template configuration */
segment-routing
  traffic-eng
    on-demand color 11000
      dynamic
        pcep
        !
        metric
          type igp
          !
          disjoint-path group-id 775 type link
          !
          !

Configuration: Site 2 Node C

This section depicts relevant configuration of Node C at Site 2. It includes service configuration, BGP color extended community, and RPL. It also includes the corresponding ODN template required to achieve the disjointness SLA.

Nodes in Site 2 are configured to set color 11000 on originating EVPN routes, while matching color 10000 on incoming EVPN routes from routers located at Site 1.

Since both nodes on Site 2 request path computation from SR-PCE using the same disjoint-path group-id (776), the PCE will attempt to compute disjointness for the pair of LSPs originating from Site 2 toward Site 1.

/* EVPN-VPWS configuration */
interface GigabitEthernet0/0/0/3.2500 l2transport
  encapsulation dot1q 2500
  rewrite ingress tag pop 1 symmetric
  
  l2vpn
    xconnect group evpn_vpws_group
      p2p evpn_vpws_100
        interface GigabitEthernet0/0/0/3.2500
          neighbor evpn evi 100 target 11 source 21
          
          !
          !

  /* BGP color community and RPL configuration */
  extcommunity-set opaque color-11000
    11000
  end-set
  route-policy SET_COLOR_EVPN_VPWS
    if evpn-route-type is 1 and rd in (ios-regex '.*..*..*:({100})') then
      set extcommunity color color-11000
router bgp 65000
neighbor 1.1.1.253
address-family l2vpn evpn
  route-policy SET_COLOR_EVPN_VPWS out
!}

/* ODN template configuration */
segment-routing
  traffic-eng
    on-demand color 10000
dynamic
  pcep
! metric
type 1igp
! disjoint-path group-id 776 type link
!
!

Configuration: Site 2 Node D
This section depicts relevant configuration of Node D at Site 2.

/* EVPN-VPWS configuration */
interface GigabitEthernet0/0/0/1.2500 l2transport
  encapsulation dot1q 2500
  rewrite ingress tag pop 1 symmetric
! l2vpn
  xconnect group evpn_vpws_group
  p2p evpn_vpws_101
  interface GigabitEthernet0/0/0/1.2500
    neighbor evpn evi 101 target 12 source 22
!
!
/* BGP color community and RPL configuration */
  extcommunity-set opaque color-11000
    11000
  end-set
route-policy SET_COLOR_EVPN_VPWS
  if evpn-route-type is 1 and rd in (ios-regex '\\.*\.*\.*\*:(101)') then
    set extcommunity color color-11000
  endif
  pass
end-policy
}

router bgp 65000
neighbor 1.1.1.253
address-family l2vpn evpn
    route-policy SET_COLOR_EVPN_VPWS out
!

/ * ODN template configuration */

segment-routing
    traffic-eng
        on-demand color 10000
        dynamic
        pcep
        !
        metric
        type igp
        !
        disjoint-path group-id 776 type link
!

Verification: SR-PCE

Use the show pce ipv4 peer command to display the SR-PCE’s PCEP peers and session status. SR-PCE
performs path computation for the 4 nodes depicted in the use-case.

RP/0/0/CPU0:SR-PCE# show pce ipv4 peer
Mon Jul 15 19:41:43.622 UTC

PCE's peer database:
---------------------
Peer address: 1.1.1.2
  State: Up
  Capabilities: Stateful, Segment-Routing, Update, Instantiation

Peer address: 1.1.1.4
  State: Up
  Capabilities: Stateful, Segment-Routing, Update, Instantiation

Peer address: 1.1.1.5
  State: Up
  Capabilities: Stateful, Segment-Routing, Update, Instantiation

Peer address: 1.1.1.6
  State: Up
  Capabilities: Stateful, Segment-Routing, Update, Instantiation

Use the show pce association group-id command to display information for the pair of LSPs assigned to a
given association group-id value.

Based on the goals of this use case, SR-PCE computes link-disjoint paths for the SR policies associated with
a pair of ELINE services between site 1 and site 2. In particular, disjoint LSPs from site 1 to site 2 are identified
by association group-id 775. The output includes high-level information for LSPs associated to this group-id:

- At Node A (1.1.1.5): LSP symbolic name = bgp_c_11000_ep_1.1.1.2_discr_100
- At Node B (1.1.1.6): LSP symbolic name = bgp_c_11000_ep_1.1.1.4_discr_100

In this case, the SR-PCE was able to achieve the desired disjointness level; therefore the Status is shown as
"Satisfied".
RP/0/0/CPU0:SR-PCE# show pce association group-id 775
Thu Jul 11 03:52:20.770 UTC

PCE's association database:

Association: Type Link-Disjoint, Group 775, Not Strict
Associated LSPs:
LSP[0]:
  PCC 1.1.1.6, tunnel name bgp_c_11000_ep_1.1.1.4_discr_100, PLSP ID 18, tunnel ID 17, LSP ID 3, Configured on PCC
LSP[1]:
  PCC 1.1.1.5, tunnel name bgp_c_11000_ep_1.1.1.2_discr_100, PLSP ID 18, tunnel ID 18, LSP ID 3, Configured on PCC

Status: Satisfied

Use the show pce lsp command to display detailed information of an LSP present in the PCE's LSP database. This output shows details for the LSP at Node A (1.1.1.5) that is used to carry traffic of EVPN VPWS EVI 100 towards node C (1.1.1.2).

RP/0/0/CPU0:SR-PCE# show pce lsp pcc ipv4 1.1.1.5 name bgp_c_11000_ep_1.1.1.2_discr_100
Thu Jul 11 03:58:45.903 UTC

PCE's tunnel database:

PCC 1.1.1.5:
  Tunnel Name: bgp_c_11000_ep_1.1.1.2_discr_100
  Color: 11000
  Interface Name: srte_c_11000_ep_1.1.1.2

LSPs:
LSP[0]:
  source 1.1.1.5, destination 1.1.1.2, tunnel ID 18, LSP ID 3
  State: Admin up, Operation up
  Setup type: Segment Routing
  Binding SID: 80037
  Maximum SID Depth: 10
  Absolute Metric Margin: 0
  Relative Metric Margin: 0%
  Preference: 100
  Bandwidth: signaled 0 kbps, applied 0 kbps
  PCEP information:
    PLSP-ID 0x12, flags: D:1 S:0 R:0 A:1 O:1 C:0
    LSP Role: Exclude LSP
    State-sync PCE: None
    PCC: 1.1.1.5
    LSP is subdelegated to: None
  Reported path:
    Metric type: IGP, Accumulated Metric 40
    SID[0]: Adj, Label 80003, Address: local 11.5.8.5 remote 11.5.8.8
    SID[1]: Node, Label 16007, Address 1.1.1.7
    SID[2]: Node, Label 16002, Address 1.1.1.2
  Computed path: (Local PCE)
    Computed Time: Thu Jul 11 03:49:48 UTC 2019 (00:08:58 ago)
    Metric type: IGP, Accumulated Metric 40
    SID[0]: Adj, Label 80003, Address: local 11.5.8.5 remote 11.5.8.8
    SID[1]: Node, Label 16007, Address 1.1.1.7
    SID[2]: Node, Label 16002, Address 1.1.1.2
  Recorded path:
    None

Disjoint Group Information:
Type Link-Disjoint, Group 775

This output shows details for the LSP at Node B (1.1.1.6) that is used to carry traffic of EVPN VPWS EVI 101 towards node D (1.1.1.4).
Based on the goals of this use case, SR-PCE computes link-disjoint paths for the SR policies associated with a pair of ELINE services between site 1 and site 2. In particular, disjoint LSPs from site 2 to site 1 are identified by association group-id 776. The output includes high-level information for LSPs associated to this group-id:

- At Node C (1.1.1.2): LSP symbolic name = bgp_c_10000_ep_1.1.1.5_discr_100
- At Node D (1.1.1.4): LSP symbolic name = bgp_c_10000_ep_1.1.1.6_discr_100

In this case, the SR-PCE was able to achieve the desired disjointness level; therefore, the Status is shown as "Satisfied".
PCC 1.1.1.2, tunnel name bgp_c_10000_ep_1.1.1.5_discr_100, PLSP ID 6, tunnel ID 21, LSP ID 3, Configured on PCC

Status: Satisfied

Use the `show pce lsp` command to display detailed information of an LSP present in the PCE's LSP database. This output shows details for the LSP at Node C (1.1.1.2) that is used to carry traffic of EVPN VPWS EVI 100 towards node A (1.1.1.5).

RP/0/0/CP00:SR-PCE# show pce lsp pcc ipv4 1.1.1.2 name bgp_c_10000_ep_1.1.1.5_discr_100
Thu Jul 11 03:55:21.706 UTC

PCE's tunnel database:

---

Tunnel Name: bgp_c_10000_ep_1.1.1.5_discr_100
Color: 10000
Interface Name: srte_c_10000_ep_1.1.1.5

LSPs:

- LSP[0]:
  - source 1.1.1.2, destination 1.1.1.5, tunnel ID 21, LSP ID 3
  - State: Admin up, Operation up
  - Setup type: Segment Routing
  - Binding SID: 80052
  - Maximum SID Depth: 10
  - Absolute Metric Margin: 0
  - Relative Metric Margin: 0%
  - Preference: 100
  - Bandwidth: signaled 0 kbps, applied 0 kbps
  - PCEP information:
    - PLSP-ID 0x6, flags: D:1 S:0 R:0 A:1 O:1 C:0
    - LSP Role: Exclude LSP
    - State-sync PCE: None
    - PCC: 1.1.1.2
    - LSP is subdelegated to: None
  - Reported path:
    - Metric type: IGP, Accumulated Metric 40
    - SID[0]: Node, Label 16007, Address 1.1.1.7
    - SID[1]: Node, Label 16008, Address 1.1.1.8
    - SID[2]: Adj, Label 80005, Address: local 11.5.8.8 remote 11.5.8.5
  - Computed path: (Local PCE)
    - Computed Time: Thu Jul 11 03:50:03 UTC 2019 (00:05:18 ago)
    - Metric type: IGP, Accumulated Metric 40
    - SID[0]: Node, Label 16007, Address 1.1.1.7
    - SID[1]: Node, Label 16008, Address 1.1.1.8
    - SID[2]: Adj, Label 80005, Address: local 11.5.8.8 remote 11.5.8.5
  - Recorded path: None

**Disjoint Group Information:**
Type Link=Disjoint, Group 776

This output shows details for the LSP at Node D (1.1.1.4) used to carry traffic of EVPN VPWS EVI 101 towards node B (1.1.1.6).

RP/0/0/CP00:SR-PCE# show pce lsp pcc ipv4 1.1.1.4 name bgp_c_10000_ep_1.1.1.6_discr_100
Thu Jul 11 03:55:23.296 UTC

PCE's tunnel database:

---

Tunnel Name: bgp_c_10000_ep_1.1.1.6_discr_100
Color: 10000
Interface Name: srte_c_10000_ep_1.1.1.6
LSPs:
LSP[0]:
  source 1.1.1.4, destination 1.1.1.6, tunnel ID 14, LSP ID 1
  State: Admin up, Operation up
  Setup type: Segment Routing
  Binding SID: 80047
  Maximum SID Depth: 10
  Absolute Metric Margin: 0
  Relative Metric Margin: 0%
  Preference: 100
  Bandwidth: signaled 0 kbps, applied 0 kbps

PCEP information:
  FLSP-ID 0x10, flags: D:1 S:0 R:0 A:1 O:1 C:0
  LSP Role: Disjoint LSP
  State-sync PCE: None
  PCC: 1.1.1.4
  LSP is subdelegated to: None

Reported path:
  Metric type: IGP, Accumulated Metric 40
  SID[0]: Node, Label 16001, Address 1.1.1.1
  SID[1]: Node, Label 16006, Address 1.1.1.6

Computed path: (Local PCE)
  Computed Time: Thu Jul 11 03:50:03 UTC 2019 (00:05:20 ago)
  Metric type: IGP, Accumulated Metric 40
  SID[0]: Node, Label 16001, Address 1.1.1.1
  SID[1]: Node, Label 16006, Address 1.1.1.6

Recorded path:
  None

Disjoint Group Information:
  Type Link-Disjoint, Group 776

Verification: Site 1 Node A

This section depicts verification steps at Node A.

Use the `show bgp l2vpn evpn` command to display BGP prefix information for EVPN-VPWS EVI 100 (rd 1.1.1.5:100). The output includes an EVPN route-type 1 route with color 11000 originated at Node C (1.1.1.2).

```
RP/0/RSP0/CPU0:Node-A# show bgp l2vpn evpn rd 1.1.1.5:100
Wed Jul 10 18:57:57.704 PST
BGP router identifier 1.1.1.5, local AS number 65000
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0  RD version: 0
BGP main routing table version 360
BGP NSR Initial initsync version 1 ( Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
  i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
Network  Next Hop  Metric LocPrf Weight Path
Route Distinguisher: 1.1.1.5:100 (default for vrf VPWS:100)
* > [1][0000.0000.0000.0000.0000][21]/120
  0.0.0.0  0 i
* > i[1][0000.0000.0000.0000.0000][21]/120
  1.1.1.2 C:11000  100 0 i
```
The following output displays the details for the incoming EVPN RT1. Note the presence of BGP extended color community 11000, and that the prefix is associated with an SR policy with color 11000 and BSID value of 80044.

```
RP/0/RSP0/CPU0:Node-A# show bgp l2vpn evpn rd 1.1.1.5:100
[1][0000.0000.0000.0000.0000][21]/120
Wed Jul 10 18:57:58.107 PST
BGP routing table entry for [1][0000.0000.0000.0000.0000][21]/120, Route Distinguisher: 1.1.1.5:100
Versions:
  Process  bRIB/RIB  SendTblVer
  Speaker  360  360
Last Modified: Jul 10 18:36:18.369 for 00:21:40
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
Local 1.1.1.2 C:11000 (bsid:80044) (metric 40) from 1.1.1.253 (1.1.1.2)
  Received Label 80056
  Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate, imported, rib-install
  Received Path ID 0, Local Path ID 1, version 358
  Extended community: Color:11000
  RT:65000:100
  Originator: 1.1.1.2, Cluster list: 1.1.1.253
  SR policy color 11000, up, registered, bsid 80044, if-handle 0x00001b20
  Source AFI: L2VPN EVPN, Source VRF: default, Source Route Distinguisher: 1.1.1.2:100
```

Use the `show l2vpn xconnect` command to display the state associated with EVPN-VPWS EVI 100 service.

```
RP/0/RSP0/CPU0:Node-A# show l2vpn xconnect group evpn_vpws_group
Wed Jul 10 18:58:02.333 PST
XConnect Segment 1 Segment 2
Group Name ST Description ST Description ST
---------------------------------------------

evpn_vpws_group evpn_vpws_100
  UP Gi0/0/0/3.2500
  EVPN 100,21,1.1.1.2
```

The following output shows the details for the service. Note that the service is associated with the on-demand SR policy with color 11000 and end-point 1.1.1.2 (node C).

```
RP/0/RSP0/CPU0:Node-A# show l2vpn xconnect group evpn_vpws_group xc-name evpn_vpws_100 detail
Wed Jul 10 18:58:02.755 PST
Group evpn_vpws_group, XC evpn_vpws_100, state is up; Interworking none
  AC: GigabitEthernet0/0/0/3.2500, state is up
  Type VLAN; Num Ranges: 1
  Rewrite Tags: []
  VLAN ranges: [2500, 2500]
  MTU 1500; XC ID 0x120000c; interworking none
  Statistics:
    packets: received 0, sent 0
    bytes: received 0, sent 0
    drops: illegal VLAN 0, illegal length 0
  EVPN: neighbor 1.1.1.2, FW ID: evi 100, ac-id 21, state is up (established)
```
Encapsulation MPLS
Source address 1.1.1.5
Encap type Ethernet, control word enabled
Sequencing not set
Preferred path Active: SR TE srte_c_11000_ep_1.1.1.2, On-Demand, fallback enabled
Tunnel: Up
Load Balance Hashing: src-dst-mac

<table>
<thead>
<tr>
<th>EVPN</th>
<th>Local</th>
<th>Remote</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>80040</td>
<td>80056</td>
</tr>
<tr>
<td>MTU</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>Control word enabled</td>
<td>enabled</td>
<td></td>
</tr>
<tr>
<td>AC ID</td>
<td>11</td>
<td>21</td>
</tr>
<tr>
<td>EVPN type</td>
<td>Ethernet</td>
<td>Ethernet</td>
</tr>
</tbody>
</table>

Create time: 10/07/2019 18:31:30 (1d17h ago)
Last time status changed: 10/07/2019 19:42:00 (1d16h ago)
Last time PW went down: 10/07/2019 19:40:55 (1d16h ago)
Statistics:
- packets: received 0, sent 0
- bytes: received 0, sent 0

Use the show segment-routing traffic-eng policy command with tabular option to display SR policy summary information.

The following output shows the on-demand SR policy with BSID 80044 that was triggered by EVPN RT1 prefix with color 11000 advertised by node C (1.1.1.2).

RP/0/RSP0/CPU0:Node-A# show segment-routing traffic-eng policy color 11000 tabular

<table>
<thead>
<tr>
<th>Color</th>
<th>Endpoint</th>
<th>Admin State</th>
<th>Oper State</th>
<th>Binding SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>11000</td>
<td>1.1.1.2</td>
<td>up</td>
<td>up</td>
<td>80044</td>
</tr>
</tbody>
</table>

The following output shows the details for the on-demand SR policy. Note that the SR policy's active candidate path (preference 100) is computed by SR-PCE (1.1.1.207).

Based on the goals of this use case, SR-PCE computes link-disjoint paths for the SR policies associated with a pair of ELINE services between site 1 and site 2. Specifically, from site 1 to site 2, LSP at Node A (srte_c_11000_ep_1.1.1.2) is link-disjoint from LSP at Node B (srte_c_11000_ep_1.1.1.4).

RP/0/RSP0/CPU0:Node-A# show segment-routing traffic-eng policy color 11000

SR-TE policy database

<table>
<thead>
<tr>
<th>Color</th>
<th>Endpoint</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>11000</td>
<td>1.1.1.2</td>
<td>up</td>
</tr>
</tbody>
</table>

Name: srte_c_11000_ep_1.1.1.2
Status:
- Admin: up
- Operational: up for 00:39:31 (since Jul 10 18:36:00.471)
Candidate-paths:
- Preference: 200 (BGP ODN) (shutdown)
- Requested BSID: dynamic
- PCC info:
  Symbolic name: bgp_c_11000_ep_1.1.1.2_discr_200
  PLSP-ID: 19
**Preference:** 100 (BGP ODN) (active)

Requested BSID: dynamic

PCC info:
Symbolic name: bgp_c_11000_ep_1.1.1.2_discr_100
PLSP-ID: 18

**Dynamic (pce 1.1.1.207) (valid)**

Metric Type: IGP, Path Accumulated Metric: 40
80003 [Adjacency-SID, 11.5.8.5 - 11.5.8.8]
16007 [Prefix-SID, 1.1.1.7]
16002 [Prefix-SID, 1.1.1.2]

Attributes:
- **Binding SID:** 80044
- Forward Class: 0
- Steering BGP disabled: no
- IPv6 caps enable: yes

**Verification: Site 1 Node B**

This section depicts verification steps at Node B.

Use the `show bgp l2vpn evpn` command to display BGP prefix information for EVPN-VPWS EVI 101 (rd 1.1.1.6:101). The output includes an EVPN route-type 1 route with color 11000 originated at Node D (1.1.1.4).

```
RP/0/RSP0/CPU0:Node-B# show bgp l2vpn evpn rd 1.1.1.6:101
Wed Jul 10 19:08:54.964 PST
BGP router identifier 1.1.1.6, local AS number 65000
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0 RD version: 0
BGP main routing table version 322
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
Origin codes: i - IGP, e - EGP, ? - incomplete
Network Metric LocPrf Weight Path
Route Distinguisher: 1.1.1.6:101 (default for vrf VPWS:101)

*> [1][0000.0000.0000.0000.0000][12]/120
  0.0.0.0  i
*>[1][0000.0000.0000.0000.0000][22]/120
  1.1.1.4 C:11000

processed 2 prefixes, 2 paths
```

The following output displays the details for the incoming EVPN RT1. Note the presence of BGP extended color community 11000, and that the prefix is associated with an SR policy with color 11000 and BSID value of 80061.

```
RP/0/RSP0/CPU0:Node-B# show bgp l2vpn evpn rd 1.1.1.6:101
[1][0000.0000.0000.0000.0000][22]/120
Wed Jul 10 19:08:55.039 PST
BGP routing table entry for [1][0000.0000.0000.0000.0000][22]/120, Route Distinguisher: 1.1.1.6:101
Versions:
  Process bRIB/RIB SendTblVer
  Speaker 322 322
Last Modified: Jul 10 18:42:10.408 for 00:26:44
Path: (1 available, best #1)
```

Segment Routing Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 7.0.x
Not advertised to any peer
Path #1: Received by speaker 0
Not advertised to any peer
Local 1.1.1.4 C:11000 (bsid:80061) (metric 40) from 1.1.1.253 (1.1.1.4)
  Received Label 80045
  Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate, imported, rib-install
  Received Path ID 0, Local Path ID 1, version 319
  Extended community: Color:11000 RT:65000:101
  Originator: 1.1.1.4, Cluster list: 1.1.1.253
  SR policy color 11000, up, registered, bsid 80061, if-handle 0x000000560
  Source AFI: L2VPN EVPN, Source VRF: default, Source Route Distinguisher: 1.1.1.4:101

Use the `show l2vpn xconnect` command to display the state associated with EVPN-VPWS EVI 101 service.

<table>
<thead>
<tr>
<th>XConnect</th>
<th>Name</th>
<th>ST</th>
<th>Segment 1 Description</th>
<th>ST</th>
<th>Segment 2 Description</th>
<th>ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>evpn_vpws_group</td>
<td>evpn_vpws_101</td>
<td>UP</td>
<td>Te0/3/0/0/8.2500</td>
<td>UP</td>
<td>EVPN 101,22,1.1.1.4</td>
<td>UP</td>
</tr>
</tbody>
</table>

The following output shows the details for the service. Note that the service is associated with the on-demand SR policy with color 11000 and end-point 1.1.1.4 (node D).

| Group evpn_vpws_group, XC evpn_vpws_101, state is up; Interworking none
| AC: TenGigE0/3/0/0/8.2500, state is up
| Type VLAN; Num Ranges: 1
| Rewrite Tags: []
| VLAN ranges: [2500, 2500]
| MTU 1500; XC ID 0x2a000000; interworking none
| Statistics:
| packets: received 0, sent 0
| bytes: received 0, sent 0
| drops: illegal VLAN 0, illegal length 0
| EVPN: neighbor 1.1.1.4, PW ID: evi 101, ac-id 22, state is up (established)
| XC ID 0xa0000000
| Encapsulation MPLS
| Source address 1.1.1.6
| Encap type Ethernet, control word enabled
| Sequencing not set
| Preferred path Active: SR TE srte_c_11000_ep_1.1.1.4, On-Demand, fallback enabled
| Tunnel: Up
| Load Balance Hashing: src-dst-mac

<table>
<thead>
<tr>
<th>EVPN</th>
<th>Local</th>
<th>Remote</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>80060</td>
<td>80045</td>
</tr>
<tr>
<td>MTU</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>Control word enabled</td>
<td>enabled</td>
<td></td>
</tr>
<tr>
<td>AC ID</td>
<td>12</td>
<td>22</td>
</tr>
<tr>
<td>EVPN type</td>
<td>Ethernet</td>
<td>Ethernet</td>
</tr>
</tbody>
</table>
Use the `show segment-routing traffic-eng policy` command with the `tabular` option to display SR policy summary information.

The following output shows the on-demand SR policy with BSID 80061 that was triggered by EVPN RT1 prefix with color 11000 advertised by node D (1.1.1.4).

```
RP/0/RSP0/CPU0:Node-B# show segment-routing traffic-eng policy color 11000 tabular
Wed Jul 10 19:08:56.146 PST
Color Endpoint Admin Oper Binding SID
------ -------------------- ------ ------ --------------------
11000 1.1.1.4 up up 80061
```

The following output shows the details for the on-demand SR policy. Note that the SR policy’s active candidate path (preference 100) is computed by SR-PCE (1.1.1.207).

```
RP/0/RSP0/CPU0:Node-B# show segment-routing traffic-eng policy color 11000
Wed Jul 10 19:08:56.207 PST
SR-TE policy database
Name: srte_c_11000_ep_1.1.1.4
Status: Admin: up Operational: up for 00:26:47 (since Jul 10 18:40:05.868)
Candidate-paths:
Preferencce: 200 (BGP ODN) (shutdown)
Requested BSID: dynamic
PCC info:
Symbolic name: bgp_c_11000_ep_1.1.1.4_discr_200
PLSP-ID: 19
Dynamic (invalid)
Preferencce: 100 (BGP ODN) (active)
Requested BSID: dynamic
PCC info:
Symbolic name: bgp_c_11000_ep_1.1.1.4_discr_100
PLSP-ID: 18
Dynamic (pce 1.1.1.207) (valid)
Metric Type: IGP, Path Accumulated Metric: 40
16001 [Prefix-SID, 1.1.1.1]
16004 [Prefix-SID, 1.1.1.4]
Attributes:
Binding SID: 80061
Forward Class: 0
Steering BGP disabled: no
IPv6 caps enable: yes
```
Verification: Site 2 Node C

This section depicts verification steps at Node C.

Use the `show bgp l2vpn evpn` command to display BGP prefix information for EVPN-VPWS EVI 100 (rd 1.1.1.2:100). The output includes an EVPN route-type 1 route with color 10000 originated at Node A (1.1.1.5).

```
RP/0/RSP0/CPU0:Node-C# show bgp l2vpn evpn rd 1.1.1.2:100
BGP router identifier 1.1.1.2, local AS number 65000
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0  RD version: 0
BGP main routing table version 21
BGP NSR Initial initiSync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
   i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
  Network    Next Hop    Metric  LocPrf  Weight  Path
Route Distinguisher: 1.1.1.2:100 (default for vrf VPWS:100)
  *> [1][0000.0000.0000.0000.0000][11]/120
     1.1.1.5 C:10000 100 0 i
  *> [1][0000.0000.0000.0000.0000][21]/120
     0.0.0.0 0 i
```

The following output displays the details for the incoming EVPN RT1. Note the presence of BGP extended color community 10000, and that the prefix is associated with an SR policy with color 10000 and BSID value of 80058.

```
RP/0/RSP0/CPU0:Node-C# show bgp l2vpn evpn rd 1.1.1.2:100 [1][0000.0000.0000.0000.0000][11]/120
BGP routing table entry for [1][0000.0000.0000.0000.0000][11]/120, Route Distinguisher: 1.1.1.2:100
Versions:
  Process bRIB/RIB SendTblVer
  Speaker 20 20
Last Modified: Jul 10 18:36:20.503 for 00:45:21
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
Local
  1.1.1.5 C:10000 (bsid:80058) (metric 40) from 1.1.1.253 (1.1.1.5)
  Received label 80040
  Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate, imported, rib-install
  Received Path ID 0, Local Path ID 1, version 18
  Extended community: Color:10000 RT:65000:100
  Originator: 1.1.1.5, Cluster list: 1.1.1.253
  SR policy color 10000, up, registered, bsid 80058, if-handle 0x0000006a0
  Source AFI: L2VPN EVPN, Source VRF: default, Source Route Distinguisher: 1.1.1.5:100
```

Use the `show l2vpn xconnect` command to display the state associated with EVPN-VPWS EVI 100 service.

```
RP/0/RSP0/CPU0:Node-C# show l2vpn xconnect group evpn_vpws_group
XConnect    Segment 1    Segment 2
```

Configure SR-TE Policies

Configuring SR-ODN for EVPN-VPWS: Use Case
The following output shows the details for the service. Note that the service is associated with the on-demand SR policy with color 10000 and end-point 1.1.1.5 (node A).

RP/0/RSP0/CPU0:Node-C# show l2vpn xconnect group evpn_vpws_group xc-name evpn_vpws_100

Group evpn_vpws_group, XC evpn_vpws_100, state is up; Interworking none
AC: GigabitEthernet0/0/0/3.2500, state is up
Type VLAN; Num Ranges: 1
Rewrite Tags: []
VLAN ranges: [2500, 2500]
MTU 1500; XC ID 0x1200008; interworking none
Statistics:
packets: received 0, sent 0
bytes: received 0, sent 0
drops: illegal VLAN 0, illegal length 0
EVPN: neighbor 1.1.1.5, PW ID: evi 100, ac-id 11, state is up (established)
XC ID 0xa0000003
Encapsulation MPLS
Source address 1.1.1.2
Encap type Ethernet, control word enabled
Sequencing not set
Preferred path Active: SR TE srte_c_10000_ep_1.1.1.5, On-Demand, fallback enabled
Tunnel: Up
Load Balance Hashing: src-dst-mac
EVPN Local Remote
------------------ ------------------
Label 80056 80040
MTU 1500 1500
Control word enabled enabled
AC ID 21 11
EVPN type Ethernet Ethernet

Create time: 10/07/2019 18:36:16 (1d19h ago)
Last time status changed: 10/07/2019 19:41:59 (1d18h ago)
Last time PW went down: 10/07/2019 19:40:54 (1d18h ago)
Statistics:
packets: received 0, sent 0
bytes: received 0, sent 0

Use the `show segment-routing traffic-eng policy` command with `tabular` option to display SR policy summary information.

The following output shows the on-demand SR policy with BSID 80058 that was triggered by EVPN RT1 prefix with color 10000 advertised by node A (1.1.1.5).

RP/0/RSP0/CPU0:Node-C# show segment-routing traffic-eng policy color 10000 tabular

<table>
<thead>
<tr>
<th>Color</th>
<th>Endpoint</th>
<th>Admin State</th>
<th>Oper State</th>
<th>Binding SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>1.1.1.5</td>
<td>up</td>
<td>up</td>
<td>80058</td>
</tr>
</tbody>
</table>
The following output shows the details for the on-demand SR policy. Note that the SR policy's active candidate path (preference 100) is computed by SR-PCE (1.1.1.207).

Based on the goals of this use case, SR-PCE computes link-disjoint paths for the SR policies associated with a pair of ELINE services between site 1 and site 2. Specifically, from site 2 to site 1, LSP at Node C (srte_c_10000_ep_1.1.1.5) is link-disjoint from LSP at Node D (srte_c_10000_ep_1.1.1.6).

RP/0/RSP0/CPU0:Node-C# show segment-routing traffic-eng policy color 10000
SR-TE policy database
---------------------
Color: 10000, End-point: 1.1.1.5
Name: srte_c_10000_ep_1.1.1.5
Status:
  Admin: up Operational: up for 00:12:35 (since Jul 10 19:49:21.890)
Candidate-paths:
  Preference: 200 (BGP ODN) (shutdown)
  Requested BSID: dynamic
  PCC info:
    Symbolic name: bgp_c_10000_ep_1.1.1.5_discr_200
    PLSP-ID: 7
    Dynamic (invalid)
  Preference: 100 (BGP ODN) (active)
  Requested BSID: dynamic
  PCC info:
    Symbolic name: bgp_c_10000_ep_1.1.1.5_discr_100
    PLSP-ID: 6
    Dynamic (pce 1.1.1.207) (valid)
    Metric Type: IGP, Path Accumulated Metric: 40
    16007 [Prefix-SID, 1.1.1.7]
    16008 [Prefix-SID, 1.1.1.8]
    80005 [Adjacency-SID, 11.5.8.8 - 11.5.8.5]
Attributes:
  Binding SID: 80058
  Forward Class: 0
  Steering BGP disabled: no
  IPv6 caps enable: yes

Verification: Site 2 Node D

This section depicts verification steps at Node D.

Use the show bgp l2vpn evpn command to display BGP prefix information for EVPN-VPWS EVI 101 (rd 1.1.1.4:101). The output includes an EVPN route-type 1 route with color 10000 originated at Node B (1.1.1.6).

RP/0/RSP0/CPU0:Node-D# show bgp l2vpn evpn rd 1.1.1.4:101
BGP router identifier 1.1.1.4, local AS number 65000
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0  RD version: 0
BGP main routing table version 570
BGP NSF Initial intsync version 1 (Reached)
BGP NSF/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs
Status codes: s suppressed, d damped, h history, * valid, > best
  i = internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i = IGP, e = EGP, ? = incomplete
Network Next Hop Metric LocPrf Weight Path
Route Distinguisher: 1.1.1.4:101 (default for vrf VPWS:101)
*>>1[1][0000.0000.0000.0000]:10000[12]/120
The following output displays the details for the incoming EVPN RT1. Note the presence of BGP extended color community 10000, and that the prefix is associated with an SR policy with color 10000 and BSID value of 80047.

```
RP/0/RSP0/CPU0:Node-D# show bgp l2vpn evpn rd 1.1.1.4:101
[1][0000.0000.0000.0000.0000][12]/120
BGP routing table entry for [1][0000.0000.0000.0000.0000][12]/120, Route Distinguisher: 1.1.1.4:101
Versions:
  Process  bRIB/RIB  SendTblVer
  Speaker  569      569
Last Modified: Jul 10 18:42:12.455 for 00:45:38
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
Local
1.1.1.6 C:10000 (bsid:80047) (metric 40) from 1.1.1.253 (1.1.1.6)
  Received Label 80060
  Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate, imported, rib-install
  Received Path ID 0, Local Path ID 1, version 568
  Extended community: Color:10000 RT:65000:101
  Originator: 1.1.1.6, Cluster list: 1.1.1.253
  SR policy color 10000, up, registered, bsid 80047, if-handle 0x00001720
Source AFI: L2VPN EVPN, Source VRF: default, Source Route Distinguisher: 1.1.1.6:101
```

Use the `show l2vpn xconnect` command to display the state associated with EVPN-VPWS EVI 101 service.

```
RP/0/RSP0/CPU0:Node-D# show l2vpn xconnect group evpn_vpws_group

XConnect Group Name ST  Segment 1 Description ST  Segment 2 Description ST
-----------------------------------------
  evpn_vpws_group
  evpn_vpws_101
     UP  Gi0/0/0/1.2500  UP  EVPN 101,12,1.1.1.6  UP
```

The following output shows the details for the service. Note that the service is associated with the on-demand SR policy with color 10000 and end-point 1.1.1.6 (node B).

```
RP/0/RSP0/CPU0:Node-D# show l2vpn xconnect group evpn_vpws_group xc-name evpn_vpws_101
Group evpn_vpws_group, XC evpn_vpws_101, state is up; Interworking none
  AC: GigabitEthernet0/0/0/1.2500, state is up
  Type VLAN; Num Ranges: 1
  Rewrite Tags: []
  VLAN ranges: [2500, 2500]
  MTU 1500; XC ID 0x120000c; interworking none
  Statistics:
    packets: received 0, sent 0
    bytes: received 0, sent 0
```
drops: illegal VLAN 0, illegal length 0
EVPN: neighbor 1.1.1.6, PW ID: evi 101, ac-id 12, state is up (established)
XC ID 0xa000000d
Encapsulation MPLS
Source address 1.1.1.4
Encap type Ethernet, control word enabled
Sequencing not set
Preferred path Active: SR TE srte_c_10000_ep_1.1.1.6, On-Demand, fallback enabled
Tunnel: Up
Load Balance Hashing: src-dst-mac

<table>
<thead>
<tr>
<th>EVPN</th>
<th>Local</th>
<th>Remote</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>80045</td>
<td>80060</td>
</tr>
<tr>
<td>MTU</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>Control word enabled</td>
<td>enabled</td>
<td>enabled</td>
</tr>
<tr>
<td>AC ID</td>
<td>22</td>
<td>12</td>
</tr>
<tr>
<td>EVPN type</td>
<td>Ethernet</td>
<td>Ethernet</td>
</tr>
</tbody>
</table>

Create time: 10/07/2019 18:42:07 (00:45:49 ago)
Last time status changed: 10/07/2019 18:42:09 (00:45:47 ago)
Statistics:
  packets: received 0, sent 0
  bytes: received 0, sent 0

Use the `show segment-routing traffic-eng policy` command with `tabular` option to display SR policy summary information.

The following output shows the on-demand SR policy with BSID 80047 that was triggered by EVPN RT1 prefix with color 10000 advertised by node B (1.1.1.6).

```
RP/0/RSP0/CPU0:Node-D# show segment-routing traffic-eng policy color 10000 tabular

<table>
<thead>
<tr>
<th>Color</th>
<th>Endpoint</th>
<th>Admin State</th>
<th>Oper State</th>
<th>Binding SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>1.1.1.6</td>
<td>up</td>
<td>up</td>
<td>80047</td>
</tr>
</tbody>
</table>
```

The following output shows the details for the on-demand SR policy. Note that the SR policy's active candidate path (preference 100) is computed by SR-PCE (1.1.1.207).

Based on the goals of this use case, SR-PCE computes link-disjoint paths for the SR policies associated with a pair of ELINE services between site 1 and site 2. Specifically, from site 2 to site 1, LSP at Node D (srte_c_10000_ep_1.1.1.6) is link-disjoint from LSP at Node C (srte_c_10000_ep_1.1.1.5).

```
RP/0/RSP0/CPU0:Node-D# show segment-routing traffic-eng policy color 10000

SR-TE policy database

Color: 10000, End-point: 1.1.1.6
Name: srte_c_10000_ep_1.1.1.6
Status:
  Admin: up  Operational: up for 01:23:04 (since Jul 10 18:42:07.350)
Candidate-paths:
  Preference: 200 (BGP ODN) (shutdown)
  Requested BSID: dynamic
  PCC info:
    Symbolic name: bgp_c_10000_ep_1.1.1.6_discr_200
    PLSP-ID: 17
    Dynamic (invalid)
```
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 112 section for information on manually provisioning an SR policy using dynamic or explicit paths.

PCE-Initiated SR Policy

An SR-TE policy can be configured on the path computation element (PCE) to reduce link congestion or to minimize the number of network touch points.

The PCE collects network information, such as traffic demand and link utilization. When the PCE determines that a link is congested, it identifies one or more flows that are causing the congestion. The PCE finds a suitable path and deploys an SR-TE policy to divert those flows, without moving the congestion to another part of the network. When there is no more link congestion, the policy is removed.

To minimize the number of network touch points, an application, such as a Network Services Orchestrator (NSO), can request the PCE to create an SR-TE policy. PCE deploys the SR-TE policy using PCC-PCE communication protocol (PCEP).

For more information, see the PCE-Initiated SR Policies for Traffic Management, on page 175 section.

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.
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 Cisco ASR 9000 Series Routers*.

- TE metric — See the Configure Interface TE Metrics, on page 114 section for information about configuring TE metrics.

- Delay — See the Configure Performance Measurement, on page 181 chapter for information about measuring delay for links or SR policies.

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

- 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 114 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).

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 256 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.
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.

```bash
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
```

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.

```bash
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.

```bash
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
```
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 113 section.
2. Define the constraints. See the Constraints, on page 114 section.
3. Create the policy.

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.

```conf
segment-routing
traffic-eng
policy foo
color 100 end-point ipv4 1.1.1.2
candidate-paths
preference 100
dynamic
metric
type te
!
constraints
affinity
exclude-any
name RED
!
!
!
!
!
!

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 SR-PCE.

```conf
segment-routing
traffic-eng
policy baa
color 101 end-point ipv4 1.1.1.2
candidate-paths
preference 100
dynamic
pcep
!
metric
type te
!
constraints
affinity
exclude-any
name BLUE
!
!
!```
**Anycast SID-Aware Path Computation**

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, providing load-balancing and redundancy. Packets addressed to an Anycast address are forwarded to the topologically nearest nodes.

---

**Note**

For information on configuring Anycast SID, see Configuring a Prefix-SID on the IS-IS Enabled Loopback Interface, on page 44 and Configuring a Prefix-SID on the OSPF-Enabled Loopback Interface, on page 63.

This example shows how Anycast SIDs are inserted into a computed SID list.

The following figure shows 3 isolated IGP domains without redistribution and without BGP 3107. Each Area Border Router (ABR) 1 through 4 is configured with a node SID. ABRs 1 and 2 share Anycast SID 16012 and ABRs 3 and 4 share Anycast SID 16034.

Consider the case where routers A and Z are provider edge (PE) routers in the same VPN. Router A receives a VPN route with BGP next-hop to router Z. Router A resolves the SR path to router Z using SR-ODN or SR-PCE.

Before considering Anycast SIDs, the head-end router or SR-PCE computes the SID list.
In this case, the optimized computed path from router A to router Z is 16002 > 16004 > 1600Z.

The path computation process reiterates the original SID-list and replaces node SIDs with Anycast SIDs (when possible). SR-TE verifies that the Anycast-encoded SID list maintains an optimum path and does not violate any path constraints (link affinity, metric bounds). If the SID list is verified, then the Anycast-encoded SID list is signaled and instantiated in the forwarding.

Using the Anycast-encoded SID list, the optimized computed path from router A to router Z is 16012 > 16034 > 1600Z. The Anycast SID-aware path computation provides load-balancing.

The Anycast SID aware path computation also provides resiliency. For example, if one of the ABRs (in this case, ABR 1) becomes unavailable or unreachable, the path from router A to router Z (16012 > 16034 > 1600Z) will still be valid and usable.
**Configuration Examples**

1. Configure Prefix SIDs on the ABR nodes.
   a. Configure each node with a node SID.
   b. Configure each group of nodes with a shared Anycast SID.

   See Configuring a Prefix-SID on the IS-IS Enabled Loopback Interface, on page 44 and Configuring a Prefix-SID on the OSPF-Enabled Loopback Interface, on page 63.

2. Configure SR policies to include Anycast SIDs for path computation using the **anycast-sid-inclusion** command.

   This example shows how to configure a local SR policy to include Anycast SIDs for PCC-initiated path computation at the head-end router:

   ```
   Router(config)# segment-routing traffic-eng
   Router(config-sr-te)# policy FOO
   Router(config-sr-te-policy)# color 10 end-point ipv4 1.1.1.10
   Router(config-sr-te-policy)# candidate-paths
   Router(config-sr-te-policy-path)# preference 100
   Router(config-sr-te-policy-path-pref)# dynamic
   Router(config-sr-te-pp-info)# anycast-sid-inclusion
   ```

**Running Configuration**

Use the **anycast-sid-inclusion** command to include Anycast SIDs into the computed paths of the following policy types:

- **Local SR policy with PCC-initiated path computation at the head-end router:**

  ```
  segment-routing
  traffic-eng
  policy FOO
  color 10 end-point ipv4 1.1.1.10
  candidate-paths
  preference 100
  dynamic
  anycast-sid-inclusion
  ```

- **Local SR policy with PCC-initiated/PCE-delegated path computation at the SR-PCE:**

  ```
  segment-routing
  traffic-eng
  policy BAR
  color 20 end-point ipv4 1.1.1.20
  candidate-paths
  preference 100
  dynamic
  pcep
  anycast-sid-inclusion
  ```

- **On-demand SR policies with a locally computed dynamic path at the head-end, or centrally computed dynamic path at the SR-PCE:**

  ```
  segment-routing
  traffic-eng
  on-demand color 10
  dynamic
  anycast-sid-inclusion
  ```
- On-demand SR policies with centrally computed dynamic path at the SR-PCE:

```plaintext
segment-routing
traffic-eng
on-demand color 20
dynamic
  pcep
  anycast-sid-inclusion
```

## Explicit Paths

### Configure SR-TE Policy with Explicit Path

To configure a SR-TE policy with an explicit path, complete the following configurations:

1. Create the segment lists. A segment list can use IP addresses or MPLS labels, or a combination of both.

   - **Note**
   - A segment list can use both IP addresses and MPLS labels, but once you enter an MPLS label, you cannot enter an IP address.

2. Create the SR-TE policy.

### 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:
Configure SR-TE Policies

```
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
```
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
Configure SR-TE Policies

Configuring Explicit Path with Affinity Constraint Validation

Running Configuration

Router(config-sr-te-if-affinity)# blue
Router(config-sr-te-if-affinity)# 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-policy-path-pref)# explicit segment-list SIDLIST2
Router(config-sr-te-policy-path-pref)# explicit segment-list SIDLIST3

Router(config-sr-te)#

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
    !
Explicit Path with Affinity Constraint Validation for Anycast SIDs

For information about configuring Anycast SIDs, see Configuring a Prefix-SID on the IS-IS Enabled Loopback Interface, on page 44 or Configuring a Prefix-SID on the OSPF-Enabled Loopback Interface, on page 63.

Routers that are configured with the same Anycast SID, on the same Loopback address and with the same SRGB, advertise the same prefix SID (Anycast).

The shortest path with the lowest IGP metric is then verified against the affinity constraints. If multiple nodes have the same shortest-path metric, all their paths are validated against the affinity constraints. A path that is not the shortest path is not validated against the affinity constraints.
**Affinity Support for Anycast SIDs: Examples**

In the following examples, nodes 3 and 5 advertise the same Anycast prefix (1.1.1.8) and assign the same prefix SID (16100).

Node 1 uses the following SR-TE policy:

```
segment-routing
  traffic-eng
  policy POLICY1
    color 20 end-point ipv4 1.1.1.4
    binding-sid mpls 1000
    candidate-paths
      preference 100
      explicit segment-list SIDLIST1
      constraints
        affinity
          exclude-any
          red
    segment-list name SIDLIST1
      index 10 address ipv4 100.100.100.100
      index 20 address ipv4 4.4.4.4
```

**Affinity Constraint Validation With ECMP Anycast SID: Example**

In this example, the shortest path to both node 3 and node 5 has an equal accumulative IGP metric of 20. Both paths are validated against affinity constraints.
Candidate-paths:
Preference 100:
Constraints:
Affinity:
exclude-any: red
Explicit: segment-list SIDLIST1 (active)
Weight: 0, Metric Type: IGP
  16100 [Prefix-SID, 1.1.1.8]
  16004 [Prefix-SID, 4.4.4.4]

Affinity Constraint Validation With Non-ECMP Anycast SID: Example

In this example, the shortest path to node 5 has an accumulative IGP metric of 20, and the shortest path to node 3 has an accumulative IGP metric of 30. Only the shortest path to node 5 is validated against affinity constraints.

Even though parallel link (23) is marked with red, it is still considered valid since anycast traffic flows only on the path to node 5.

Invalid Path Based on Affinity Constraint: Example

In this example, parallel link (23) is marked as red, so the path to anycast node 3 is invalidated.

SR-TE policy database
----------------------
Name: POLICY1 (Color: 2, End-point: 198.51.100.6)
Status:
Admin: up Operational: up for 00:03:52 (since Jan 24 01:52:14.215)
Candidate-paths:
Preference 100:
Constraints:
  Affinity:
    exclude-any: red
  Explicit: segment-list SIDLIST1 (inactive)
  Inactive Reason: Link [2.2.21.23,2.2.21.32] failed to satisfy affinity exclude-any constraint=0x00000008, link attributes=0x0000000A

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.

Configure the Head-End Router as PCEP PCC

Configure the head-end router as PCEP Path Computation Client (PCC) to establish a connection to the PCE. The PCC and PCE addresses must be routable so that TCP connection (to exchange PCEP messages) can be established between PCC and PCE.

Configure the PCC to Establish a Connection to the PCE

Use the `segment-routing traffic-eng pcc` command to configure the PCC source address, the SR-PCE address, and SR-PCE options.

A PCE can be given an optional precedence. If a PCC is connected to multiple PCEs, the PCC selects a PCE with the lowest precedence value. If there is a tie, a PCE with the highest IP address is chosen for computing path. The precedence value range is from 0 to 255.

```
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# pcc
Router(config-sr-te-pcc)# source-address ipv4 local-source-address
Router(config-sr-te-pcc)# pce address ipv4 PCE-address[precedence value]
Router(config-sr-te-pcc)# pce address ipv4 PCE-address[keychain WORD]
```

Configure PCEP-Related Timers

Use the `timers keepalive` command to specify how often keepalive messages are sent from PCC to its peers. The range is from 0 to 255 seconds; the default value is 30.

```
Router(config-sr-te-pcc)# timers keepalive seconds
```
Use the `timers deadtimer` command to specify how long the remote peers wait before bringing down the PCEP session if no PCEP messages are received from this PCC. The range is from 1 to 255 seconds; the default value is 120.

```
Router(config-sr-te-pcc)# timers deadtimer seconds
```

Use the `timers delegation-timeout` command to specify how long a delegated SR policy can remain up without an active connection to a PCE. The range is from 0 to 3600 seconds; the default value is 60.

```
Router(config-sr-te-pcc)# timers delegation-timeout seconds
```

**PCE-Initiated SR Policy Timers**

Use the `timers initiated orphans` command to specify the amount of time that a PCE-initiated SR policy will remain delegated to a PCE peer that is no longer reachable by the PCC. The range is from 10 to 180 seconds; the default value is 180.

```
Router(config-sr-te-pcc)# timers initiated orphans seconds
```

Use the `timers initiated state` command to specify the amount of time that a PCE-initiated SR policy will remain programmed while not being delegated to any PCE. The range is from 15 to 14440 seconds (24 hours); the default value is 600.

```
Router(config-sr-te-pcc)# timers initiated state seconds
```

To better understand how the PCE-initiated SR policy timers operate, consider the following example:

- PCE A instantiates SR policy P at head-end N.
- Head-end N delegates SR policy P to PCE A and programs it in forwarding.
- If head-end N detects that PCE A is no longer reachable, then head-end N starts the PCE-initiated `orphan` and `state` timers for SR policy P.
- If PCE A reconnects before the `orphan` timer expires, then SR policy P is automatically delegated back to its original PCE (PCE A).
- After the `orphan` timer expires, SR policy P will be eligible for delegation to any other surviving PCE(s).
- If SR policy P is not delegated to another PCE before the `state` timer expires, then head-end N will remove SR policy P from its forwarding.

**Enable SR-TE SYSLOG Alarms**

Use the `logging policy status` command to enable SR-TE related SYSLOG alarms.

```
Router(config-sr-te)# logging policy status
```

**Enable PCEP Reports to SR-PCE**

Use the `report-all` command to enable the PCC to report all SR policies in its database to the PCE.

```
Router(config-sr-te-pcc)# report-all
```
**Customize MSD Value at PCC**

Use the `maximum-sid-depth value` command to customize the Maximum SID Depth (MSD) signaled by PCC during PCEP session establishment.

The default MSD value is equal to the maximum MSD supported by the platform (10).

```
Router(config-sr-te)# maximum-sid-depth value
```

For cases with path computation at PCE, a PCC can signal its MSD to the PCE in the following ways:

- **During PCEP session establishment** – The signaled MSD is treated as a node-wide property.
  - MSD is configured under `segment-routing traffic-eng maximum-sid-depth value` command

- **During PCEP LSP path request** – The signaled MSD is treated as an LSP property.
  - On-demand (ODN) SR Policy: MSD is configured using the `segment-routing traffic-eng on-demand color color maximum-sid-depth value` command
  - Local SR Policy: MSD is configured using the `segment-routing traffic-eng policy WORD candidate-paths preference preference dynamic metric sid-limit value` command.

**Note**

If the configured MSD values are different, the per-LSP MSD takes precedence over the per-node MSD.

After path computation, the resulting label stack size is verified against the MSD requirement.

- If the label stack size is larger than the MSD and path computation is performed by PCE, then the PCE returns a "no path" response to the PCC.
- If the label stack size is larger than the MSD and path computation is performed by PCC, then the PCC will not install the path.

**Note**

A sub-optimal path (if one exists) that satisfies the MSD constraint could be computed in the following cases:

- For a dynamic path with TE metric, when the PCE is configured with the `pce segment-routing te-latency` command or the PCC is configured with the `segment-routing traffic-eng te-latency` command.
- For a dynamic path with LATENCY metric
- For a dynamic path with affinity constraints

For example, if the PCC MSD is 4 and the optimal path (with an accumulated metric of 100) requires 5 labels, but a sub-optimal path exists (with accumulated metric of 110) requiring 4 labels, then the sub-optimal path is installed.

**Customize the SR-TE Path Calculation**

Use the `te-latency` command to enable ECMP-aware path computation for TE metric.

```
Router(config-sr-te)# te-latency
```
Configure PCEP Redundancy Type

Use the `redundancy pcc-centric` command to enable PCC-centric high-availability model, where the PCC allows only the PCE with the lowest precedence to initiate policies.

Router(config-sr-te-pcc)# redundancy pcc-centric

Configuring Head-End Router as PCEP PCC and Customizing SR-TE Related Options: Example

The following example shows how to configure an SR-TE head-end router with the following functionality:

- Enable the SR-TE head-end router as a PCEP client (PCC) with 3 PCEP servers (PCE) with different precedence values. The PCE with IP address 1.1.1.57 is selected as BEST.
- Enable SR-TE related syslogs.
- Set the Maximum SID Depth (MSD) signaled during PCEP session establishment to 5.
- Enable PCEP reporting for all policies in the node.

```
segment-routing
traffic-eng
pcc
  source-address ipv4 1.1.1.2
  pce address ipv4 1.1.1.57
    precedence 150
    password clear <password>
  !
  pce address ipv4 1.1.1.58
    precedence 200
    password clear <password>
  !
  pce address ipv4 1.1.1.59
    precedence 250
    password clear <password>
  !
  !
logging
  policy status
!
  maximum-sid-depth 5
pcc
  report-all
!
!
end
```

Verification

RP/0/RSP0/CPU0:Router# show segment-routing traffic-eng pcc ipv4 peer

PCC's peer database:
-------------------
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.

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:

**SUMMARY STEPS**

1. configure
2. router bgp as-number
3. bgp router-id ip-address
4. address-family {ipv4 | ipv6} sr-policy
5. exit
6. neighbor ip-address
7. remote-as as-number
8. address-family {ipv4 | ipv6} sr-policy
9. route-policy route-policy-name {in | out}
# DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>router bgp as-number</code></td>
<td>Specifies the BGP AS number and enters the BGP configuration mode, allowing you to configure the BGP routing process.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config)# router bgp 65000</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>bgp router-id ip-address</code></td>
<td>Configures the local router with a specified router ID.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-bgp)# bgp router-id 1.1.1.1</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>`address-family {ipv4</td>
<td>ipv6} sr-policy`</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-bgp)# address-family ipv4 sr-policy</code></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>exit</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td><code>neighbor ip-address</code></td>
<td>Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-bgp)# neighbor 10.10.0.1</code></td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td><code>remote-as as-number</code></td>
<td>Creates a neighbor and assigns a remote autonomous system number to it.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-bgp-nbr)# remote-as 1</code></td>
<td></td>
</tr>
<tr>
<td>Step 8</td>
<td>`address-family {ipv4</td>
<td>ipv6} sr-policy`</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-bgp-nbr)# address-family ipv4 sr-policy</code></td>
<td></td>
</tr>
<tr>
<td>Step 9</td>
<td>`route-policy route-policy-name {in</td>
<td>out}`</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-bgp-nbr-af)#</code></td>
<td></td>
</tr>
</tbody>
</table>
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.

```conf
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.

```conf
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) refers to the functionality where BGP service traffic is automatically steered on the right SLA path programmed by an SR policy. The decision to steer traffic into an SR policy is based on intent (color) and next-hop of the service route, regardless of the instantiation method of a policy (pushed by BGP-TE, provisioned manually, automatically instantiate on-demand [SR-ODN], or pushed by PCEP). AS provides
per-destination steering. A matching SR policy can already be present at the head-end router or can be instantiated on-demand (SR-ODN) when receiving the service route update.

See the Verifying BGP VRF Information, on page 86 and Verifying Forwarding (CEF) Table, on page 87 sections for sample output that shows AS implementation.

Color-Only Automated Steering

Color-only steering is a traffic steering mechanism where a policy is created with given color, regardless of the endpoint.

You can create an SR-TE policy for a specific color that uses a NULL end-point (0.0.0.0 for IPv4 NULL, and ::0 for IPv6 NULL end-point). This means that you can have a single policy that can steer traffic that is based on that color and a NULL endpoint for routes with a particular color extended community, but different destinations (next-hop).

Every SR-TE policy with a NULL end-point must have an explicit path-option. The policy cannot have a dynamic path-option (where the path is computed by the head-end or PCE) since there is no destination for the policy.

You can also specify a color-only (CO) flag in the color extended community for overlay routes. The CO flag allows the selection of an SR-policy with a matching color, regardless of endpoint Sub-address Family Identifier (SAFI) (IPv4 or IPv6). See Setting CO Flag, on page 135.

Configure Color-Only Steering

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# policy P1
Router(config-sr-te-policy)# color 1 end-point ipv4 0.0.0.0

Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# policy P2
Router(config-sr-te-policy)# color 2 end-point ipv6 ::0
```

```
Router# show running-configuration
segment-routing
  traffic-eng
    policy P1
      color 1 end-point ipv4 0.0.0.0
    !
    policy P2
      color 2 end-point ipv6 ::0
    !
  end
```
Setting CO Flag

The BGP-based steering mechanism matches BGP color and next-hop with that of an SR-TE policy. If the policy does not exist, BGP requests SR-PCE to create an SR-TE policy with the associated color, end-point, and explicit paths. For color-only steering (NULL end-point), you can configure a color-only (CO) flag as part of the color extended community in BGP.

Note

See Color-Only Automated Steering, on page 134 for information about color-only steering (NULL end-point).

The behavior of the steering mechanism is based on the following values of the CO flags:

| co-flag 00 | 1. The BGP next-hop and color <N, C> is matched with an SR-TE policy of same <N, C>.  
|            | 2. If a policy does not exist, then IGP path for the next-hop N is chosen. |
| co-flag 01 | 1. The BGP next-hop and color <N, C> is matched with an SR-TE policy of same <N, C>.  
|            | 2. If a policy does not exist, then an SR-TE policy with NULL end-point with the same address-family as N and color C is chosen.  
|            | 3. If a policy with NULL end-point with same address-family as N does not exist, then an SR-TE policy with any NULL end-point and color C is chosen.  
|            | 4. If no match is found, then IGP path for the next-hop N is chosen. |

Configuration Example

Router(config)# extcommunity-set opaque overlay-color  
Router(config-ext)# co-flag 01  
Router(config-ext)# end-set  
Router(config)# route-policy color  
Router(config-rpl)# if destination in (5.5.5.1/32) then  
Router(config-rpl-if)# set extcommunity color overlay-color  
Router(config-rpl-if)# endif  
Router(config-rpl)# pass  
Router(config-rpl)# end-policy  
Router(config)#

Address-Family Agnostic Automated Steering

Address-family agnostic steering uses an SR-TE policy to steer both labeled and unlabeled IPv4 and IPv6 traffic. This feature requires support of IPv6 encapsulation (IPv6 caps) over IPv4 endpoint policy.

IPv6 caps for IPv4 NULL end-point is enabled automatically when the policy is created in Segment Routing Path Computation Element (SR-PCE). The binding SID (BSID) state notification for each policy contains an "ipv6_caps" flag that notifies SR-PCE clients (PCC) of the status of IPv6 caps (enabled or disabled).
An SR-TE policy with a given color and IPv4 NULL end-point could have more than one candidate path. If any of the candidate paths has IPv6 caps enabled, then all of the remaining candidate paths need IPv6 caps enabled. If IPv6 caps is not enabled on all candidate paths of same color and end-point, traffic drops can occur.

You can disable IPv6 caps for a particular color and IPv4 NULL end-point using the `ipv6 disable` command on the local policy. This command disables IPv6 caps on all candidate paths that share the same color and IPv4 NULL end-point.

**Disable IPv6 Encapsulation**

```plaintext
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# policy P1
Router(config-sr-te-policy)# color 1 end-point ipv4 0.0.0.0
Router(config-sr-te-policy)# ipv6 disable
```

**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.

---

**Note**

In Cisco IOS XR 6.3.2 and later releases, you can specify an explicit BSID for an SR-TE policy. See the following **Explicit Binding SID** section.

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 Policies 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 4: Stitching SR-TE Policies Using Binding SID*
### Table 5: Router IP Address

<table>
<thead>
<tr>
<th>Router</th>
<th>Prefix Address</th>
<th>Prefix SID/Adj-SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Loopback0 - 1.1.1.3</td>
<td>Prefix SID - 16003</td>
</tr>
<tr>
<td>4</td>
<td>Loopback0 - 1.1.1.4, Link node 4 to node 6 - 10.4.6.4</td>
<td>Prefix SID - 16004, Adjacency SID - dynamic</td>
</tr>
<tr>
<td>5</td>
<td>Loopback0 - 1.1.1.5</td>
<td>Prefix SID - 16005</td>
</tr>
<tr>
<td>6</td>
<td>Loopback0 - 1.1.1.6, Link node 4 to node 6 - 10.4.6.6</td>
<td>Prefix SID - 16006, Adjacency SID - dynamic</td>
</tr>
<tr>
<td>9</td>
<td>Loopback0 - 1.1.1.9</td>
<td>Prefix SID - 16009</td>
</tr>
<tr>
<td>10</td>
<td>Loopback0 - 1.1.1.10</td>
<td>Prefix SID - 16010</td>
</tr>
</tbody>
</table>

### 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**

```plaintext
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_BSIDs
  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_BSIDs
  
  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: srtc_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
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.

Static Route Traffic-Steering using SR-TE Policy

In previous releases, you could only associate Segment Routing Label Switched Paths (SR-LSP) with a static route. The Static Route Traffic-Steering using SRTE Policy feature allows you to specify a Segment Routing (SR) policy as an interface type when configuring static routes for MPLS and IPv6 data planes.

For information on configuring static routes, see the "Implementing Static Routes" chapter in the Routing Configuration Guide for Cisco ASR 9000 Series Routers.

Configuration Example

```bash
Router(config)# router static
Router (config-static)# address-family ipv4 unicast

//configure administrative distance
Router (config-static-afi)# 1.1.1.1/32 sr-policy policy1 110

//Configure load metric
Router (config-static-afi)# 1.1.1.1/32 sr-policy policy1 metric 5

//Install the route in RIB regardless of reachability
Router (config-static-afi)# 1.1.1.1/32 sr-policy policy1 permanent
```

Running Configuration

```bash
configure
router static
address-family ipv4 unicast
1.1.1.1/32 sr-policy policy1 110
1.1.1.1/32 sr-policy policy1 metric 5
1.1.1.1/32 sr-policy policy1 permanent
!
```

Segment Routing Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 7.0.x
Autoroute Include

You can configure SR-TE policies with Autoroute Include to steer specific IGP (IS-IS, OSPF) prefixes over non-shortest paths and to divert the traffic for those prefixes on to the SR-TE policy. Autoroute Include applies Autoroute Announce functionality to the specified destinations or prefixes.

The Autoroute SR-TE policy adds the prefixes into the IGP, which determines if the prefixes on the endpoint or downstream of the endpoint are eligible to use the SR-TE policy. If a prefix is eligible, then the IGP checks if the prefix is listed in the Autoroute Include configuration. If the prefix is included, then the IGP downloads the prefix route with the SR-TE policy as the outgoing path.

Autoroute Include supports three metric types:

- Default (no metric): The path over the SR-TE policy inherits the shortest path metric.
- Absolute metric: The shortest path metric to the policy endpoint is replaced with the configured absolute metric. The metric to any prefix that is Autoroute Included is modified to the absolute metric.
- Relative metric: The shortest path metric to the policy endpoint is modified with the relative value configured (plus or minus).

To prevent load-balancing over IGP paths, you can specify a metric that is lower than the value that IGP takes into account for autorouted destinations (for example, autoroute metric relative -1).

Configuration Example

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# policy P1
Router(config-sr-te-policy)# color 20 end ipv4 1.1.1.2
Router(config-sr-te-policy)# autoroute include ipv4 1.1.1.21/32
Router(config-sr-te-policy)# autoroute include ipv4 1.1.1.23/32
Router(config-sr-te-policy)# autoroute metric constant 1
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 100
Router(config-sr-te-pp-index)# explicit segment-list Plist-1
```

Policy-Based Tunnel Selection for SR-TE Policy

Policy-Based Tunnel Selection (PBTS) is a mechanism that lets you direct traffic into specific SR-TE policies based on different classification criteria. PBTS benefits Internet service providers (ISPs) that carry voice and data traffic through their networks, who want to route this traffic to provide optimized voice service.

PBTS works by selecting SR-TE policies based on the classification criteria of the incoming packets, which are based on the IP precedence, experimental (EXP), differentiated services code point (DSCP), or type of service (ToS) field in the packet. Default-class configured for paths is always zero (0). If there is no TE for
a given forward-class, then the default-class (0) will be tried. If there is no default-class, then the packet is dropped. PBTS supports up to seven (exp 1 - 7) EXP values associated with a single SR-TE policy. For more information about PBTS, refer to the Policy-Based Tunnel Selection section in the MPLS Configuration Guide for Cisco ASR 9000 Series Routers.

Configure Policy-Based Tunnel Selection for SR-TE Policies

The following section lists the steps to configure PBTS for an SR-TE policy.

Steps 1 through 4 are detailed in the Implementing MPLS Traffic Engineering chapter of the MPLS Configuration Guide for Cisco ASR 9000 Series Routers.

1. Define a class-map based on a classification criteria.
2. Define a policy-map by creating rules for the classified traffic.
3. Associate a forward-class to each type of ingress traffic.
4. Enable PBTS on the ingress interface, by applying this service-policy.
5. Create one or more egress SR-TE policies (to carry packets based on priority) to the destination and associate the egress SR-TE policy to a forward-class.

Configuration Example

Router(config)# segment-routing traffic-eng
Router(config-sr-te)# policy POLICY-PBTS
Router(config-sr-te-policy)# color 1001 end-point ipv4 1.1.1.20
Router(config-sr-te-policy)# autoroute
Router(config-sr-te-policy-autoroute)# include all
Router(config-sr-te-policy-autoroute)# forward-class 1
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 1
Router(config-sr-te-policy-path-pref)# explicit segment-list SIDLIST1

Running Configuration

segment-routing
traffic-eng
policy POLICY-PBTS
color 1001 end-point ipv4 1.1.1.20
autoroute
include all
forward-class 1
!
candidate-paths
preference 1
explicit segment-list SIDLIST1
!
!
!
Miscellaneous

LDP over Segment Routing Policy

The LDP over Segment Routing Policy feature enables an LDP-targeted adjacency over a Segment Routing (SR) policy between two routers. This feature extends the existing MPLS LDP address family neighbor configuration to specify an SR policy as the targeted end-point.

LDP over SR policy is supported for locally configured SR policies with IPv4 end-points.

For more information about MPLS LDP, see the "Implementing MPLS Label Distribution Protocol" chapter in the [MPLS Configuration Guide](#).

For more information about Autoroute, see the [Autoroute Announce for SR-TE](#) section.

**Note**

Before you configure an LDP targeted adjacency over SR policy name, you need to create the SR policy under Segment Routing configuration. The SR policy interface names are created internally based on the color and endpoint of the policy. LDP is non-operational if SR policy name is unknown.

The following functionality applies:

1. Configure the SR policy – LDP receives the associated end-point address from the interface manager (IM) and stores it in the LDP interface database (IDB) for the configured SR policy.

2. Configure the SR policy name under LDP – LDP retrieves the stored end-point address from the IDB and uses it. Use the auto-generated SR policy name assigned by the router when creating an LDP targeted adjacency over an SR policy. Auto-generated SR policy names use the following naming convention: `srte_c_color_val_ep_endpoint-address`. For example, `srte_c_1000_ep_1.1.1.2`

**Configuration Example**

```bash
/* Enter the SR-TE configuration mode and create the SR policy. This example corresponds to a local SR policy with an explicit path. */
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# segment-list sample-sid-list
Router(config-sr-te-sl)# index 10 address ipv4 1.1.1.7
Router(config-sr-te-sl)# index 20 address ipv4 1.1.1.2
Router(config-sr-te-sl)# exit
Router(config-sr-te)# policy sample_policy
Router(config-sr-te-policy)# color 1000 end-point ipv4 1.1.1.2
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 sample-sid-list
Router(config-sr-te-pp-info)# end

/* Configure LDP over an SR policy */
Router(config)# mpls ldp
Router(config-ldp)# address-family ipv4
Router(config-ldp-af)# neighbor sr-policy srte_c_1000_ep_1.1.1.2 targeted
```
Do one of the following to configure LDP discovery for targeted hellos:

- Active targeted hellos (SR policy head end):

  ```
  mpls ldp
  interface GigabitEthernet0/0/0/0
  !
  !
  ```

- Passive targeted hellos (SR policy end-point):

  ```
  mpls ldp
  address-family ipv4
  discovery targeted-hello accept
  !
  !
  ```

### Running Configuration

```
segment-routing
traffic-eng
  segment-list sample-sid-list
  index 10 address ipv4 1.1.1.7
  index 20 address ipv4 1.1.1.2
  !
  policy sample_policy
  color 1000 end-point ipv4 1.1.1.2
  candidate-paths
  preference 100
  explicit segment-list sample-sid-list
  !
  !
  !
mpls ldp
address-family ipv4
neighbor sr-policy srte_c_1000_ep_1.1.1.2 targeted
discovery targeted-hello accept
!
!
```

### Verification

```
Router# show mpls ldp interface brief
Interface | VRF Name | Config | Enabled | IGP-Auto-Cfg | TE-Mesh-Grp | cfg
---------- | -------- | ------ | ------- | ------------ | ----------- |
Te0/3/0/0/3 | default | Y     | Y       | 0 | N/A |
Te0/3/0/0/6 | default | Y     | Y       | 0 | N/A |
Te0/3/0/0/7 | default | Y     | Y       | 0 | N/A |
Te0/3/0/0/8 | default | N     | N       | 0 | N/A |
Te0/3/0/0/9 | default | N     | N       | 0 | N/A |
srte_c_1000  | default | Y     | Y       | 0 | N/A |
```
Router# show mpls ldp interface
Interface TenGigE0/3/0/0/3 (0xa000340)
  VRF: 'default' (0x60000000)
  Enabled via config: LDP interface
Interface TenGigE0/3/0/0/6 (0xa000400)
  VRF: 'default' (0x60000000)
  Enabled via config: LDP interface
Interface TenGigE0/3/0/0/7 (0xa000440)
  VRF: 'default' (0x60000000)
  Enabled via config: LDP interface
Interface TenGigE0/3/0/0/8 (0xa000480)
  VRF: 'default' (0x60000000)
  Disabled:
Interface TenGigE0/3/0/0/9 (0xa0004c0)
  VRF: 'default' (0x60000000)
  Disabled:
Interface srte_c_1000_ep_1.1.1.2 (0x520)
  VRF: 'default' (0x60000000)
  Enabled via config: LDP interface

Router# show segment-routing traffic-eng policy color 1000

SR-TE policy database
---------------------
Color: 1000, End-point: 1.1.1.2
Name: srte_c_1000_ep_1.1.1.2
Status: Admin: up Operational: up for 00:02:00 (since Jul 2 22:39:06.663)
  Candidate-paths: Preference: 100 (configuration) (active)
                  Name: sample_policy
                  Requested BSID: dynamic
                  PCC info:
                  Symbolic name: cfg_sample_policy_discr_100
                  PLSP-ID: 17
                  Explicit: segment-list sample-sid-list (valid)
                  Weight: 1, Metric Type: TE
                  16007 [Prefix-SID, 1.1.1.7]
                  16002 [Prefix-SID, 1.1.1.2]
Attributes:
  Binding SID: 80011
  Forward Class: 0
  Steering BGP disabled: no
  IPv6 caps enable: yes

Router# show mpls ldp neighbor 1.1.1.2 detail
Peer LDP Identifier: 1.1.1.2:0
  TCP connection: 1.1.1.2:646 - 1.1.1.6:57473
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 421/423; Downstream-Unsolicited:
  Up time: 05:22:02
  LDP Discovery Sources:
    IPv4: (1)
      Targeted Hello (1.1.1.6 -> 1.1.1.2, active/passive)
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (9)
    1.1.1.2  2.2.2.99  10.1.2.2  10.2.3.2
    10.2.4.2 10.2.22.2 10.2.222.2 10.30.110.132

Segment Routing Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 7.0.x
Configure Seamless Bidirectional Forwarding Detection

Bidirectional forwarding detection (BFD) provides low-overhead, short-duration detection of failures in the path between adjacent forwarding engines. BFD allows a single mechanism to be used for failure detection over any media and at any protocol layer, with a wide range of detection times and overhead. The fast detection of failures provides immediate reaction to failure in the event of a failed link or neighbor.

In BFD, each end of the connection maintains a BFD state and transmits packets periodically over a forwarding path. Seamless BFD (SBFD) is unidirectional, resulting in faster session activation than BFD. The BFD state and client context is maintained on the head-end (initiator) only. The tail-end (reflector) validates the BFD packet and responds, so there is no need to maintain the BFD state on the tail-end.

**Initiators and Reflectors**

SBFD runs in an asymmetric behavior, using initiators and reflectors.

The following figure represents the roles of the SBFD initiator and reflector.
The initiator is an SBFD session on a network node that performs a continuity test to a remote entity by sending SBFD packets. The initiator injects the SBFD packets into the segment-routing traffic-engineering (SRTE) policy. The initiator triggers the SBFD session and maintains the BFD state and client context.

The reflector is an SBFD session on a network node that listens for incoming SBFD control packets to local entities and generates response SBFD control packets. The reflector is stateless and only reflects the SBFD packets back to the initiator.

A node can be both an initiator and a reflector, if you want to configure different SBFD sessions.

For SR-TE, SBFD control packets are label switched in forward and reverse direction. For SBFD, the tail-end node is the reflector node; other nodes cannot be a reflector. When using SBFD with SR-TE, if the forward and return directions are label-switched paths, SBFD need not be configured on the reflector node.

**Discriminators**

The BFD control packet carries 32-bit discriminators (local and remote) to demultiplex BFD sessions. SBFD requires globally unique SBFD discriminators that are known by the initiator.

The SBFD control packets contain the discriminator of the initiator, which is created dynamically, and the discriminator of the reflector, which is configured as a local discriminator on the reflector.

**Configure the SBFD Reflector**

To ensure the SBFD packet arrives on the intended reflector, each reflector has at least one globally unique discriminator. Globally unique discriminators of the reflector are known by the initiator before the session starts. An SBFD reflector only accepts BFD control packets where "Your Discriminator" is the reflector discriminator.

This task explains how to configure local discriminators on the reflector.
Before you begin

Enable mpls oam on the reflector to install a routing information base (RIB) entry for 127.0.0.0/8.

```bash
Router_5# configure
Router_5(config)# mpls oam
Router_5(config-oam)#
```

**SUMMARY STEPS**

1. configure
2. sbfd
3. local-discriminator \( \{ \text{ipv4-address} | \text{32-bit-value} | \text{dynamic} | \text{interface interface} \} \)
4. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbfd</td>
<td>Enters SBFD configuration mode.</td>
</tr>
<tr>
<td>Example: <code>Router_5(config)# sbfd</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> local-discriminator ( { \text{ipv4-address}</td>
<td>\text{32-bit-value}</td>
</tr>
</tbody>
</table>
| Example: `Router_5(config-sbfd)# local-discriminator 1.1.1.5`
`Router_5(config-sbfd)# local-discriminator 987654321`
`Router_5(config-sbfd)# local-discriminator dynamic`
`Router_5(config-sbfd)# local-discriminator interface Loopback0` | |
| **Step 4** commit | Verify the local discriminator configuration. |

**Example**

```bash
Router_5# show bfd target-identifier local
```
## Configure the SBFD Initiator

Configure the SBFD Initiator.

### Enable Line Cards to Host BFD Sessions

The SBFD initiator sessions are hosted by the line card CPU.

This task explains how to enable line cards to host BFD sessions.

## SUMMARY STEPS

1. configure
2. bfd
3. multipath include location node-id

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong> bfd</td>
<td>Enters BFD configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router_1(config)# bfd</td>
</tr>
<tr>
<td><strong>Step 3</strong> multipath include location node-id</td>
<td>Configures BFD multiple path on specific line card. Any of the configured line cards can be instructed to host a BFD session.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router_1(config-bfd)# multipath include location</td>
</tr>
</tbody>
</table>
### Purpose

**Command or Action**

```
0/1/CPU0
Router_1(config-bfd)# multipath include location
0/2/CPU0
Router_1(config-bfd)# multipath include location
0/3/CPU0
```

### Purpose

**What to do next**

Map a destination address to a remote discriminator.

**Map a Destination Address to a Remote Discriminator**

The SBFD initiator uses a Remote Target Identifier (RTI) table to map a destination address (Target ID) to a remote discriminator.

This task explains how to map a destination address to a remote discriminator.

### SUMMARY STEPS

1. **configure**
2. **sbfd**
3. **remote-target ipv4 ipv4-address**
4. **remote-discriminator remote-discriminator**

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>configure</strong></td>
<td><strong>Example:</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>sbfd</strong></td>
<td>Enters SBFD configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router_1(config)# sbfd</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>remote-target ipv4 ipv4-address</strong></td>
<td>Configures the remote target.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router_1(config-sbfd)# remote-target ipv4 1.1.1.5</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>remote-discriminator remote-discriminator</strong></td>
<td>Maps the destination address (Target ID) to a remote discriminator.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router_1(config-sbfd-nnnn)# remote-discriminator</code></td>
<td></td>
</tr>
</tbody>
</table>
Verify the remote discriminator configuration.

**Example**

```
Router_1# show bfd target-identifier remote
Remote Target Identifier Table
-----------------------------
Discr Discr Src VRF TID Type Status
  ------ --------- ------- -------- -----
     Target ID Name
  ------ -------        ------ ------
16843013 Remote default ipv4 enable
     1.1.1.5
```

Legend: TID = Target Identifier

**What to do next**

Enable SBFD on an SR-TE policy.

**Enable Seamless BFD Under an SR-TE Policy or SR-ODN Color Template**

This example shows how to enable SBFD on an SR-TE policy or an SR on-demand (SR-ODN) color template.

**Note**

Do not use BFD with disjoint paths. The reverse path might not be disjoint, causing a single link failure to bring down BFD sessions on both the disjoint paths.

**Enable BFD**

- Use the `bfd` command in SR-TE policy configuration mode to enable BFD and enters BFD configuration mode.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# policy POLICY1
Router(config-sr-te-policy)# bfd
Router(config-sr-te-policy-bfd)#
```

Use the `bfd` command in SR-ODN configuration mode to enable BFD and enters BFD configuration mode.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# on-demand color 10
Router(config-sr-te-color)# bfd
Router(config-sr-te-color-bfd)#
```
Configure BFD Options

- Use the `minimum-interval milliseconds` command to set the interval between sending BFD hello packets to the neighbor. The range is from 15 to 200. The default is 15.
  
  Router(config-sr-te-policy-bfd)# minimum-interval 50

- Use the `multiplier multiplier` command to set the number of times a packet is missed before BFD declares the neighbor down. The range is from 2 to 10. The default is 3.
  
  Router(config-sr-te-policy-bfd)# multiplier 2

- Use the `invalidation-action {down | none}` command to set the action to be taken when BFD session is invalidated.
  
  - `down`: LSP can only be operationally up if the BFD session is up
  - `none`: BFD session state does not affect LSP state, use for diagnostic purposes

  Router(config-sr-te-policy-bfd)# invalidation-action down

- (SR-TE policy only) Use the `reverse-path binding-label label` command to specify BFD packets return to head-end by using a binding label.

  By default, the S-BFD return path (from tail-end to head-end) is via IPv4. You can use a reverse binding label so that the packet arrives at the tail-end with the reverse binding label as the top label. This label is meant to point to a policy that will take the BFD packets back to the head-end. The reverse binding label is configured per-policy.

  Note that when MPLS return path is used, BFD uses echo mode packets, which means the tail-end’s BFD reflector does not process BFD packets at all.

  The MPLS label value at the tail-end and the head-end must be synchronized by the operator or controller. Because the tail-end binding label should remain constant, configure it as an explicit BSID, rather than dynamically allocated.

  Router(config-sr-te-policy-bfd)# reverse-path binding-label 24036

- Use the `logging session-state-change` command to log when the state of the session changes

  Router(config-sr-te-policy-bfd)# logging session-state-change

Examples

This example shows how to enable SBFD on an SR-TE policy.

Router(config)# segment-routing traffic-eng
Router(config-sr-te)# policy POLICY1
Router(config-sr-te-policy)# bfd
Router(config-sr-te-policy-bfd)# invalidation-action down
Router(config-sr-te-policy-bfd)# minimum-interval 50
Router(config-sr-te-policy-bfd)# multiplier 2
Router(config-sr-te-policy-bfd)# reverse-path binding-label 24036
Router(config-sr-te-policy-bfd)# logging session-state-change
traffic-eng
policy POLICY1
  bfd
    minimum-interval 50
    multiplier 2
    invalidation-action down
    reverse-path
    binding-label 24036
  !
  logging
  session-state-change
  !
  !

This example shows how to enable SBFD on an SR-ODN color.

Router(config)# segment-routing traffic-eng
Router(config-sr-te)# on-demand color 10
Router(config-sr-te-color)# bfd
Router(config-sr-te-color-bfd)# minimum-interval 50
Router(config-sr-te-color-bfd)# multiplier 2
Router(config-sr-te-color-bfd)# logging session-state-change
Router(config-sr-te-color-bfd)# invalidation-action down

SR-TE Head-End IPv4 Unnumbered Interface Support

This feature allows IPv4 unnumbered interfaces to be part of an SR-TE head-end router topology database.

An unnumbered IPv4 interface is not identified by its own unique IPv4 address. Instead, it is identified by the router ID of the node where this interface resides and the local SNMP index assigned for this interface.

This feature provides enhancements to the following components:

- IGPs (IS-IS and OSPF):
  - Support the IPv4 unnumbered interfaces in the SR-TE context by flooding the necessary interface information in the topology

- SR-PCE:
Note

SR-PCE and path computation clients (PCCs) need to be running Cisco IOS XR 7.0.2 or later.

- Compute and return paths from a topology containing IPv4 unnumbered interfaces.
- Process reported SR policies from a head-end router that contain hops with IPv4 unnumbered adjacencies.

PCEP extensions for IPv4 unnumbered interfaces adhere to IETF RFC8664 “PCEP Extensions for Segment Routing” (https://datatracker.ietf.org/doc/rfc8664/). The unnumbered hops use a Node or Adjacency Identifier (NAI) of type 5. This indicates that the segment in the explicit routing object (ERO) is an unnumbered adjacency with an IPv4 ID and an interface index.

- SR-TE process at the head-end router:
  - Compute its own local path over a topology, including unnumbered interfaces.
  - Process PCE-computed paths that contain hops with IPv4 unnumbered interfaces.
  - Report a path that contains hops with IPv4 unnumbered interfaces to the PCE.

Configuration Example

The following example shows how to configure an IPv4 unnumbered interface:

```plaintext
RP/0/0/CPU0:rtrA(config)# interface GigabitEthernet0/0/0/0
RP/0/0/CPU0:rtrA(config-if)# ipv4 point-to-point
RP/0/0/CPU0:rtrA(config-if)# ipv4 unnumbered Loopback0
```

To bring up the IPv4 unnumbered adjacency under the IGP, configure the link as point-to-point under the IGP configuration. The following example shows how to configure the link as point-to-point under the IGP configuration:

```plaintext
RP/0/0/CPU0:rtrA(config)# router ospf one
RP/0/0/CPU0:rtrA(config-ospf)# area 0
RP/0/0/CPU0:rtrA(config-ospf-ar)# interface GigabitEthernet0/0/0/0
RP/0/0/CPU0:rtrA(config-ospf-ar-if)# network point-to-point
```

Verification

Use the `show ipv4 interface` command to display information about the interface:

```plaintext
RP/0/0/CPU0:rtrA# show ipv4 interface GigabitEthernet0/0/0/0 brief
Tue Apr  2 12:59:53.140 EDT
Interface            IP-Address     Status  Protocol
GigabitEthernet0/0/0/0             192.168.0.1 Up       Up
```

This interface shows the IPv4 address of Loopback0.

Use the `show snmp interface` command to find the SNMP index for this interface:

```plaintext
RP/0/0/CPU0:rtrA# show snmp interface
Tue Apr  2 13:02:49.190 EDT
ifName : Null10 ifIndex : 3
ifName : Loopback0 ifIndex : 10
```
The interface is identified with the pair (IPv4:192.168.0.1, index:6).

Use the `show ospf neighbor` command to display the adjacency:

```
RP/0/0/CPU0:rtrA# show ospf neighbor gigabitEthernet 0/0/0/0 detail
```

```
Neighbor 192.168.0.4, interface address 192.168.0.4
   In the area 0 via interface GigabitEthernet0/0/0/0
   Neighbor priority is 1, State is FULL, 6 state changes
   Adjacency SIDs:
      Label: 24001, Dynamic, Unprotected
      Neighbor Interface ID: 4
```

The output of the `show segment-routing traffic-eng ipv4 topology` command is enhanced to display the interface index instead of the IP address for unnumbered interfaces:

```
RP/0/0/CPU0:rtrA# show segment-routing traffic-eng ipv4 topology
```

```
Link[2]: Unnumbered local index 6, remote index 4
Local node:
   OSPF router ID: 192.168.0.1 area ID: 0 ASN: 0
Remote node:
   TE router ID: 192.168.0.4
OSPF router ID: 192.168.0.4 area ID: 0 ASN: 0
Metric: IGP 1, TE 1, Latency 1 microseconds
Bandwidth: Total 125000000 Bps, Reservable 0 Bps
Admin-groups: 0x00000000
Adj SID: 24001 (unprotected)
```

The output of the `show segment-routing traffic-eng policy detail` command includes unnumbered hops:

```
RP/0/0/CPU0:rtrA# show segment-routing traffic-eng policy detail
```

```
Dynamic (pce 192.168.0.5) (valid)
   Metric Type: TE, Path Accumulated Metric: 3
   24001 [Adjacency-SID, unnumbered 192.168.0.1(6) - 192.168.0.4(4)]
   24002 [Adjacency-SID, unnumbered 192.168.0.4(7) - 192.168.0.3(7)]
   24000 [Adjacency-SID, unnumbered 192.168.0.3(5) - 192.168.0.2(5)]
```
Segment Routing Tree Segment Identifier

Tree Segment Identifier (Tree-SID) is a tree-building solution that uses a Segment Routing Path Computation Element (SR-PCE) using path computation element protocol (PCEP) to calculate the point-to-multipoint (P2MP) tree using SR policies. Tree-SID uses a single MPLS label for building a multicast replication tree in an SR network. Tree-SID does not require multicast control protocols such as RSVP, mLDP, and PIM.

A P2MP SR policy provides an SR-based TE solution for transporting multicast traffic. It works on existing data-plane (MPLS and IP) and supports TE capabilities and single/multi routing domains. At each node of the tree, the forwarding state is represented by the same segment (using a global Tree-SID specified from the SRLB range of labels). P2MP SR policy prevents transient loop and packet loss when updating the path of a P2MP SR policy.

A P2MP SR policy request contains the following:

- Policy name
- SID for the P2MP Tree (Tree-SID)
- Address of the root-node
- Addresses of the leaf-nodes
- TE optimization criteria (for example, TE or IGP metric) and constraints

- Configure Segment Routing Tree-SID, on page 157
- Running Config, on page 159

Configure Segment Routing Tree-SID

To configure Segment Routing Tree-SID for Point-to-Multipoint (P2MP) SR policies, complete the following configurations:

1. Configure Path Computation Element Protocol (PCEP) Path Computation Client (PCC) on all nodes involved in the Tree-SID path (root, mid-point, leaf)
2. Configure Affinity Maps on the SR-PCE
3. Configure P2MP SR Policy on SR-PCE
4. Configure Multicast on the Root and Leaf Nodes
Configure PCEP PCC on All Nodes in Tree-SID Path

Configure all nodes involved in the Tree-SID path (root, mid-point, leaf) as PCEP PCC. For detailed PCEP PCC configuration information, see the Configure the Head-End Router as PCEP PCC, on page 127 section.

Configure Affinity Maps on the SR-PCE

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

Router# configure
Router(config)# pce
Router(config-pce)# segment-routing traffic-eng
Router(config-pce-sr-te)# affinity bit-map RED 23
Router(config-pce-sr-te)# affinity bit-map BLUE 24
Router(config-pce-sr-te)# affinity bit-map CROSS 25

Configure P2MP SR Policy on SR-PCE

Configure the end-point name and addresses, Tree-SID label, and constraints for the P2MP policy.

Use the endpoint-set NAME command in SR-PCE P2MP sub-mode to enter the name of the end-point set and to define the set of end-point addresses.

Router(config-pce-sr-te)# p2mp
Router(config-pce-sr-te-p2mp)# endpoint-set BAR
Router(config-pce-p2mp-ep-set)# ipv4 1.1.1.2
Router(config-pce-p2mp-ep-set)# ipv4 1.1.1.3
Router(config-pce-p2mp-ep-set)# ipv4 1.1.1.4
Router(config-pce-p2mp-ep-set)# exit
Router(config-pce-sr-te-p2mp)#

Use the policy policy command to configure the P2MP policy name and enter P2MP Policy sub-mode. Configure the source address, endpoint-set color, Tree-SID label, affinity constraints, and metric type.

Router(config-pce-sr-te-p2mp)# policy FOO
Router(config-pce-p2mp-policy)# source ipv4 1.1.1.6
Router(config-pce-p2mp-policy)# color 10 endpoint-set BAR
Router(config-pce-p2mp-policy)# treesid mpls 15200
Router(config-pce-p2mp-policy)# candidate-paths
Router(config-pce-p2mp-policy-path)# constraints
Router(config-pce-p2mp-path-const)# affinity
Router(config-pce-p2mp-path-affinity)# exclude BLUE
Router(config-pce-p2mp-path-affinity)# exit
Router(config-pce-p2mp-path-const)# exit
Router(config-pce-p2mp-policy-path)# preference 100
Router(config-pce-p2mp-policy-path-preference)# dynamic
Router(config-pce-p2mp-path-info)# metric type te
Router(config-pce-p2mp-path-info)# root
Router(config)#

Configure Multicast on the Root and Leaf Nodes

On the root node of the SR P2MP segment, use the router pim command to enter Protocol Independent Multicast (PIM) configuration mode to statically steer multicast flows into an SR P2MP policy.
Enter this configuration only on an SR P2MP segment. Multicast traffic cannot be steered into a P2P policy.

```
Router(config)# router pim
Router(config-pim)# vrf name
Router(config-pim-name)# address-family ipv4
Router(config-pim-name-ipv4)# sr-p2mp-policy FOO
Router(config-pim-name-ipv4-srp2mp)# static-group 235.1.1.5 1.1.1.6
Router(config-pim-name-ipv4-srp2mp)# root
Router(config)#
```

On the root and leaf nodes of the SR P2MP tree, use the **mdt static segment-routing** command to configure the multicast distribution tree (MDT) core as Tree-SID from the multicast VRF configuration submode.

```
Router(config)# multicast-routing
Router(config-mcast)# vrf TEST
Router(config-mcast-TEST)# address-family ipv4
Router(config-mcast-TEST-ipv4)# mdt static segment-routing
```

On the leaf nodes of an SR P2MP segment, use the **static sr-policy p2mp-policy** command to configure the static SR P2MP Policy from the multicast VRF configuration submode to statically decapsulate multicast flows.

```
Router(config)# multicast-routing
Router(config-mcast)# vrf TEST
Router(config-mcast-TEST)# address-family ipv4
Router(config-mcast-TEST-ipv4)# static sr-policy FOO
```

**Running Config**

The following example shows how to configure the end point addresses and P2MP SR policy with affinity constraints on SR-PCE.

```
pce
  segment-routing
  traffic-eng
    affinity bit-map
      RED 23
      BLUE 24
      CROSS 25
  !
p2mp
  endpoint-set BAR
    ipv4 1.1.1.2
    ipv4 1.1.1.3
    ipv4 1.1.1.4
  !
policy FOO
  source ipv4 1.1.1.6
  color 10 endpoint-set BAR
treesid mpls 15200
candidate-paths
  preference 100
dynamic
  metric
  type te
```
The following example shows how to statically decapsulate multicast flows on the leaf nodes.

```
multicast-routing
vrf TEST
    address-family ipv4
        static sr-policy FOO
```

The following example shows to configure the multicast distribution tree (MDT) core as Tree-SID on the root and leaf nodes.

```
multicast-routing
vrf TEST
    address-family ipv4
        mdt static segment-routing
```

The following example shows how to steer traffic to the SR P2MP policy on the root node.

```
router pim
vrf TEST
    address-family ipv4
        sr-p2mp-policy FOO
            static-group 232.1.1.5 1.1.1.6
```

Segment Routing Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 7.0.x
Enabling Segment Routing Flexible Algorithm

Segment Routing Flexible Algorithm allows operators to customize IGP shortest path computation according to their own needs. An operator can assign custom SR prefix-SIDs to realize forwarding beyond link-cost-based SPF. As a result, Flexible Algorithm provides a traffic engineered path automatically computed by the IGP to any destination reachable by the IGP.

The SR architecture associates prefix-SIDs to an algorithm which defines how the path is computed. Flexible Algorithm allows for user-defined algorithms where the IGP computes paths based on a user-defined combination of metric type and constraint.

This document describes the IS-IS and OSPF extensions to support Segment Routing Flexible Algorithm on an MPLS data-plane.

- Prerequisites for Flexible Algorithm, on page 161
- Building Blocks of Segment Routing Flexible Algorithm, on page 161
- Configuring Flexible Algorithm, on page 163
- Example: Configuring IS-IS Flexible Algorithm, on page 165
- Example: Configuring OSPF Flexible Algorithm, on page 165
- Example: Traffic Steering to Flexible Algorithm Paths, on page 166

Prerequisites for Flexible Algorithm

Segment routing must be enabled on the router before the Flexible Algorithm functionality is activated.

Building Blocks of Segment Routing Flexible Algorithm

This section describes the building blocks that are required to support the SR Flexible Algorithm functionality in IS-IS and OSPF.

Flexible Algorithm Definition

Many possible constrains may be used to compute a path over a network. Some networks are deployed with multiple planes. A simple form of constrain may be to use a particular plane. A more sophisticated form of constrain can include some extended metric, like delay, as described in [RFC7810]. Even more advanced case could be to restrict the path and avoid links with certain affinities. Combinations of these are also possible. To provide a maximum flexibility, the mapping between the algorithm value and its meaning can be defined
by the user. When all the routers in the domain have the common understanding what the particular algorithm value represents, the computation for such algorithm is consistent and the traffic is not subject to looping. Here, since the meaning of the algorithm is not defined by any standard, but is defined by the user, it is called as Flexible Algorithm.

Flexible Algorithm Support Advertisement

An algorithm defines how the best path is computed by IGP. Routers advertise the support for the algorithm as a node capability. Prefix-SIDs are also advertised with an algorithm value and are tightly coupled with the algorithm itself.

An algorithm is a one octet value. Values from 128 to 255 are reserved for user defined values and are used for Flexible Algorithm representation.

Flexible Algorithm Definition Advertisement

To guarantee the loop free forwarding for paths computed for a particular Flexible Algorithm, all routers in the network must share the same definition of the Flexible Algorithm. This is achieved by dedicated router(s) advertising the definition of each Flexible Algorithm. Such advertisement is associated with the priority to make sure that all routers will agree on a single and consistent definition for each Flexible Algorithm.

Definition of Flexible Algorithm includes:

- Metric type
- Affinity constraints

To enable the router to advertise the definition for the particular Flexible Algorithm, `advertise-definition` command is used. At least one router in the area, preferably two for redundancy, must advertise the Flexible Algorithm definition. Without the valid definition being advertised, the Flexible Algorithm will not be functional.

Flexible Algorithm Prefix-SID Advertisement

To be able to forward traffic on a Flexible Algorithm specific path, all routers participating in the Flexible Algorithm will install a MPLS labeled path for the Flexible Algorithm specific SID that is advertised for the prefix. Only prefixes for which the Flexible Algorithm specific Prefix-SID is advertised is subject to Flexible Algorithm specific forwarding.

Calculation of Flexible Algorithm Path

A router may compute path for multiple Flexible Algorithms. A router must be configured to support particular Flexible Algorithm before it can compute any path for such Flexible Algorithm. A router must have a valid definition of the Flexible Algorithm before such Flexible Algorithm is used.

When computing the shortest path tree for particular Flexible Algorithm:

- All nodes that do not advertise support for such Flexible Algorithm will be pruned from the topology.
- If the Flexible Algorithm definition includes affinities that are excluded, then all links for which any of such affinities are advertised will be pruned from the topology.
Router uses the metric that is part of the Flexible Algorithm definition. If the metric is not advertised for the particular link, such link will be pruned from the topology.

For IS-IS, Loop Free Alternate (LFA) paths, TI-LFA backup paths, and Microloop Avoidance paths for particular Flexible Algorithm are computed using the same constraints as the calculation of the primary paths for such Flexible Algorithm. These paths use Prefix-SIDs advertised specifically for such Flexible Algorithm in order to enforce a backup or microloop avoidance path.

LFA, TI-LFA, and Microloop Avoidance for Flexible Algorithm routes are not supported in OSPF.

**Installation of Forwarding Entries for Flexible Algorithm Paths**

Flexible Algorithm path to any prefix must be installed in the forwarding using the Prefix-SID that was advertised for such Flexible Algorithm. If the Prefix-SID for Flexible Algorithm is not known, such Flexible Algorithm path is not installed in forwarding for such prefix.

Only MPLS to MPLS entries are installed for a Flexible Algorithm path. No IP to IP or IP to MPLS entries are installed. These follow the native IPG paths computed based on the default algorithm and regular IGP metrics.

**Flexible Algorithm Prefix-SID Redistribution**

Previously, prefix redistribution from IS-IS to another IS-IS instance or protocol was limited to SR algorithm 0 (regular SPF) prefix SIDs; SR algorithm 1 (Strict SPF) and SR algorithms 128-255 (Flexible Algorithm) prefix SIDs were not redistributed along with the prefix. The Segment Routing IS-IS Flexible Algorithm Prefix SID Redistribution feature allows redistribution of strict and flexible algorithms prefix SIDs from IS-IS to another IS-IS instance or protocols. This feature is enabled automatically when you configure redistribution of IS-IS Routes with strict or Flexible Algorithm SIDs.

**Configuring Flexible Algorithm**

For information about the commands usage, see the Segment Routing Command Reference for Cisco ASR 9000 Series Routers.

The following ISIS and OSPF configuration sub-mode is used to configure Flexible Algorithm:

```
flex-algo algorithm number
```

algorithm number — value from 128 to 255

**Commands under Flexible Algorithm Configuration Mode**

The following commands are used to configure Flexible Algorithm definition under the flex-algo sub-mode:

- IS-IS
**Configuring Flexible Algorithm**

**metric-type delay**

| Note | By default the regular IGP metric is used. If delay metric is enabled, the advertised delay on the link is used as a metric for Flexible Algorithm computation. |

**OSPF**

**metric-type {delay | te-metric}**

| Note | By default the regular IGP metric is used. If delay or TE metric is enabled, the advertised delay or TE metric on the link is used as a metric for Flexible Algorithm computation. |

| • | **affinity exclude-any name1, name2, ...** |
| • | **name**—name of the affinity map |
| • | **priority priority value** |
| • | **priority value**—priority used during the Flexible Algorithm definition election. |

The following command is used to enable advertisement of the Flexible Algorithm definition in IS-IS:

```
advertise-definition
```

**Commands for Affinity Configuration**

The following command is used for defining the affinity-map. Affinity-map associates the name with the particular bit positions in the Extended Admin Group bitmask.

```
affinity-map name bit-position bit number
```

| • | **name**—name of the affinity-map. |
| • | **bit number**—bit position in the Extended Admin Group bitmask. |

The following command is used to associate the affinity with an interface:

```
affinity flex-algo name1, name2, ...
```

| • | **name**—name of the affinity-map |

**Command for Prefix-SID Configuration**

The following command is used to advertise prefix-SID for default and strict-SPF algorithm:

```
prefix-sid [strict-spf | algorithm algorithm-number] [index | absolute] sid value
```

| • | **algorithm-number**—Flexible Algorithm number |
| • | **sid value**—SID value |
Example: Configuring IS-IS Flexible Algorithm

```plaintext
router isis 1
  affinity-map red bit-position 65
  affinity-map blue bit-position 8
  affinity-map green bit-position 201

  flex-algo 128
  advertise-definition
  affinity exclude-any red
  affinity include-any blue

  flex-algo 129
  affinity exclude-any green

  address family ipv4 unicast
  segment-routing mpls

  interface Loopback0
    address-family ipv4 unicast
    prefix-sid algorithm 128 index 100
    prefix-sid algorithm 129 index 101

  interface GigabitEthernet0/0/0/0
    affinity flex-algo red

  interface GigabitEthernet0/0/0/1
    affinity flex-algo blue red

  interface GigabitEthernet0/0/0/2
    affinity flex-algo blue
```

Example: Configuring OSPF Flexible Algorithm

```plaintext
router ospf 1
  flex-algo 130
  priority 200
  affinity exclude-any
  red
  blue

  metric-type delay

  flex-algo 140
  affinity include-all
  green

  affinity include-any
  red

  interface Loopback0
    prefix-sid index 10
    prefix-sid strict-spf index 40
    prefix-sid algorithm 128 absolute 16128
```
Example: Traffic Steering to Flexible Algorithm Paths

BGP Routes on PE – Color Based Steering

SR-TE On Demand Next-Hop (ODN) feature can be used to steer the BGP traffic towards the Flexible Algorithm paths.

The following example configuration shows how to setup BGP steering local policy, assuming two router: R1 (2.2.2.2) and R2 (4.4.4.4), in the topology.

Configuration on router R1:

```
vrf Test
address-family ipv4 unicast
 import route-target
 1:150
 !
 export route-policy SET_COLOR_RED_HI_BW
 export route-target
 1:150
 !
!
interface Loopback0
ipv4 address 2.2.2.2 255.255.255.255
!
interface Loopback150
vrf Test
ipv4 address 2.2.2.222 255.255.255.255
!
interface TenGigE0/1/0/3/0
description exr1 to cxr1
ipv4 address 10.0.20.2 255.255.255.0
!
extcommunity-set opaque color129-red-igp
129
end-set
!
route-policy PASS
 pass
end-policy
!```
Enabling Segment Routing Flexible Algorithm

BGP Routes on PE – Color Based Steering

Segment Routing Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 7.0.x

167

route-policy SET_COLOR_RED_HI_BW
  set extcommunity color color129-red-igp
  pass
end-policy
!
router isis 1
  is-type level-2-only
  net 49.0001.0000.0000.0002.00
  log adjacency changes
  affinity-map RED bit-position 28
  flex-algo 128
  priority 228
!
address-family ipv4 unicast
  metric-style wide
  advertise link attributes
  router-id 2.2.2.2
  segment-routing mpls
!
interface Loopback0
  address-family ipv4 unicast
  prefix-sid index 2
  prefix-sid algorithm 128 index 282
!
interface TenGigE0/1/0/3/0
  point-to-point
  address-family ipv4 unicast
!
!
router bgp 65000
  bgp router-id 2.2.2.2
  address-family ipv4 unicast
  address-family vpnv4 unicast
  retain route-target all
!
neighbor-group RR-services-group
  remote-as 65000
  update-source Loopback0
  address-family ipv4 unicast
  address-family vpnv4 unicast
!
neighbor 4.4.4.4
  use neighbor-group RR-services-group
!
vrf Test
  rd auto
  address-family ipv4 unicast
  redistribute connected
!
segment-routing
traffic-eng
  logging
  policy status
!
segment-list sl-cxr1
  index 10 mpls label 16294
!
policy pol-foo
  color 129 end-point ipv4 4.4.4.4
candidate-paths
  preference 100
  explicit segment-list sl-cxrl

Configuration on router R2:

vrf Test
  address-family ipv4 unicast
    import route-target 1:150
    export route-policy SET_COLOR_RED_HI_BW
      export route-target 1:150
    !
    interface TenGigE0/1/0/1
description cxr1 to exr1
ipv4 address 10.0.20.1 255.255.255.0
  extcommunity-set opaque color129-red-igp
d 129
end-set
  route-policy PASS
    pass
end-policy
  route-policy SET_COLOR_RED_HI_BW
    set extcommunity color color129-red-igp
    pass
end-policy
  router isis 1
    is-type level-2-only
    net 49.0001.0000.0000.0004.00
    log adjacency changes
    affinity-map RED bit-position 28
    affinity-map BLUE bit-position 29
    affinity-map GREEN bit-position 30
    flex-algo 128
      priority 228
    flex-algo 129
      priority 229
    flex-algo 130
      priority 230
    address-family ipv4 unicast
      metric-style wide
      advertise link attributes
      router-id 4.4.4.4
    segment-routing mpls
    interface Loopback0
      address-family ipv4 unicast
prefix-sid index 4
prefix-sid algorithm 128 index 284
prefix-sid algorithm 129 index 294
prefix-sid algorithm 130 index 304
!
interface GigabitEthernet0/0/0/0
  point-to-point
  address-family ipv4 unicast
!
interface TenGigE0/1/0/1
  point-to-point
  address-family ipv4 unicast
!
router bgp 65000
  bgp router-id 4.4.4.4
  address-family ipv4 unicast
  address-family vpnv4 unicast
  neighbor-group RR-services-group
    remote-as 65000
    update-source Loopback0
    address-family ipv4 unicast
    address-family vpnv4 unicast
  !
  neighbor 1.1.1.1
    use neighbor-group RR-services-group
  !
  neighbor 2.2.2.2
    use neighbor-group RR-services-group
  !
vrf Test
  rd auto
  address-family ipv4 unicast
  redistribute connected
  !
  neighbor 25.1.1.2
    remote-as 4
    address-family ipv4 unicast
    route-policy PASS in
    route-policy PASS out
  !
  !
segment-routing
!
end
BGP Routes on PE – Color Based Steering
CHAPTER 11

Configure Segment Routing Path Computation Element

The Segment Routing Path Computation Element (SR-PCE) provides stateful PCE functionality by extending the existing IOS-XR PCEP functionality with additional capabilities. SR-PCE is supported on the MPLS data plane and IPv4 control plane.

Note

To install SR-PCE, you need to install an instance of Cisco IOS XRv 9000 Router. Refer to the Cisco IOS XRv 9000 Router Installation and Configuration Guide for more information.

- About SR-PCE, on page 171
- Configure SR-PCE, on page 172
- PCE-Initiated SR Policies for Traffic Management, on page 175
- ACL Support for PCEP Connection, on page 177
- SR-PCE IPv4 Unnumbered Interface Support, on page 177

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.

Note

For more information on PCE, PCC, and PCEP, refer to the Path Computation Element section in the MPLS Configuration Guide for Cisco ASR 9000 Series Routers.

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**

This task explains how to configure SR-PCE.

**Before you begin**

Optionally install and configure an instance of Cisco IOS XRv 9000 Router.

**SUMMARY STEPS**

1. configure
2. pce
3. address ipv4  address
4. state-sync ipv4  address
5. tcp-buffer  size
6. password  {clear | encrypted}  password
7. segment-routing  {strict-sid-only | te-latency}
8. timers
9. keepalive  time
10. minimum-peer-keepalive  time
11. reoptimization  time
12. exit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router#  configure</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>---</td>
<td>------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **Step 2**  | pce  
  Example:  
  ```text
  RP/0/RSP0/CPU0:router(config)# pce
  ``` | Enables PCE and enters PCE configuration mode. |
| **Step 3**  | address ipv4  
  *address*  
  Example:  
  ```text
  RP/0/RSP0/CPU0:router(config-pce)# address ipv4 192.168.0.1
  ``` | Configures a PCE IPv4 address. |
| **Step 4**  | state-sync ipv4  
  *address*  
  Example:  
  ```text
  RP/0/RSP0/CPU0:router(config-pce)# state-sync ipv4 192.168.0.3
  ``` | Configures the remote peer for state synchronization. |
| **Step 5**  | tcp-buffer  
  *size*  
  Example:  
  ```text
  RP/0/RSP0/CPU0:router(config-pce)# tcp-buffer 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:  
  ```text
  RP/0/RSP0/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:  
  ```text
  RP/0/RSP0/CPU0:router(config-pce)# segment-routing strict-sid-only
  ``` | 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. |
| **Step 8**  | timers  
  Example:  
  ```text
  RP/0/RSP0/CPU0:router(config-pce)# timers
  ``` | Enters timer configuration mode. |
| **Step 9**  | keepalive  
  *time*  
  Example:  
  ```text
  ``` | Configures the timer value for locally generated keep-alive messages. The default time is 30 seconds. |
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.

**SUMMARY STEPS**

1. disjoint-path
2. group-id value type {link | node | srlg | srlg-node} [sub-id value]
3. strict
4. lsp {1 | 2} pcc ipv4 address lsp-name lsp_name [shortest-path]

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 disjoint-path</td>
<td>Enters disjoint configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-pce)# disjoint-path</td>
<td></td>
</tr>
</tbody>
</table>

### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 10 minimum-peer-keepalive time</td>
<td>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.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-pce-timers)# minimum-peer-keepalive 30</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 11 reoptimization time</td>
<td>Configures the re-optimization timer. The default timer is 60 seconds.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-pce-timers)# reoptimization 30</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 12 exit</td>
<td>Exits timer configuration mode and returns to PCE configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-pce-timers)# exit</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
</tr>
</tbody>
</table>
| 2    | group-id value type | 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):  
- **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. |
|      | type {link | node | srlg | srlg-node} [sub-id value] | If a pair of paths that meet the requested disjointness level cannot be found, then the paths will automatically fallback to a lower level:  
- 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. |
| 3    | strict | (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. |
| 4    | lsp {1 | 2} pcc ipv4 address lsp-name lsp_name | Adds LSPs to the disjoint group. 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 first LSP specified. |
|      | [shortest-path] | |

**PCE-Initiated SR Policies for Traffic Management**

An SR-TE policy can be configured on the path computation element (PCE) to reduce link congestion or to minimize the number of network touch points.
The PCE-initiated SR-TE policies are entered in PCE configuration mode. For more information on configuring SR-TE policies, see the SR-TE Policy Overview, on page 77.

The PCE collects network information, such as traffic demand and link utilization. When the PCE determines that a link is congested, it identifies one or more flows that are causing the congestion. The PCE finds a suitable path and deploys an SR-TE policy to divert those flows, without moving the congestion to another part of the network. When there is no more link congestion, the policy is removed.

To minimize the number of network touch points, an application, such as a Network Services Orchestrator (NSO), can request the PCE to create an SR-TE policy. PCE deploys the SR-TE policy using PCC-PCE communication protocol (PCEP).

1. PCE sends a PCInitiate message to the PCC.
2. If the PCInitiate message is valid, the PCC sends a PCRpt message; otherwise, it sends PCErr message.
3. If the PCInitiate message is accepted, the PCE updates the SR-TE policy by sending PCUpd message.

You can achieve high-availability by configuring multiple PCEs with SR-TE policies. If the head-end (PCC) loses connectivity with one PCE, another PCE can assume control of the SR-TE policy.

**Configuration Example**

To configure a PCE-initiated SR-TE policy, you must complete the following configurations:

1. Enter PCE configuration mode.
2. Create the segment list.
3. Create the policy.

```bash
/* Enter PCE configuration mode and create the SR-TE segment lists */
Router# configure
Router(config)# pce

/* Create the SR-TE segment lists */
Router(config-pce)# segment-routing
Router(config-pce-sr)# traffic-eng
Router(config-pce-sr-te)# segment-list name addr2a
Router(config-pce-sr-te-sl)# index 1 address ipv4 14.14.14.4
Router(config-pce-sr-te-sl)# exit

/* Create the SR-TE policy */
Router(config-pce-sr-te)# peer ipv4 1.1.1.1
Router(config-pce-sr-te)# policy P1
Router(config-pce-sr-te-policy)# color 2 end-point ipv4 2.2.2.2
Router(config-pce-sr-te-policy)# candidate-paths
Router(config-pce-sr-te-policy-path)# preference 50
Router(config-pce-sr-te-pp-index)# explicit segment-list addr2a
Router(config-pce-sr-te-pp-info)# end
Router(config)#
```
Running Config

```
pce
  segment-routing
  traffic-eng
  segment-list name addr2a
    index 1 address ipv4 14.14.14.4
  !
  peer ipv4 1.1.1.1
  policy P1
  color 2 end-point ipv4 2.2.2.2
  candidate-paths
  preference 50
  explicit segment-list addr2a
  !
  !
```

ACL Support for PCEP Connection

PCE protocol (PCEP) (RFC5440) is a client-server model running over TCP/IP, where the server (PCE) opens a port and the clients (PCC) initiate connections. After the peers establish a TCP connection, they create a PCE session on top of it.

The ACL Support for PCEP Connection feature provides a way to protect a PCE server using an Access Control List (ACL) to restrict IPv4 PCC peers at the time the TCP connection is created based on the source address of a client. When a client initiates the TCP connection, the ACL is referenced, and the client source address is compared. The ACL can either permit or deny the address and the TCP connection will proceed or not.

Refer to the Implementing Access Lists and Prefix Lists chapter in the IP Addresses and Services Configuration Guide for Cisco ASR 9000 Series Routers for detailed ACL configuration information.

To apply an ACL to the PCE, use the `pce peer-filter ipv4 access-list acl_name` command.

SR-PCE IPv4 Unnumbered Interface Support

This feature allows IPv4 unnumbered interfaces to be part of an SR-PCE topology database.

An unnumbered IPv4 interface is not identified by its own unique IPv4 address. Instead, it is identified by the router ID of the node where this interfaces resides and the local SNMP index assigned for this interface.

This feature provides enhancements to the following components:

- IGPs (IS-IS and OSPF):
  - Support the IPv4 unnumbered interfaces in the SR-TE context by flooding the necessary interface information in the topology

- SR-PCE:

  ![Note]

  SR-PCE and path computation clients (PCCs) need to be running Cisco IOS XR 7.0.2 or later.
• Compute and return paths from a topology containing IPv4 unnumbered interfaces.
• Process reported SR policies from a head-end router that contain hops with IPv4 unnumbered adjacencies.

PCEP extensions for IPv4 unnumbered interfaces adhere to IETF RFC8664 “PCEP Extensions for Segment Routing” (https://datatracker.ietf.org/doc/rfc8664/). The unnumbered hops use a Node or Adjacency Identifier (NAI) of type 5. This indicates that the segment in the explicit routing object (ERO) is an unnumbered adjacency with an IPv4 ID and an interface index.

• SR-TE process at the head-end router:
  • Compute its own local path over a topology, including unnumbered interfaces.
  • Process PCE-computed paths that contain hops with IPv4 unnumbered interfaces.
  • Report a path that contains hops with IPv4 unnumbered interfaces to the PCE.

### Configuration Example

The following example shows how to configure an IPv4 unnumbered interface:

```
RP/0/0/CPU0:rtrA(config)# interface GigabitEthernet0/0/0/0
RP/0/0/CPU0:rtrA(config-if)# ipv4 point-to-point
RP/0/0/CPU0:rtrA(config-if)# ipv4 unnumbered Loopback0
```

To bring up the IPv4 unnumbered adjacency under the IGP, configure the link as point-to-point under the IGP configuration. The following example shows how to configure the link as point-to-point under the IGP configuration:

```
RP/0/0/CPU0:rtrA(config)# router ospf one
RP/0/0/CPU0:rtrA(config-ospf)# area 0
RP/0/0/CPU0:rtrA(config-ospf-ar)# interface GigabitEthernet0/0/0/0
RP/0/0/CPU0:rtrA(config-ospf-ar-if)# network point-to-point
```

### Verification

Use the `show ipv4 interface` command to display information about the interface:

```
RP/0/0/CPU0:rtrA# show ipv4 interface GigabitEthernet0/0/0/0 brief
Tue Apr  2 12:59:53.140 EDT
Interface     IP-Address     Status    Protocol
GigabitEthernet0/0/0/0               192.168.0.1 Up       Up
```

This interface shows the IPv4 address of Loopback0.

Use the `show snmp interface` command to find the SNMP index for this interface:

```
RP/0/0/CPU0:rtrA# show snmp interface
Tue Apr  2 13:02:49.190 EDT
ifName : Null0 ifIndex : 3
ifName : Loopback0 ifIndex : 10
ifName : GigabitEthernet0/0/0/0 ifIndex : 6
```

The interface is identified with the pair (IPv4:192.168.0.1, index:6).

Use the `show ospf neighbor` command to display the adjacency:
The output of the `show ospf neighbor gigabitEthernet 0/0/0/0 detail` command:

```
Neighbor 192.168.0.4, interface address 192.168.0.4
  In the area 0 via interface GigabitEthernet0/0/0/0
  Neighbor priority is 1, State is FULL, 6 state changes
  Adjacency SIDs:
    Label: 24001, Dynamic, Unprotected
```

The output of the `show pce ipv4 topology` command:

```
The output of the `show pce ipv4 topology` command is enhanced to display the interface index instead of the IP address for unnumbered interfaces:
```

```
Link[2]: unnumbered local index 6, remote index 4
  Local node:
    OSPF router ID: 192.168.0.1 area ID: 0 ASN: 0
    Remote node:
      TE router ID: 192.168.0.4
    OSPF router ID: 192.168.0.4 area ID: 0 ASN: 0
  Metric: IGP 1, TE 1, Latency 1 microseconds
  Bandwidth: Total 125000000 Bps, Reservable 0 Bps
  Admin-groups: 0x00000000
  Adj SID: 24001 (unprotected)
```

The output of the `show pce lsp detail` command:

```
The output of `show pce lsp detail` command includes unnumbered hops:
```

```
Reported path:
  Metric type: TE, Accumulated Metric 3
  SID[0]: Adj unnumbered, Label 24001, local 192.168.0.1(6), remote 192.168.0.4(4)
  SID[1]: Adj unnumbered, Label 24002, local 192.168.0.4(7), remote 192.168.0.3(7)
  SID[2]: Adj unnumbered, Label 24000, local 192.168.0.3(5), remote 192.168.0.2(5)
  Computed path: (Local PCE)
  Computed Time: Wed Apr 03 11:01:46 EDT 2019 (00:01:06 ago)
  Metric type: TE, Accumulated Metric 3
  SID[0]: Adj unnumbered, Label 24001, local 192.168.0.1(6), remote 192.168.0.4(4)
  SID[1]: Adj unnumbered, Label 24002, local 192.168.0.4(7), remote 192.168.0.3(7)
  SID[2]: Adj unnumbered, Label 24000, local 192.168.0.3(5), remote 192.168.0.2(5)
```
SR-PCE IPv4 Unnumbered Interface Support
Configure Performance Measurement

Network performance metrics is a critical measure for traffic engineering (TE) in service provider networks. Network performance metrics include the following:

- Packet loss
- Delay
- Delay variation
- Bandwidth utilization

These network performance metrics provide network operators information about the performance characteristics of their networks for performance evaluation and help to ensure compliance with service level agreements. The service-level agreements (SLAs) of service providers depend on the ability to measure and monitor these network performance metrics. Network operators can use performance measurement (PM) feature to monitor the network metrics for links and end-to-end TE label switched paths (LSPs).

The following table explains the functionalities supported by performance measurement feature for measuring delay for links or SR policies.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Details</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Profiles</td>
<td>You can configure different profiles for different types of delay measurements. Use the &quot;interfaces&quot; delay profile type for link-delay measurement. The &quot;sr-policy&quot; delay profile type is used for SR policy delay measurements. Delay profile allows you to schedule probe and configure metric advertisement parameters for delay measurement.</td>
<td></td>
</tr>
<tr>
<td>Protocols</td>
<td>MPLS (using RFC6374 with MPLS encap) or Two-Way Active Measurement Protocol (TWAMP) Light (using RFC 5357 with IP/UDP encap).</td>
<td></td>
</tr>
<tr>
<td>Probe and burst scheduling</td>
<td>Schedule probes and configure metric advertisement parameters for delay measurement.</td>
<td></td>
</tr>
<tr>
<td>Metric advertisements</td>
<td>Advertise measured metrics periodically using configured thresholds. Also supports accelerated advertisements using configured thresholds.</td>
<td></td>
</tr>
<tr>
<td>Measurement history and counters</td>
<td>Maintain packet delay and loss measurement history, session counters, and packet advertisement counters.</td>
<td></td>
</tr>
</tbody>
</table>
Link Delay Measurement

The PM for link delay uses the MPLS packet format defined in RFC 6374 for probes. The MPLS packet format requires the remote side line card to be MPLS capable. For link delay measurement, MPLS multicast MAC address is used to send delay measurement probe packets to next-hops. So, the user does not need to configure next-hop addresses for the links. The remote side line card needs to support the MPLS multicast MAC address.

The following figure explains the PM query and response for link delay.

*Figure 6: Performance Measurement for Link Delay*

![PM Query and Response for Link Delay Diagram](image)

The PM query and response for link delay can be described in the following steps:

1. The local-end router sends PM query packets periodically to the remote side once the egress line card on the router applies timestamps on packets.
2. Ingress line card on the remote-end router applies time-stamps on packets as soon as they are received.
3. The remote-end router sends the PM packets containing time-stamps back to the local-end router. The remote-end router time-stamps the packet just before sending it for two-way measurement.
4. The local-end router time-stamps the packet as soon as the packet is received for two-way measurement.
5. One-way delay and optionally two-way delay is measured using the time-stamp values in the PM packet.

Restrictions and Usage Guidelines for PM for Link Delay

The following restrictions and guidelines apply for the PM for link delay feature for different links.

- For protocol pm-mpls, remote-end line card needs to be MPLS-capable.
- For broadcast links, only point-to-point (P2P) links are supported. P2P configuration on IGP is required for flooding the value.
• For one-way delay measurement, clocks should be synchronized on two end-point nodes of the link using PTP.

Configuration Example: PM for Link Delay

This example shows how to configure performance-measurement functionalities for link delay as a global default profile. The default values for the different parameters in the PM for link delay is given as follows:

• probe measurement mode: The default measurement mode for probe is two-way delay measurement. If you are configuring one-way delay measurement, hardware clocks must be synchronized between the local-end and remote-end routers using precision time protocol (PTP).

• protocol:
  • pm-mpls: Interface delay measurement using RFC6374 with MPLS encap. This is the default protocol.
  • pm-udp: Interface delay measurement using RFC 6374-UDP with UDP encap.

• burst interval: Interval for sending probe packet. The default value is 3000 milliseconds and the range is from 30 to 15000 milliseconds.

• computation interval: Interval for metric computation. Default is 30 seconds; range is 1 to 3600 seconds.

• periodic advertisement: Periodic advertisement is enabled by default.

• periodic-advertisement interval: The default value is 120 seconds and the interval range is from 30 to 3600 seconds.

• periodic-advertisement threshold: Checks the minimum-delay metric change for threshold crossing for periodic advertisement. The default value is 10 percent and the range is from 0 to 100 percent.

• periodic-advertisement minimum change: The default value is 1000 microseconds (usec) and the range is from 0 to 100000 microseconds.

• accelerated advertisement: Accelerated advertisement is disabled by default.

• accelerated-advertisement threshold: Checks the minimum-delay metric change for threshold crossing for accelerated advertisement. The default value is 20 percent and the range is from 0 to 100 percent.

• accelerated-advertisement minimum change: The default value is 500 microseconds and the range is from 0 to 100000 microseconds.

```
RP/0/0/CPU0:router (config)# performance-measurement delay-profile interfaces
RP/0/0/CPU0:router (config-pm-dm-intf)# probe
RP/0/0/CPU0:router (config-pm-dm-intf-probe)# measurement-mode one-way
RP/0/0/CPU0:router (config-pm-dm-intf-probe)# burst-interval 60
RP/0/0/CPU0:router (config-pm-dm-intf-probe)# computation-interval 60
RP/0/0/CPU0:router (config-pm-dm-intf-probe)# exit

RP/0/0/CPU0:router (config-pm-dm-intf)# advertisement periodic
RP/0/0/CPU0:router (config-pm-dm-intf-adv-per)# interval 120
RP/0/0/CPU0:router (config-pm-dm-intf-adv-per)# threshold 20
RP/0/0/CPU0:router (config-pm-dm-intf-adv-per)# minimum-change 1000
RP/0/0/CPU0:router (config-pm-dm-intf-adv-per)# exit

RP/0/0/CPU0:router (config-pm-dm-intf)# advertisement accelerated
```
Configure the UDP Destination Port

When you specify PM-UDP protocol, you need to configure the UDP destination port. The UDP port is configured for each PM measurement probe type (delay, loss, protocol, authentication mode, etc.) on querier and responder nodes. The UDP port for each PM measurement probe type must match on querier and responder nodes.

---

**Note**

The same UDP destination port is used for delay measurement for links and SR Policy.

This example shows how to configure the UDP destination port for delay.

```
Router(config)# performance-measurement
Router(config-perf-meas)# protocol pm-udp
Router(config-pm-protocol)# measurement delay unauthenticated
Router(config-pm-proto-mode)# querier-dst-port 12000
```

Enable PM for Link Delay Over an Interface

This example shows how to enable PM for link delay over an interface.

```
RP/0/0/CPU0:router(config)# performance-measurement
RP/0/0/CPU0:router(config-perf-meas)# interface TenGigE0/0/0
RP/0/0/CPU0:router(config-pm-intf)# delay-measurement
RP/0/0/CPU0:router(config-pm-dm-intf)# exit
```

Verification

```
RP/0/0/CPU0:router# show performance-measurement profile interface
Thu Dec 12 14:13:16.029 PST

-------------------------------------------------------------------------------
/0/CPU0
-------------------------------------------------------------------------------
Interface Delay-Measurement:
Profile configuration:
  Measurement Type : Two-Way
  Probe computation interval : 30 (effective: 30) seconds
  Type of services : Traffic Class: 6, DSCP: 48
  Burst interval : 3000 (effective: 3000) mSec
  Burst count : 10 packets
  Encap mode : UDP
  Payload Type : TWAMP-light
  Destination sweeping mode : Disabled
  Periodic advertisement : Enabled
  Interval : 120 (effective: 120) sec
  Threshold : 10%
  Minimum-Change : 500 uSec
  Advertisement accelerated : Disabled
  Threshold crossing check : Minimum-delay
```
show performance-measurement summary detail location 0/2/CPU0

Thu Dec 12 14:09:59.162 PST

0/2/CPU0

Total interfaces : 1
Total SR Policies : 0
Total RSVP-TE tunnels : 0
Total Maximum PPS : 2000 pkts/sec
Total Interfaces PPS : 0 pkts/sec
Maximum Allowed Multi-hop PPS : 2000 pkts/sec
Multi Hop Requested PPS : 0 pkts/sec (0% of max allowed)
Inuse Burst Interval Adjustment Factor : 100% of configuration

Interface Delay-Measurement:
Total active sessions : 1
Counters:
   Packets:
      Total sent : 26
      Total received : 26
   Errors:
      TX:
         Reason interface down : 0
         Reason no MPLS caps : 0
         Reason no IP address : 0
         Reason other : 0
      RX:
         Reason negative delay : 0
         Reason delay threshold exceeded : 0
         Reason missing TX timestamp : 0
         Reason missing RX timestamp : 0
         Reason probe full : 0
         Reason probe not started : 0
         Reason control code error : 0
         Reason control code notif : 0
   Probes:
      Total started : 3
      Total completed : 2
      Total incomplete : 0
      Total advertisements : 0

SR Policy Delay-Measurement:
Total active sessions : 0
Counters:
   Packets:
      Total sent : 0
      Total received : 0
   Errors:
      TX:
         Reason interface down : 0
         Reason no MPLS caps : 0
         Reason no IP address : 0
         Reason other : 0
      RX:
         Reason negative delay : 0
         Reason delay threshold exceeded : 0
         Reason missing TX timestamp : 0
         Reason missing RX timestamp : 0
         Reason probe full : 0
         Reason probe not started : 0
         Reason control code error : 0
         Reason control code notif : 0
Reason control code error : 0
Reason control code notif : 0
Probes:
  Total started : 0
  Total completed : 0
  Total incomplete : 0
  Total advertisements : 0

RSVP-TE Delay-Measurement:
  Total active sessions : 0

Counters:
  Packets:
    Total sent : 0
    Total received : 0
  Errors:
    TX:
        Reason interface down : 0
        Reason no MPLS caps : 0
        Reason no IP address : 0
        Reason other : 0
    RX:
        Reason negative delay : 0
        Reason delay threshold exceeded : 0
        Reason missing TX timestamp : 0
        Reason missing RX timestamp : 0
        Reason probe full : 0
        Reason probe not started : 0
        Reason control code error : 0
        Reason control code notif : 0
  Probes:
    Total started : 0
    Total completed : 0
    Total incomplete : 0
    Total advertisements : 0

Global Delay Counters:
  Total packets sent : 26
  Total query packets received : 26
  Total invalid session id : 0
  Total missing session : 0
RP/0/0/CPU0:router# show performance-measurement interfaces detail
Thu Dec 12 14:16:09.692 PST

-------------------------------------------------------
0/0/CPU0
-------------------------------------------------------

-------------------------------------------------------
0/2/CPU0
-------------------------------------------------------

Interface Name: GigabitEthernet0/2/0/0 (ifh: 0x1004060)
  Delay-Measurement : Enabled
  Loss-Measurement : Disabled
  Configured IPv4 Address : 10.10.10.2
  Configured IPv6 Address : 10:10:10::2
  Link Local IPv6 Address : fe80::3a:6fff:fec9:cd6b
  Configured Next-hop Address : Unknown
  Local MAC Address : 023a.6fc9.cd6a
  Next-hop MAC Address : 0291.e460.6707
  Primary VLAN Tag : None
  Secondary VLAN Tag : None
  State : Up
Delay Measurement session:
  Session ID : 1

Last advertisement:
  Advertised at: Dec 12 2019 14:10:43.138 (326.782 seconds ago)
  Advertised reason: First advertisement
  Advertised delays (uSec): avg: 839, min: 587, max: 8209, variance: 297

Next advertisement:
  Threshold check scheduled in 1 more probe (roughly every 120 seconds)
  Aggregated delays (uSec): avg: 751, min: 589, max: 905, variance: 112
  Rolling average (uSec): 756

Current Probe:
  Started at Dec 12 2019 14:15:43.154 (26.766 seconds ago)
  Packets Sent: 9, received: 9
  Measured delays (uSec): avg: 795, min: 631, max: 1199, variance: 164
  Next probe scheduled at Dec 12 2019 14:16:13.132 (in 3.212 seconds)
  Next burst packet will be sent in 0.212 seconds
  Burst packet sent every 3.0 seconds

Probe samples:
  Packet Rx Timestamp  Measured Delay (nsec)
  Dec 12 2019 14:15:43.156 689223
  Dec 12 2019 14:15:46.156 876561
  Dec 12 2019 14:15:49.156 913548
  Dec 12 2019 14:15:52.157 1199620
  Dec 12 2019 14:15:55.156 794008
  Dec 12 2019 14:15:58.156 631437
  Dec 12 2019 14:16:01.157 656440
  Dec 12 2019 14:16:04.157 658267
  Dec 12 2019 14:16:07.157 736880

You can also use the following commands for verifying the PM for link delay on the local-end router.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show performance-measurement history probe interfaces [interface]</td>
<td>Displays the PM link-delay probe history for interfaces.</td>
</tr>
<tr>
<td>show performance-measurement history aggregated interfaces [interface]</td>
<td>Displays the PM link-delay aggregated history for interfaces.</td>
</tr>
<tr>
<td>show performance-measurement history advertisement interfaces [interface]</td>
<td>Displays the PM link-delay advertisement history for interfaces.</td>
</tr>
<tr>
<td>show performance-measurement counters [interface] [location location-name]</td>
<td>Displays the PM link-delay session counters.</td>
</tr>
</tbody>
</table>

You can also use the following commands for verifying the PM for link-delay configuration on the remote-end router.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show performance-measurement responder summary [location location-name]</td>
<td>Displays the PM for link-delay summary on the remote-end router (responder).</td>
</tr>
<tr>
<td>show performance-measurement responder interfaces [interface]</td>
<td>Displays PM for link-delay for interfaces on the remote-end router.</td>
</tr>
</tbody>
</table>
SR Policy End-to-End Delay Measurement

The PM for SR Policy uses the MPLS packet format defined in RFC 6374 for probes. The MPLS packet format requires the remote-side line card to be MPLS-capable.

The extended TE link delay metric (minimum-delay value) can be used to compute paths for SR policies as an optimization metric or as an accumulated delay bound.

There is a need to monitor the end-to-end delay experienced by the traffic sent over an SR policy to ensure that the delay does not exceed the requested “upper-bound” and violate SLAs. You can verify the end-to-end delay values before activating the candidate-path or the segment lists of the SR policy in forwarding table, or to deactivate the active candidate-path or the segment lists of the SR policy in forwarding table.

---

**Note**

The end-to-end delay value of an SR policy will be different than the path computation result (for example, the sum of TE link delay metrics) due to several factors, such as queuing delay within the routers.

---

**Figure 7: Performance Measurement for SR Policy End-to-End Delay**

The PM query and response for end-to-end SR Policy delay can be described in the following steps:

1. The local-end router sends PM query packets periodically to the remote side once the egress line card on the router applies timestamps on packets.

2. The ingress line card on the remote-end router applies time-stamps on packets as soon as they are received.

3. The remote-end router sends the PM packets containing time-stamps back to the local-end router.

4. One-way delay is measured using the time-stamp values in the PM packet.

---

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show performance-measurement responder</td>
<td>Displays the PM link-delay session counters on the remote-end router.</td>
</tr>
<tr>
<td>counters [interface interface] [location</td>
<td></td>
</tr>
<tr>
<td>location-name]</td>
<td></td>
</tr>
</tbody>
</table>
Restrictions and Usage Guidelines for PM for SR Policy Delay

Hardware clocks must be synchronized between the querier and the responder nodes of the link using PTP for one-way delay measurement.

Configuring Performance Measurement Parameters

This example shows how to configure performance-measurement parameters for SR policy delay as a global default profile. The default values for the different parameters in the PM for SR policy delay is given as follows:

- **probe**: The default mode for probe is one-way delay measurement.
- **burst interval**: Interval for sending probe packet. The default value is 3000 milliseconds and the range is from 30 to 15000 milliseconds.
- **computation interval**: Interval for metric computation. Default is 30 seconds; range is 1 to 3600 seconds.
- **protocol**:
  - **pm-mpls**: SR Policy delay measurement using RFC6374 with MPLS encap. This is the default protocol.
  - **pm-udp**: SR Policy delay measurement using RFC 6374-UDP with UDP encap.
- **tos**: Type of Service
  - **dscp value**: The default value is 48 and the range is from 0 to 63.
  - **traffic-class value**: The default value is 6 and the range is from 0 to 7.
- **advertisement threshold-check**: minimum-delay/maximum-delay - The default value of periodic advertisement threshold-check is maximum-delay.
- **periodic advertisement**: Periodic advertisement is enabled by default.
- **periodic-advertisement interval**: The default value is 120 seconds and the interval range is from 30 to 3600 seconds.
- **periodic-advertisement threshold**: Checks the minimum-delay metric change for threshold crossing for periodic advertisement. The default value is 10 percent and the range is from 0 to 100 percent.
- **periodic-advertisement minimum-change**: The default value is 500 microseconds (usec) and the range is from 0 to 100000 microseconds.
- **accelerated advertisement**: Accelerated advertisement is disabled by default.
- **accelerated-advertisement threshold**: Checks the minimum-delay metric change for threshold crossing for accelerated advertisement. The default value is 20 percent and the range is from 0 to 100 percent.
- **accelerated-advertisement minimum**: The default value is 500 microseconds and the range is from 1 to 100000 microseconds.

Router(config) # performance-measurement delay-profile sr-policy
Router(config-pm-dm-srpolicy) # probe
Router(config-pm-dm-srpolicy-probe) # burst-interval 60
Router(config-pm-dm-srpolicy-probe) # computation-interval 60
Router(config-pm-dm-srpolicy-probe) # protocol pm-udp
Router(config-pm-dm-srpolicy-probe) # tos dscp 63
Configure the UDP Destination Port

When you specify PM-UDP protocol, you need to configure the UDP destination port. The UDP port is configured for each PM measurement probe type (delay, loss, protocol, authentication mode, etc.) on querier and responder nodes. The UDP port for each PM measurement probe type must match on querier and responder nodes.

The same UDP destination port is used for delay measurement for links and SR Policy.

This example shows how to configure the UDP destination port for delay.

```
Router(config)# performance-measurement
Router(config-perf-meas)# protocol pm-udp
Router(config-pm-protocol)# measurement delay unauthenticated
Router(config-pm-proto-mode)# querier-dst-port 12000
```

Enable Performance Measurement for SR Policy

This example shows how to enable PM for SR policy delay for a specific policy.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# policy foo
Router(config-sr-te-policy)# performance-measurement
Router(config-sr-te-policy-perf-meas)# delay-measurement
```

SR Policy Probe IP/UDP ECMP Hashing Configuration

This example shows how to configure SR Policy ECMP IP-hashing mode.

- The destination IPv4 address 127.x.x.x – 127.y.y.y is used in the Probe messages to take advantages of 3-tuple IP hashing (source-address, destination-address, and local router ID) for ECMP paths of SR-MPLS Policy.
The destination IPv4 address must be 127/8 range (loopback), otherwise it will be rejected.

- One PM session is always created for the actual endpoint address of the SR Policy.
- You can specify the number of IP addresses to sweep. The range is from 0 (default, no sweeping) to 128.
- Platforms may have a limitation for large label stack size to not check IP address for hashing.

```
Router(config)# performance-measurement delay-profile sr-policy
Router(config-pm-dm-srpolicy)# probe
Router(config-pm-dm-srpolicy-probe)# sweep
Router(config-pm-dm-srpolicy-probe-sweep)# destination ipv4 127.0.0.1 range 28
```

Verification

```
Router# show performance-measurement sr-policy brief
Mon Jan 20 18:48:41.002 PST
-------------------------------------------------------------------------------
0/0/CPU0
-------------------------------------------------------------------------------
Policy Name       LSP ID  Tx/Rx Avg/Min/Max/Variance
-------------------------------------------------------------------------------
srte_c_10_ep_192.168.0.4  2    6/6  27012/26906/27203/106

Router# show performance-measurement sr-policy name srte_c_10_ep_192.168.0.4 detail verbose
Mon Jan 20 18:44:22.400 PST
-------------------------------------------------------------------------------
0/0/CPU0
-------------------------------------------------------------------------------
SR Policy name: srte_c_10_ep_192.168.0.4
Color: 10
Endpoint: 192.168.0.4
Number of candidate-paths: 1

Candidate-Path:
  Instance: 2
  Preference: 100
  Protocol-origin: Configured
  Discriminator: 100
  Source address: 192.168.0.2
  Reverse path label: Not configured
  Number of segment-lists: 1

Last advertisement:
  No advertisements have ocurred
Next advertisement:
  Check scheduled at the end of the current probe (roughly every 30 seconds)
Aggregated delays (uSec): avg: 45218, min: 26512, max: 82600, variance: 18706
Rolling average (uSec): 45218
Last probe:
  Packets Sent: 9, received: 9
  Measured delays (uSec): avg: 45218, min: 26512, max: 82600, variance: 18706
Current Probe:
  Started at Jan 20 2020 18:44:19.170 (3.453 seconds ago)
  Packets Sent: 3, received: 3
```
Measured delays (uSec): avg: 26588, min: 26558, max: 26630, variance: 30
Next probe scheduled at Jan 20 2020 18:44:34.166 (in 11.543 seconds)
Next burst packet will be sent in 1.543 seconds
Burst packet sent every 5.0 seconds
Liveness Detection: Disabled

Segment-List : R4
16004
Number of atomic paths : 3
Last advertisement:
  No advertisements have occurred
Next advertisement:
  Aggregated delays (uSec): avg: 45218, min: 26512, max: 82600, variance: 18706
  Rolling average (uSec): 45218
  Last probe:
    Packets Sent: 9, received: 9
    Measured delays (uSec): avg: 45218, min: 26512, max: 82600, variance: 18706
  Current probe:
    Packets Sent: 3, received: 3
    Measured delays (uSec): avg: 26588, min: 26558, max: 26630, variance: 30
  Liveness Detection: Disabled

Atomic path:
  Hops : 127.0.0.0
  Session ID : 33554434
  Last advertisement:
    No advertisements have occurred
  Next advertisement:
    Aggregated delays (uSec): avg: 45407, min: 26629, max: 82600, variance: 18778
    Rolling average (uSec): 45407
    Last Probe:
      Packets Sent: 3, received: 3
      Measured delays (uSec): avg: 45407, min: 26629, max: 82600, variance: 18778
    Current Probe:
      Packets Sent: 1, received: 1
      Measured delays (uSec): avg: 26630, min: 26630, max: 26630, variance: 0
  Probe samples:
    Packet Rx Timestamp Measured Delay (nsec)
    Jan 20 2020 18:44:19.198 26630730
  Liveness Detection: Disabled

Atomic path:
  Hops : 127.0.1.0
  Session ID : 33554435
  Last advertisement:
    No advertisements have occurred
  Next advertisement:
    Aggregated delays (uSec): avg: 45128, min: 26521, max: 81961, variance: 18607
    Rolling average (uSec): 45128
    Last Probe:
      Packets Sent: 3, received: 3
      Measured delays (uSec): avg: 45128, min: 26521, max: 81961, variance: 18607
    Current Probe:
      Packets Sent: 1, received: 1
      Measured delays (uSec): avg: 26576, min: 26576, max: 26576, variance: 0
  Probe samples:
    Packet Rx Timestamp Measured Delay (nsec)
    Jan 20 2020 18:44:19.198 26576938
  Liveness Detection: Disabled

Atomic path:
  Hops : 192.168.0.4
  Session ID : 33554433
  Last advertisement:
No advertisements have occurred

Next advertisement:
Aggregated delays (uSec): avg: 45119, min: 26512, max: 81956, variance: 18607
Rolling average (uSec): 45119

Last Probe:
Packets Sent: 3, received: 3
Measured delays (uSec): avg: 45119, min: 26512, max: 81956, variance: 18607

Current Probe:
Packets Sent: 1, received: 1
Measured delays (uSec): avg: 26558, min: 26558, max: 26558, variance: 0

Probe samples:
Packet Rx Timestamp Measured Delay (nsec)
Jan 20 2020 18:44:19.198 26558375

Liveness Detection: Disabled

Router# show performance-measurement history probe sr-policy
Mon Jan 20 18:46:55.445 PST

Segment Routing Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 7.0.x
### Atomic path: 127.0.0.1

#### Delay-Measurement history (uSec):

<table>
<thead>
<tr>
<th>Probe Start Timestamp</th>
<th>Pkt (TX/RX)</th>
<th>Average</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 20 2020 18:46:34.174</td>
<td>3/3</td>
<td>26927</td>
<td>26747</td>
<td>27070</td>
</tr>
<tr>
<td>Jan 20 2020 18:46:04.173</td>
<td>3/3</td>
<td>26895</td>
<td>26647</td>
<td>27164</td>
</tr>
<tr>
<td>Jan 20 2020 18:45:49.172</td>
<td>3/3</td>
<td>27054</td>
<td>26764</td>
<td>27293</td>
</tr>
<tr>
<td>Jan 20 2020 18:45:34.172</td>
<td>3/3</td>
<td>26801</td>
<td>26694</td>
<td>27005</td>
</tr>
<tr>
<td>Jan 20 2020 18:45:19.171</td>
<td>3/3</td>
<td>26807</td>
<td>26524</td>
<td>27093</td>
</tr>
<tr>
<td>Jan 20 2020 18:45:04.171</td>
<td>3/3</td>
<td>27226</td>
<td>26938</td>
<td>27591</td>
</tr>
<tr>
<td>Jan 20 2020 18:44:49.171</td>
<td>3/3</td>
<td>26976</td>
<td>26644</td>
<td>27143</td>
</tr>
<tr>
<td>Jan 20 2020 18:44:34.171</td>
<td>3/3</td>
<td>26880</td>
<td>26767</td>
<td>27265</td>
</tr>
<tr>
<td>Jan 20 2020 18:44:19.170</td>
<td>3/3</td>
<td>26994</td>
<td>26630</td>
<td>27422</td>
</tr>
<tr>
<td>Jan 20 2020 18:44:06.543</td>
<td>3/3</td>
<td>45407</td>
<td>26629</td>
<td>82600</td>
</tr>
</tbody>
</table>

### Atomic path: 192.168.0.4

#### Delay-Measurement history (uSec):

<table>
<thead>
<tr>
<th>Probe Start Timestamp</th>
<th>Pkt (TX/RX)</th>
<th>Average</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 20 2020 18:46:34.174</td>
<td>3/3</td>
<td>26848</td>
<td>26684</td>
<td>26967</td>
</tr>
<tr>
<td>Jan 20 2020 18:46:04.173</td>
<td>3/3</td>
<td>26787</td>
<td>26581</td>
<td>26939</td>
</tr>
<tr>
<td>Jan 20 2020 18:45:49.172</td>
<td>3/3</td>
<td>26954</td>
<td>26728</td>
<td>27180</td>
</tr>
<tr>
<td>Jan 20 2020 18:45:34.172</td>
<td>3/3</td>
<td>26724</td>
<td>26577</td>
<td>26957</td>
</tr>
<tr>
<td>Jan 20 2020 18:45:19.171</td>
<td>3/3</td>
<td>26705</td>
<td>26452</td>
<td>27032</td>
</tr>
<tr>
<td>Jan 20 2020 18:45:04.171</td>
<td>3/3</td>
<td>27043</td>
<td>26972</td>
<td>27124</td>
</tr>
<tr>
<td>Jan 20 2020 18:44:49.171</td>
<td>3/3</td>
<td>26848</td>
<td>26550</td>
<td>27062</td>
</tr>
<tr>
<td>Jan 20 2020 18:44:34.171</td>
<td>3/3</td>
<td>26800</td>
<td>26562</td>
<td>27204</td>
</tr>
<tr>
<td>Jan 20 2020 18:44:19.170</td>
<td>3/3</td>
<td>26927</td>
<td>26576</td>
<td>27327</td>
</tr>
<tr>
<td>Jan 20 2020 18:44:06.543</td>
<td>3/3</td>
<td>45128</td>
<td>26521</td>
<td>81961</td>
</tr>
</tbody>
</table>

Router# show performance-measurement counters sr-policy name srte_c_10_ep_192.168.0.4

Mon Jan 20 18:47:55.499 PST

-----------------------------------------------------------------------------------------------

0/0/CPU0

SR Policy name: srte_c_10_ep_192.168.0.4

Candidate-Path:

- Instance: 2
- Preference: 100
- Protocol-origin: Configured
- Discriminator: 100

Packets:
- Total sent: 141
- Total received: 141

Errors:
- Total sent errors: 0
- Total received errors: 0

Probes:
- Total started: 16
<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total completed</td>
<td>15</td>
</tr>
<tr>
<td>Total incomplete</td>
<td>0</td>
</tr>
<tr>
<td>Total advertisements</td>
<td>2</td>
</tr>
<tr>
<td>Segment-List</td>
<td>R4</td>
</tr>
<tr>
<td>Packets</td>
<td></td>
</tr>
<tr>
<td>Total sent</td>
<td>141</td>
</tr>
<tr>
<td>Total received</td>
<td>141</td>
</tr>
<tr>
<td>Errors</td>
<td></td>
</tr>
<tr>
<td>Total sent errors</td>
<td>0</td>
</tr>
<tr>
<td>Total received errors</td>
<td>0</td>
</tr>
<tr>
<td>Probes</td>
<td></td>
</tr>
<tr>
<td>Total started</td>
<td>16</td>
</tr>
<tr>
<td>Total completed</td>
<td>15</td>
</tr>
<tr>
<td>Total incomplete</td>
<td>0</td>
</tr>
<tr>
<td>Total advertisements</td>
<td>2</td>
</tr>
</tbody>
</table>
SR Policy End-to-End Delay Measurement
Configure Topology-Independent Loop-Free Alternate (TI-LFA)

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

TI-LFA supports IPv4 only.

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

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

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

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

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

• Behaviors and Limitations of TI-LFA, on page 197
• Configuring TI-LFA for IS-IS, on page 198
• Configuring TI-LFA for OSPF, on page 199
• TI-LFA Node and SRLG Protection: Examples, on page 201
• Configuring Global Weighted SRLG Protection, on page 202

Behaviors and Limitations of TI-LFA

The behaviors and limitations of TI-LFA are listed below:

• The Network Virtualization (nV) satellite access interfaces do not support TI-LFA.
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 41.
- Enter the following commands in global configuration mode:

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

**SUMMARY STEPS**

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

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>router isis instance-id</td>
<td>Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# router isis 1</td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td>You can change the level of routing to be performed by a particular routing instance by using the is-type router configuration command.</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>interface type interface-path-id</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Configure Topology-Independent Loop-Free Alternate (TI-LFA)

#### 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.

#### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis)# interface GigabitEthernet0/0/2/1</td>
<td>You can configure TI-LFA under Ethernet-based interfaces and logical Bundle-Ethernet interfaces.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis)# interface Bundle-Ether1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> address-family ipv4 [unicast]</td>
<td>Specifies the IPv4 address family, and enters router address family configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-isis-if)# address-family ipv4 unicast</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> fast-reroute per-prefix</td>
<td>Enables per-prefix fast reroute.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> fast-reroute per-prefix ti-lfa</td>
<td>Enables per-prefix TI-LFA fast reroute link protection.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix ti-lfa</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> fast-reroute per-prefix tiebreaker {node-protecting</td>
<td>srlg-disjoint} index priority</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix srlg-disjoint index 100</td>
<td>The same attribute cannot be configured more than once on an interface.</td>
</tr>
</tbody>
</table>

TI-LFA has been successfully configured for segment routing.

#### 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 61.
- Enter the following commands in global configuration mode:

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

**SUMMARY STEPS**

1. `configure`
2. `router ospf process-name`
3. `area area-id`
4. `interface type interface-path-id`
5. `fast-reroute per-prefix`
6. `fast-reroute per-prefix ti-lfa`
7. `fast-reroute per-prefix tiebreaker (node-protecting | srlg-disjoint) index priority`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enables OSPF routing for the specified routing process, and places the router in router configuration mode.</td>
</tr>
<tr>
<td><code>router ospf process-name</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config)# router ospf 1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enters area configuration mode.</td>
</tr>
<tr>
<td><code>area area-id</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-ospf)# area 1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td><code>interface type interface-path-id</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-ospf-ar)# interface GigabitEthernet0/0/2/1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>You can configure TI-LFA under Ethernet-based interfaces and logical Bundle-Ethernet interfaces.</td>
</tr>
</tbody>
</table>
Configure Topology-Independent Loop-Free Alternate (TI-LFA)

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar)# interface Bundle-Ether1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> fast-reroute per-prefix</td>
<td>Enables per-prefix fast reroute.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar-if)# fast-reroute per-prefix</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> fast-reroute per-prefix ti-lfa</td>
<td>Enables per-prefix TI-LFA fast reroute link protection.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar-if)# fast-reroute per-prefix ti-lfa</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> fast-reroute per-prefix tiebreaker {node-protecting</td>
<td>srlg-disjoint} index priority</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis-ar-if)# fast-reroute per-prefix srlg-disjoint index 100</td>
<td></td>
</tr>
</tbody>
</table>

TI-LFA has been successfully configured for segment routing.

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

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

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

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

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

```plaintext
erouter 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

```plaintext
erouter 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.

Configuring Global Weighted SRLG Protection

A shared risk link group (SRLG) is a set of links sharing a common resource and thus shares the same risk of failure. The existing loop-free alternate (LFA) implementations in interior gateway protocols (IGPs) support SRLG protection. However, the existing implementation considers only the directly connected links while computing the backup path. Hence, SRLG protection may fail if a link that is not directly connected but shares the same SRLG is included while computing the backup path. Global weighted SRLG protection feature provides better path selection for the SRLG by associating a weight with the SRLG value and using the weights of the SRLG values while computing the backup path.

To support global weighted SRLG protection, you need information about SRLGs on all links in the area topology. For IS-IS, you can flood SRLGs for remote links or manually configuring SRLGs on remote links. The administrative weight (cost) of the SRLG can be configured using the `admin-weight` command. This command can be applied for all SRLG (global), or for a specific (named) SRLG. The default (global) admin-weight value is 1 for IS-IS.

Configuration Examples: Global Weighted SRLG Protection for IS-IS

There are three types of configurations that are supported for the global weighted SRLG protection feature for IS-IS:

- Local SRLG with global weighted SRLG protection
- Remote SRLG flooding
• Remote SRLG static provisioning

This example shows how to configure the local SRLG with global weighted SRLG protection feature.

```
RP/0/RP0/CPU0:router(config)# srlg
RP/0/RP0/CPU0:router(config-srlg)# interface TenGigE0/0/0
RP/0/RP0/CPU0:router(config-srlg-if)# name group1
RP/0/RP0/CPU0:router(config-srlg-if)# exit
RP/0/RP0/CPU0:router(config-srlg)# interface TenGigE0/0/1
RP/0/RP0/CPU0:router(config-srlg-if)# name group1
RP/0/RP0/CPU0:router(config-srlg-if)# exit
RP/0/RP0/CPU0:router(config-srlg)# name group value 100
RP/0/RP0/CPU0:router(config-srlg)# exit
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)# fast-reroute per-prefix srlg-protection weighted-global
RP/0/RP0/CPU0:router(config-isis-af)# fast-reroute per-prefix tiebreaker index 1
RP/0/RP0/CPU0:router(config-isis-af)# exit
RP/0/RP0/CPU0:router(config-isis-if)# point-to-point
RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix ti-lfa
RP/0/RP0/CPU0:router(config-isis-if)# exit
RP/0/RP0/CPU0:router(config-isis)# srlg
RP/0/RP0/CPU0:router(config-isis-srlg)# name group1
RP/0/RP0/CPU0:router(config-isis-srlg-name)# admin-weight 5000
```

This example shows how to configure the global weighted SRLG protection feature with remote SRLG flooding. The configuration includes local and remote router configuration. On the local router, the global weighted SRLG protection is enabled by using the `fast-reroute per-prefix srlg-protection weighted-global` command. In the remote router configuration, you can control the SRLG value flooding by using the `advertise application lfa link-attributes srlg` command. You should also globally configure SRLG on the remote router.

The local router configuration for global weighted SRLG protection with remote SRLG flooding is as follows:

```
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)# fast-reroute per-prefix srlg-protection weighted-global
RP/0/RP0/CPU0:router(config-isis-af)# fast-reroute per-prefix tiebreaker index 1
RP/0/RP0/CPU0:router(config-isis-af)# exit
RP/0/RP0/CPU0:router(config-isis-if)# interface TenGigE0/0/0
RP/0/RP0/CPU0:router(config-isis-if)# point-to-point
RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix ti-lfa
RP/0/RP0/CPU0:router(config-isis-if)# exit
RP/0/RP0/CPU0:router(config-isis)# srlg
RP/0/RP0/CPU0:router(config-isis-srlg)# name group1
RP/0/RP0/CPU0:router(config-isis-srlg-name)# admin-weight 5000
```

The remote router configuration for global weighted SRLG protection with remote SRLG flooding is as follows:
This example shows configuring the global weighted SRLG protection feature with static provisioning of SRLG values for remote links. You should perform these configurations on the local router.

```
RP/0/RP0/CPU0:router(config)# srlg
RP/0/RP0/CPU0:router(config-srlg)# interface TenGigE0/0/0
RP/0/RP0/CPU0:router(config-srlg-if)# name group1
RP/0/RP0/CPU0:router(config-srlg-if)# exit
RP/0/RP0/CPU0:router(config-srlg)# interface TenGigE0/0/1
RP/0/RP0/CPU0:router(config-srlg-if)# name group1
RP/0/RP0/CPU0:router(config-srlg-if)# name group1 value 100
RP/0/RP0/CPU0:router(config-srlg)# exit
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)# fast-reroute per-prefix srlg-protection weighted-global
RP/0/RP0/CPU0:router(config-isis-af)# fast-reroute per-prefix ti-lfa
RP/0/RP0/CPU0:router(config-isis-af)# exit
RP/0/RP0/CPU0:router(config-isis)# srlg
RP/0/RP0/CPU0:router(config-isis-srlg)# name group1
RP/0/RP0/CPU0:router(config-isis-srlg-name)# admin-weight 5000
RP/0/RP0/CPU0:router(config-isis-srlg-name)# static ipv4 address 10.0.4.1 next-hop ipv4 address 10.0.4.2
RP/0/RP0/CPU0:router(config-isis-srlg-name)# static ipv4 address 10.0.4.2 next-hop ipv4 address 10.0.4.1
```
CHAPTER 14

Configure Segment Routing Microloop Avoidance

The Segment Routing Microloop Avoidance feature enables link-state routing protocols, such as IS-IS and OSPF, to prevent or avoid microloops during network convergence after a topology change.

- About Segment Routing Microloop Avoidance, on page 205
- Configure Segment Routing Microloop Avoidance for IS-IS, on page 205
- Configure Segment Routing Microloop Avoidance for OSPF, on page 206

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.

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 41.
- Enter the following commands in global configuration mode:

  `Router(config)# ipv4 unnumbered mpls traffic-eng Loopback0`
  `Router(config)# mpls traffic-eng`
Configure Segment Routing Microloop Avoidance for OSPF

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

### SUMMARY STEPS

1. `configure`
2. `router isis instance-id`
3. `address-family ipv4 [ unicast ]`
4. `microloop avoidance segment-routing`
5. `microloop avoidance rib-update-delay delay-time`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure</code></td>
<td>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 <code>is-type</code> router configuration command.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Specifies the IPv4 address family and enters router address family configuration mode.</td>
</tr>
<tr>
<td><code>router isis instance-id</code></td>
<td>Enables Segment Routing Microloop Avoidance.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Specifies the amount of time the node uses the microloop avoidance policy before updating its forwarding table. The <code>delay-time</code> is in milliseconds. The range is from 1-60000. The default value is 5000.</td>
</tr>
<tr>
<td><code>address-family ipv4 [ unicast ]</code></td>
<td>Enables Segment Routing Microloop Avoidance.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Enables Segment Routing Microloop Avoidance.</td>
</tr>
<tr>
<td><code>microloop avoidance segment-routing</code></td>
<td>Enables Segment Routing Microloop Avoidance.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Enables Segment Routing Microloop Avoidance.</td>
</tr>
<tr>
<td><code>microloop avoidance rib-update-delay delay-time</code></td>
<td>Enables Segment Routing Microloop Avoidance.</td>
</tr>
<tr>
<td><code>microloop avoidance rib-update-delay 3000</code></td>
<td>Enables Segment Routing Microloop Avoidance.</td>
</tr>
</tbody>
</table>
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 61.

- Enter the following commands in global configuration mode:

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

**SUMMARY STEPS**

1. `configure`
2. `router ospf process-name`
3. `microloop avoidance segment-routing`
4. `microloop avoidance rib-update-delay delay-time`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>router ospf process-name</code></td>
<td>Enables OSPF routing for the specified routing process, and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# router ospf 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>microloop avoidance segment-routing</code></td>
<td>Enables Segment Routing Microloop Avoidance.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# microloop avoidance segment-routing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>microloop avoidance rib-update-delay delay-time</code></td>
<td>Specifies the amount of time the node uses the microloop avoidance policy before updating its forwarding table. The <code>delay-time</code> is in milliseconds. The range is from 1-60000. The default value is 5000.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# microloop avoidance rib-update-delay 3000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Configure Segment Routing Microloop Avoidance for OSPF
CHAPTER 15

Configure Segment Routing Mapping Server

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

- Segment Routing Mapping Server, on page 209
- Segment Routing and LDP Interoperability, on page 210
- Configuring Mapping Server, on page 212
- Enable Mapping Advertisement, on page 214
- Enable Mapping Client, on page 216

Segment Routing Mapping Server

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

- A router that acts as a mapping server allows the user to configure SID mapping entries to specify the prefix-SIDs for some or all prefixes. This creates the local SID-mapping policy. The local SID-mapping policy contains non-overlapping SID-mapping entries. The mapping server advertises the local SID-mapping policy to the mapping clients.

- A router that acts as a mapping client receives and parses remotely received SIDs from the mapping server to create remote SID-mapping entries.

- A router that acts as a mapping server and mapping client uses the remotely learnt and locally configured mapping entries to construct the non-overlapping consistent active mapping policy. IGP instance uses the active mapping policy to calculate the prefix-SIDs of some or all prefixes.

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

- Locally configured mapping entries must not overlap each other.

- The mapping server takes the locally configured mapping policy, as well as remotely learned mapping entries from a particular IGP instance, as input, and selects a single mapping entry among overlapping mapping entries according to the preference rules for that IGP instance. The result is an active mapping policy that consists of non-overlapping consistent mapping entries.

- At steady state, all routers, at least in the same area or level, must have identical active mapping policies.
Segment Routing Mapping Server Restrictions

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

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

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

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

Segment Routing and LDP Interoperability

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

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

Example: Segment Routing LDP Interoperability

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

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

In this mixed network:

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

**LDP to SR**

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

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

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

**SR to LDP**

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

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

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

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

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

## SUMMARY STEPS

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

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CFU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>segment-routing</code></td>
<td>Enables segment routing.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CFU0:router(config)# segment-routing</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>mapping-server</code></td>
<td>Enables mapping server configuration mode.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CFU0:router(config-sr)# mapping-server</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>prefix-sid-map</code></td>
<td>Enables prefix-SID mapping configuration mode.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CFU0:router(config-sr)# prefix-sid-map</code></td>
<td>Note: Two-way prefix SID can be enabled directly under IS-IS or through a mapping server.</td>
</tr>
<tr>
<td><strong>Step 5</strong> `address-family ipv4</td>
<td>ipv6`</td>
</tr>
<tr>
<td>Example: This example shows the address-family for ipv4: <code>RP/0/RSP0/CFU0:router(config-sr-ms-map)# address-family ipv4</code></td>
<td></td>
</tr>
</tbody>
</table>
### Configure Segment Routing Mapping Server

#### Purpose

This example shows the address-family for ipv6:

```
RP/0/RSP0/CPU0:router(config-sr-ms-map)#
address-family ipv6
```

Add SID-mapping entries in the active local mapping policy. In the configured example:

- Prefix 10.1.1.1/32 is assigned prefix-SID 10, prefix 10.1.1.2/32 is assigned prefix-SID 11, ..., prefix 10.1.1.199/32 is assigned prefix-SID 200
- Prefix 20.1.0.0/16 is assigned prefix-SID 400, prefix 20.2.0.0/16 is assigned prefix-SID 401, ..., and so on.

### Step 6

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

**Example:**

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

### Step 7

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:

- **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.

### Step 8

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

**Example:**

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

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

**Note** Specify the address family for IS-IS.
### Enable Mapping Advertisement

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

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

### Configure Mapping Advertisement for IS-IS

#### SUMMARY STEPS

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

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.</td>
</tr>
<tr>
<td><code>router isis instance-id</code></td>
<td>Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>You can change the level of routing to be performed by a particular routing instance by using the <code>is-type</code> router configuration command.</td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config)# router isis 1</code></td>
<td></td>
</tr>
</tbody>
</table>

| Step 2 | `address-family { ipv4 | ipv6 } [ unicast ]`  | Specifies the IPv4 or IPv6 address family, and enters router address family configuration mode. |
|---------|---------------------------------------------|---------------------------------------------|
| **Example:** | The following is an example for ipv4 address family: | |
| `RP/0/RSP0/CPU0:router(config-isis)# address-family ipv4 unicast` | |

<table>
<thead>
<tr>
<th>Step 3</th>
<th><code>segment-routing prefix-sid-map advertise-local</code></th>
<th>Configures IS-IS to advertise locally configured prefix-SID mappings.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

### Enable the advertisement of the local SID-mapping policy in the IGP.

#### Purpose

- Flags:
  - Number of mapping entries: 2

---

### What to do next

Enable the advertisement of the local SID-mapping policy in the IGP.
### Configure Mapping Advertisement for OSPF

#### SUMMARY STEPS

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

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables OSPF routing for the specified routing instance, and places the router in router configuration mode.</td>
</tr>
<tr>
<td><code>router ospf process-name</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# <code>router ospf 1</code></td>
</tr>
</tbody>
</table>

---

### Configure Segment Routing Mapping Server

### Configure Mapping Advertisement for OSPF
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>segment-routing prefix-sid-map advertise-local</td>
<td>Configures OSPF to advertise locally configured prefix-SID mappings.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# segment-routing prefix-sid-map advertise-local</td>
<td>commit — Saves the configuration changes and remains within the configuration session.</td>
</tr>
<tr>
<td></td>
<td>end — Prompts user to take one of these actions:</td>
</tr>
<tr>
<td></td>
<td>• Yes — Saves configuration changes and exits the configuration session.</td>
</tr>
<tr>
<td></td>
<td>• No — Exits the configuration session without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Cancel — Remains in the configuration session, without committing the configuration changes.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Use the commit or end command.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>show ospf database opaque-area</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# show ospf database opaque-area</td>
<td></td>
</tr>
<tr>
<td>&lt;...removed...&gt;</td>
<td></td>
</tr>
<tr>
<td>Extended Prefix Range TLV: Length: 24</td>
<td></td>
</tr>
<tr>
<td>AF</td>
<td>0</td>
</tr>
<tr>
<td>Prefix</td>
<td>10.1.1.1/32</td>
</tr>
<tr>
<td>Range Size</td>
<td>200</td>
</tr>
<tr>
<td>Flags</td>
<td>0x0</td>
</tr>
<tr>
<td>SID sub-TLV: Length: 8</td>
<td></td>
</tr>
<tr>
<td>Flags</td>
<td>0x60</td>
</tr>
<tr>
<td>MTID</td>
<td>0</td>
</tr>
<tr>
<td>Algo</td>
<td>0</td>
</tr>
<tr>
<td>SID Index</td>
<td>10</td>
</tr>
</tbody>
</table>

**Enable Mapping Client**

By default, mapping client functionality is enabled.

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

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

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

```
RP/0/RSP0/CPU0:router(config)# router ospf 1
RP/0/RSP0/CPU0:router(config-ospf)# segment-routing prefix-sid-map receive
```
Enable Mapping Client
Using Segment Routing Traffic Matrix

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

- Segment Routing Traffic Matrix, on page 219
- Traffic Collector Process, on page 219
- Configuring Traffic Collector, on page 220
- Displaying Traffic Information, on page 222

Segment Routing Traffic Matrix

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

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

Traffic Collector Process

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

Perform these tasks to configure the traffic collector.

SUMMARY STEPS

1. `configure`
2. `traffic-collector`
3. `statistics collection-interval value`
4. `statistics history-size value`
5. `statistics history-timeout value`
6. `interface type 13-interface-address`
7. Use the `commit` or `end` command.

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>2</td>
<td><code>traffic-collector</code>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>RP/0/RP0/CPU0:router(config)# traffic-collector</code></td>
<td>Enables traffic collector and places the router in traffic collector configuration mode.</td>
</tr>
<tr>
<td>3</td>
<td><code>statistics collection-interval value</code>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>RP/0/RP0/CPU0:router(config-tc)# statistics collection-interval 5</code></td>
<td>(Optional) Sets the frequency that the traffic collector collects and posts data, in minutes. Valid values are 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, and 60. The default interval is 1.</td>
</tr>
</tbody>
</table>
| 4    | `statistics history-size value`<br>**Example:**<br>`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. |
| 5    | `statistics history-timeout value`<br>**Example:**<br>`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.) |
| 6    | `interface type l3-interface-address`<br>**Example:**<br>`RP/0/RP0/CPU0:router(config-tc)# interface TenGigE 0/1/0/3` | Identifies interfaces that handle external traffic. Only L3 interfaces are supported for external traffic. |
| 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:  
• **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/RSP0/CPU0:router# show traffic-collector external-interface
  Interface Status
  ---------------- ----------------
  Te0/1/0/3 Enabled
  Te0/1/0/4 Enabled
  ```

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

  ```
  RP/0/RSP0/CPU0:router# show traffic-collector ipv4 counters prefix 1.1.1.10/32 detail
  Prefix: 1.1.1.10/32 Label: 16010 State: Active
  **Base:**
  Average over the last 5 collection intervals:
  Packet rate: 9496937 pps, Byte rate: 9363979882 Bps
  History of counters:
  23:01 - 23:02: Packets 9379529, Bytes: 9248215594
  23:00 - 23:01: Packets 9687124, Bytes: 9551504264
  22:59 - 23:00: Packets 9539200, Bytes: 9405651200
  **TM Counters:**
  Average over the last 5 collection intervals:
  Packet rate: 9528754 pps, Byte rate: 9357236821 Bps
  History of counters:
  23:01 - 23:02: Packets 9400815, Bytes: 9231600330
  23:00 - 23:01: Packets 9699455, Bytes: 9524864810
  22:59 - 23:00: Packets 9579889, Bytes: 9407450998
  ```

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

- Display the counter history database for a tunnel:

  ```
  RP/0/RSP0/CPU0:router# show traffic-collector counters tunnels tunnel-te 1 detail
  Tunnel: tt1 State: Active
  Average over the last 5 collection intervals:
  Packet rate: 9694434 pps, Byte rate: 9597489858 Bps
  History of counters:
  ```
This output shows the average Pcounter (packets, bytes), the Pcounter history, and the collection interval for the tunnel.
CHAPTER 17

Using Segment Routing OAM

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

- MPLS Ping and Traceroute for BGP and IGP Prefix-SID, on page 225
- Examples: MPLS Ping, Traceroute, and Tree Trace for Prefix-SID, on page 226
- MPLS LSP Ping and Traceroute Nil FEC Target, on page 228
- Examples: LSP Ping and Traceroute for Nil_FEC Target, on page 228
- Segment Routing Ping and Traceroute, on page 230
- Segment Routing Policy Nil-FEC Ping and Traceroute, on page 234
- Segment Routing over IPv6 OAM, on page 236
- Segment Routing Data Plane Monitoring, on page 237

**MPLS Ping and Traceroute for BGP and IGP Prefix-SID**

MPLS Ping and Traceroute operations for Prefix SID are supported for various BGP and IGP scenarios, for example:

- Within an IS-IS level or OSPF area
- Across IS-IS levels or OSPF areas
- Route redistribution from IS-IS to OSPF and from OSPF to IS-IS
- Anycast Prefix SID
- Combinations of BGP and LDP signaled LSPs

The MPLS LSP Ping feature is used to check the connectivity between ingress Label Switch Routers (LSRs) and egress LSRs along an LSP. MPLS LSP ping uses MPLS echo request and reply messages, similar to Internet Control Message Protocol (ICMP) echo request and reply messages, to validate an LSP. The destination IP address of the MPLS echo request packet is different from the address used to select the label stack. The destination IP address is defined as a 127.x.y.z/8 address and it prevents the IP packet from being IP switched to its destination, if the LSP is broken.

The MPLS LSP Traceroute feature is used to isolate the failure point of an LSP. It is used for hop-by-hop fault localization and path tracing. The MPLS LSP Traceroute feature relies on the expiration of the Time to
Live (TTL) value of the packet that carries the echo request. When the MPLS echo request message hits a transit node, it checks the TTL value and if it is expired, the packet is passed to the control plane, else the message is forwarded. If the echo message is passed to the control plane, a reply message is generated based on the contents of the request message.

The MPLS LSP Tree Trace (traceroute multipath) operation is also supported for BGP and IGP Prefix SID. MPLS LSP Tree Trace provides the means to discover all possible equal-cost multipath (ECMP) routing paths of an LSP to reach a destination Prefix SID. It uses multipath data encoded in echo request packets to query for the load-balancing information that may allow the originator to exercise each ECMP. When the packet TTL expires at the responding node, the node returns the list of downstream paths, as well as the multipath information that can lead the operator to exercise each path in the MPLS echo reply. This operation is performed repeatedly for each hop of each path with increasing TTL values until all ECMP are discovered and validated.

MPLS echo request packets carry Target FEC Stack sub-TLVs. The Target FEC sub-TLVs are used by the responder for FEC validation. The BGP and IGP IPv4 prefix sub-TLV has been added to the Target FEC Stack sub-TLV. The IGP IPv4 prefix sub-TLV contains the prefix SID, the prefix length, and the protocol (IS-IS or OSPF). The BGP IPv4 prefix sub-TLV contains the prefix SID and the prefix length.

### Examples: MPLS Ping, Traceroute, and Tree Trace for Prefix-SID

These examples use the following topology:

![Topology Diagram](image)

#### MPLS Ping for Prefix-SID

RP/0/RSP0/CPU0# router-arizona# ping mpls ipv4 1.1.1.4/32
Thu Dec 17 01:01:42.301 PST

Sending 5, 100-byte MPLS Echos to 1.1.1.4,
  timeout is 2 seconds, send interval is 0 msec:

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

Type escape sequence to abort.

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
**MPLS Traceroute for Prefix-SID**

RP/0/RSP0/CPU0:router-arizona# traceroute mpls ipv4 1.1.1.4/32
Thu Dec 17 14:45:05.563 PST


Type escape sequence to abort.

0 12.12.12.1 MRU 4470 [Labels: 16004 Exp: 0]
L 1 12.12.12.2 MRU 4470 [Labels: 16004 Exp: 0] 3 ms
L 2 23.23.23.3 MRU 4470 [Labels: implicit-null Exp: 0] 3 ms
! 3 34.34.34.4 11 ms

**MPLS Tree Trace for Prefix-SID**

RP/0/RSP0/CPU0:router-arizona# traceroute mpls multipath ipv4 1.1.1.4/32
Thu Dec 17 14:55:46.549 PST

Starting LSP Path Discovery for 1.1.1.4/32


Type escape sequence to abort.

LL!
Path 0 found,
output interface TenGigE0/0/0/0 nexthop 12.12.12.2 source 12.12.12.1 destination 127.0.0.0
L!
Path 1 found,
output interface TenGigE0/0/0/0 nexthop 12.12.12.2 source 12.12.12.1 destination 127.0.0.2
LL!
Path 2 found,
output interface TenGigE0/0/0/1 nexthop 15.15.15.5 source 15.15.15.1 destination 127.0.0.1
L!
Path 3 found,
output interface TenGigE0/0/0/1 nexthop 15.15.15.5 source 15.15.15.1 destination 127.0.0.0

Paths (found/broken/unexplored) (4/0/0)
Echo Request (sent/fail) (10/0)
Echo Reply (received/timeout) (10/0)
Total Time Elapsed 53 ms
MPLS LSP Ping and Traceroute Nil FEC Target

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

- label stack
- outgoing interface
- nexthop address

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

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

**Table 7: LSP Ping and Traceroute Nil FEC Commands**

<table>
<thead>
<tr>
<th>Command Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>ping mpls nil-fec labels {label[,label]} [output {interface tx-interface} [nexthop nexthop-ip-addr]]</td>
</tr>
<tr>
<td>traceroute mpls nil-fec labels {label[,label]} [output {interface tx-interface} [nexthop nexthop-ip-addr]]</td>
</tr>
</tbody>
</table>

Examples: LSP Ping and Traceroute for Nil_FEC Target

These examples use the following topology:

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

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

<table>
<thead>
<tr>
<th>Label</th>
<th>Label</th>
<th>Prefix</th>
<th>Outgoing Interface</th>
<th>Next Hop</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>16004</td>
<td>Pop</td>
<td>No ID</td>
<td>G10/2/0/1</td>
<td>10.1.1.4</td>
<td>1392</td>
</tr>
<tr>
<td>16005</td>
<td>Pop</td>
<td>No ID</td>
<td>G10/2/0/2</td>
<td>10.1.2.2</td>
<td>0</td>
</tr>
<tr>
<td>16005</td>
<td>16005</td>
<td>No ID</td>
<td>G10/2/0/0</td>
<td>10.1.1.4</td>
<td>0</td>
</tr>
<tr>
<td>16007</td>
<td>16007</td>
<td>No ID</td>
<td>G10/2/0/1</td>
<td>10.1.2.2</td>
<td>4752</td>
</tr>
<tr>
<td>16007</td>
<td>16007</td>
<td>No ID</td>
<td>G10/2/0/0</td>
<td>10.1.1.4</td>
<td>0</td>
</tr>
<tr>
<td>24000</td>
<td>Pop</td>
<td>SR Adj (idx 0)</td>
<td>G10/2/0/0</td>
<td>10.1.1.4</td>
<td>0</td>
</tr>
<tr>
<td>24001</td>
<td>Pop</td>
<td>SR Adj (idx 2)</td>
<td>G10/2/0/0</td>
<td>10.1.1.4</td>
<td>0</td>
</tr>
</tbody>
</table>
Ping Nil FEC Target

RP/0/RSP0/CPU0# ping mpls nil-fec labels 16005,16007 output interface GigabitEthernet 0/2/0/1 nexthop 10.1.1.4 repeat 1
Sending 1, 72-byte MPLS Echos with Nil FEC labels 16005,16007, timeout is 2 seconds, send interval is 0 msec:


Type escape sequence to abort.

! Success rate is 100 percent (1/1), round-trip min/avg/max = 1/1/1 ms
Total Time Elapsed 0 ms

Traceroute Nil FEC Target

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


Type escape sequence to abort.

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

Segment Routing Ping

The MPLS LSP ping feature is used to check the connectivity between ingress and egress of LSP. MPLS LSP ping uses MPLS echo request and reply messages, similar to Internet Control Message Protocol (ICMP) echo request and reply messages, to validate an LSP. Segment routing ping is an extension of the MPLS LSP ping to perform the connectivity verification on the segment routing control plane.

Note

Segment routing ping can only be used when the originating device is running segment routing.

You can initiate the segment routing ping operation only when Segment Routing control plane is available at the originator, even if it is not preferred. This allows you to validate the SR path before directing traffic over the path. Segment Routing ping can use either generic FEC type or SR control-plane FEC type (SR-OSPF, SR-ISIS). In mixed networks, where some devices are running MPLS control plane (for example, LDP) or do not understand SR FEC, generic FEC type allows the device to successfully process and respond to the echo request. By default, generic FEC type is used in the target FEC stack of segment routing ping echo request. Generic FEC is not coupled to a particular control plane; it allows path verification when the advertising protocol is unknown or might change during the path of the echo request. If you need to specify the target FEC, you can select the FEC type as OSPF, IS-IS, or BGP. This ensures that only devices that are running segment routing control plane, and can therefore understand the segment routing IGP FEC, respond to the echo request.

Configuration Examples

These examples show how to use segment routing ping to test the connectivity of a segment routing control plane. In the first example, FEC type is not specified. You can also specify the FEC type as shown in the other examples.

```
RP/0/RSP0/CPU0:router# ping sr-mpls 10.1.1.2/32
Sending 5, 100-byte MPLS Echos to 10.1.1.2/32,
timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'!L' - labeled output interface, '!B' - unlabeled output interface,
'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
'P' - no rx intf label prot, 'p' - premature termination of LSP,
'R' - transit router, 'I' - unknown upstream index,
'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/5 ms
RP/0/RSP0/CPU0:router# ping sr-mpls 10.1.1.2/32 fec-type generic
Sending 5, 100-byte MPLS Echos to 10.1.1.2/32,
timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
```
'L' - labeled output interface, 'B' - unlabeled output interface,
'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
'P' - no rx intf label prot, 'P' - premature termination of LSP,
'R' - transit router, 'I' - unknown upstream index,
'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms

RP/0/RSP0/CPU0:router# ping sr-mpls 10.1.1.2/32 fec-type igp ospf

Sending 5, 100-byte MPLS Echos to 10.1.1.2/32,
timeout is 2 seconds, send interval is 0 msec:

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

Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms

RP/0/RSP0/CPU0:router# ping sr-mpls 10.1.1.2/32 fec-type igp isis

Sending 5, 100-byte MPLS Echos to 10.1.1.2/32,
timeout is 2 seconds, send interval is 0 msec:

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

Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms

RP/0/RSP0/CPU0:router# ping sr-mpls 10.1.1.2/32 fec-type bgp

Sending 5, 100-byte MPLS Echos to 10.1.1.2/32,
timeout is 2 seconds, send interval is 0 msec:

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

Type escape sequence to abort.
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms

Segment Routing Traceroute

The MPLS LSP traceroute is used to isolate the failure point of an LSP. It is used for hop-by-hop fault localization and path tracing. The MPLS LSP traceroute feature relies on the expiration of the Time to Live (TTL) value of the packet that carries the echo request. When the MPLS echo request message hits a transit node, it checks the TTL value and if it is expired, the packet is passed to the control plane, else the message is forwarded. If the echo message is passed to the control plane, a reply message is generated based on the contents of the request message. Segment routing traceroute feature extends the MPLS LSP traceroute functionality to segment routing networks.

Similar to segment routing ping, you can initiate the segment routing traceroute operation only when Segment Routing control plane is available at the originator, even if it is not preferred. Segment Routing traceroute can use either generic FEC type or SR control-plane FEC type (SR-OSPF, SR-ISIS). By default, generic FEC type is used in the target FEC stack of segment routing traceroute echo request. If you need to specify the target FEC, you can select the FEC type as OSPF, IS-IS, or BGP. This ensures that only devices that are running segment routing control plane, and can therefore understand the segment routing IGP FEC, respond to the echo request.

The existence of load balancing at routers in an MPLS network provides alternate paths for carrying MPLS traffic to a target router. The multipath segment routing traceroute feature provides a means to discover all possible paths of an LSP between the ingress and egress routers.

Configuration Examples

These examples show how to use segment routing traceroute to trace the LSP for a specified IPv4 prefix SID address. In the first example, FEC type is not specified. You can also specify the FEC type as shown in the other examples.

```
RP/0/RSP0/CPU0:router# traceroute sr-mpls 10.1.1.2/32
Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds
Codes: '! - success, 'Q' - request not sent, '.' - timeout,
      'L' - labeled output interface, 'B' - unlabeled output interface,
      'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
      'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
      'P' - no rx intf label prot, 'p' - premature termination of LSP,
      'R' - transit router, 'I' - unknown upstream index,
      'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

0 10.12.12.1 MRU 1500 [Labels: implicit-null Exp: 0]  
! 1 10.12.12.2 3 ms
```

```
RP/0/RSP0/CPU0:router# traceroute sr-mpls 10.1.1.2/32  fec-type generic
Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds
Codes: '! - success, 'Q' - request not sent, '.' - timeout,
      'L' - labeled output interface, 'B' - unlabeled output interface,
      'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
      'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
```

Using Segment Routing OAM

Segment Routing Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 7.0.x
'P' - no rx intf label prot, 'p' - premature termination of LSP,
'R' - transit router, 'I' - unknown upstream index,
'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

0 10.12.12.1 MRU 1500 [Labels: implicit-null Exp: 0]
! 1 10.12.12.2 2 ms

RP/0/RSP0/CPU0:router# traceroute sr-mpls 10.1.1.2/32 fec-type igp ospf

Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds

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

Type escape sequence to abort.

0 10.12.12.1 MRU 1500 [Labels: implicit-null Exp: 0]
! 1 10.12.12.2 2 ms

RP/0/RSP0/CPU0:router# traceroute sr-mpls 10.1.1.2/32 fec-type igp isis

Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds

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

Type escape sequence to abort.

0 10.12.12.1 MRU 1500 [Labels: implicit-null Exp: 0]
! 1 10.12.12.2 2 ms

RP/0/RSP0/CPU0:router# traceroute sr-mpls 10.1.1.2/32 fec-type bgp

Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds

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

Type escape sequence to abort.

0 10.12.12.1 MRU 1500 [Labels: implicit-null/implicit-null Exp: 0/0]
! 1 10.12.12.2 2 ms
This example shows how to use multipath traceroute to discover all the possible paths for a IPv4 prefix SID.

RP/0/RSP0/CPU0:router# traceroute sr-mpls multipath 10.1.1.2/32

Starting LSP Path Discovery for 10.1.1.2/32


Type escape sequence to abort.

! Path 0 found, output interface GigabitEthernet0/0/0/2 nexthop 10.13.13.2 source 10.13.13.1 destination 127.0.0.0

! Path 1 found, output interface Bundle-Ether1 nexthop 10.12.12.2 source 10.12.12.1 destination 127.0.0.0

Paths (found/broken/unexplored) (2/0/0) Echo Request (sent/fail) (2/0) Echo Reply (received/timeout) (2/0) Total Time Elapsed 14 ms

Segment Routing Policy Nil-FEC Ping and Traceroute

Segment routing OAM supports Nil-FEC LSP ping and traceroute operations to verify the connectivity for segment routing MPLS data plane. For the existing Nil-FEC ping and traceroute commands, you need to specify the entire outgoing label stack, outgoing interface, as well as the next hop. SR policy Nil-FEC ping and SR policy Nil-FEC traceroute enhancements extend the data plane validation functionality of installed SR policies through Nil-FEC ping and traceroute commands while simplifying the operational process. Instead of specifying the entire outgoing label-stack, interface, and next-hop, you can use the policy name or the policy binding-SID label value to initiate Nil-FEC ping and traceroute operations for the SR policies. Specification of outgoing interface and next-hop is also not required for policy Nil-FEC OAM operations.

Restrictions and Usage Guidelines

The following restrictions and guidelines apply for this feature:

- You cannot select a specific candidate path for SR policy Nil-FEC ping and traceroute.
- You cannot use SR policy Nil-FEC ping or traceroute for non-selected candidate paths.

Examples: SR Policy Nil-FEC Ping

These examples show how to use SR policy Nil-FEC ping for a SR policy. The first example refers the SR policy-name while the second example refers the BSID.

RP/0/0/CP00:router# ping sr-mpls nil-fec policy name POLICY1
Thu Feb 22 06:56:50.006 PST Sending 5, 100-byte MPLS Echos with Nil FEC for SR-TE Policy POLICY1, timeout is 2 seconds, send interval is 0 msec:
Codes: '! - success, 'Q' - request not sent, '.' - timeout,
'L' - labeled output interface, 'B' - unlabeled output interface,
'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
'P' - no rx intf label prot, 'p' - premature termination of LSP,
'R' - transit router, 'I' - unknown upstream index,
'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/5/22 ms

Examples: SR Policy Nil-FEC Traceroute

These examples show how to use SR policy Nil-FEC traceroute for a SR policy. The first example refers the SR policy-name while the second example refers the binding SID (BSID).

```
RP/0/0/CPU0:router# ping sr-mpls nil-fec policy binding-sid 100001
Thu Dec 17 12:41:02.381 EST
Sending 5, 100-byte MPLS Echos with Nil FEC with labels [16002,16003],
timeout is 2 seconds, send interval is 0 msec:
Codes: '! - success, 'Q' - request not sent, '.' - timeout,
'L' - labeled output interface, 'B' - unlabeled output interface,
'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
'P' - no rx intf label prot, 'p' - premature termination of LSP,
'R' - transit router, 'I' - unknown upstream index,
'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/3/3 ms

RP/0/0/CPU0:router# traceroute sr-mpls nil-fec policy name POLICY1
Thu Feb 22 06:57:03.637 PST
Tracing MPLS Label Switched Path with Nil FEC for SR-TE Policy POLICY1, timeout is 2 seconds
Codes: '! - success, 'Q' - request not sent, '.' - timeout,
'L' - labeled output interface, 'B' - unlabeled output interface,
'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
'P' - no rx intf label prot, 'p' - premature termination of LSP,
'R' - transit router, 'I' - unknown upstream index,
'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.

0 11.11.11.1 MRU 4470 [Labels: 16002/16004/explicit-null Exp: 0/0/0] 2 ms
L 1 99.1.2.2 MRU 4470 [Labels: implicit-null/explicit-null Exp: 0/0] 4 ms
  2 14.14.14.3 2 ms

RP/0/0/CPU0:router# traceroute sr-mpls nil-fec binding-sid 100001
Tracing MPLS Label Switched Path with Nil FEC with labels [16002/16004], timeout is 2 seconds
Codes: '! - success, 'Q' - request not sent, '.' - timeout,
'L' - labeled output interface, 'B' - unlabeled output interface,
'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
'P' - no rx intf label prot, 'p' - premature termination of LSP,
'R' - transit router, 'I' - unknown upstream index,
'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.

0 99.1.2.1 MRU 4470 [Labels: 16002/16004/explicit-null Exp: 0/0/0] 3 ms
L 1 99.1.2.2 MRU 4470 [Labels: 16004/explicit-null Exp: 0/0] 3 ms
```
Segment Routing over IPv6 OAM

Segment Routing over IPv6 data plane (SRv6) implementation adds a new type of routing extension header. Hence, the existing ICMPv6 mechanisms including ping and traceroute can be used in the SRv6 network. There is no change in the way ping and traceroute operations work for IPv6- or SRv6-capable nodes in an SRv6 network.

Restrictions and Usage Guidelines

The following restriction applies for SRv6 OAM:

- Ping to an SRv6 SID is not supported.

Examples: SRv6 OAM

The following example shows using ping in an SRv6 network.

```
RP/0/RP0/CPU0:Router# ping ipv6 2001::33:33:33:33
Mon Sep 17 20:04:10.068 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001::33:33:33:33, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms
```

The following example shows using traceroute in an SRv6 network.

```
RP/0/RP0/CPU0:Router# traceroute ipv6 2001::33:33:33:33 probe 1 timeout 0 srv6
Fri Sep 14 15:59:25.170 UTC
Type escape sequence to abort.
Tracing the route to 2001::33:33:33:33
3 2001::44:44:44:44 2 msec
4 2001::33:33:33:33 3 msec
```

The following example shows using traceroute in an SRv6 network without an SRH.

```
RP/0/RSP1/CPU0:Router# traceroute ipv6 2001::44:44:44:44 srv6
Wed Jan 16 14:35:27.511 UTC
Type escape sequence to abort.
Tracing the route to 2001::44:44:44:44
1 2001::2:2:2:2 3 msec 2 msec 2 msec
2 2001::44:44:44:44 3 msec 3 msec 3 msec
```

The following example shows using ping for a specified IP address in the VRF.

```
RP/0/RP0/CPU0:Router# ping 10.15.15.1 vrf red
Mon Sep 17 20:07:10.085 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.15.15.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```
The following example shows using traceroute for a specified IP address in the VRF:

```
RP/0/RP0/CPU0:Router# traceroute 10.15.15.1 vrf red
Mon Sep 17 20:07:18.478 UTC
Type escape sequence to abort.
Tracing the route to 10.15.15.1
  1 10.15.15.1 3 msec 2 msec 2 msec
```

The following example shows using traceroute for CE1 (4.4.4.5) to CE2 (5.5.5.5) in the VRF:

```
RP/0/RP0/CPU0:Router# traceroute 5.5.5.5 vrf a
Wed Jan 16 15:08:46.264 UTC
Type escape sequence to abort.
Tracing the route to 5.5.5.5
  1 14.14.14.1 5 msec 1 msec 1 msec
  2 15.15.15.1 3 msec 2 msec 2 msec
  3 15.15.15.2 2 msec * 3 msec
```

### Segment Routing Data Plane Monitoring

Unreported traffic drops in MPLS networks could be difficult to detect and isolate. They can be caused by user configuration, out-of-sync neighbors, or incorrect data-plane programming. Segment Routing Data Plane Monitoring (SR DPM) provides a scalable solution to address data-plane consistency verification and detection of unreported traffic drops. SR DPM validates the actual data plane status of all FIB entries associated with SR IGP prefix SIDs.

The primary benefits of SR DPM include:

- **Automation** – A node automatically verifies the integrity of the actual forwarding entries exercised by transit traffic.

- **Comprehensive Coverage** – Tests validate forwarding consistency for each set of destination prefixes across each combination of upstream and downstream neighbors and across all ECMP possibilities.

- **Scalability** – SR DPM is a highly scalable solution due to its localized detection process.

- **Proactive and Reactive modes of operation** – Solution caters to both continuous and on-demand verification.

- **Standards-based** – SR DPM uses existing MPLS OAM tools and leverages SR to enforce test traffic path.

DPM performs data plane validation in two phases:

- **Adjacency Validation**—Using special MPLS echo request packets, adjacency validation ensures that all local links are able to forward and receive MPLS traffic correctly from their neighbors. It also ensures that DPM is able to verify all local adjacency SID labels and to flag any inconsistencies, including traffic drops, forwarding by the local or neighboring device to an incorrect neighbor that is not associated with the specified adjacency, or forwarding by the local or neighboring device to the correct neighbor but over an incorrect link not associated with the specified adjacency. DPM validates the following adjacencies for each link when available:

  - Unprotected adjacency
• Protected adjacency
• Static adjacency
• Dynamic adjacency
• Shared adjacency

Note
Observe the following limitations for adjacency validation:

• The adjacency validation phase only validates links that are participating in IGP (OSPF and IS-IS) instances. If one or more link is not part of the IGP, it will not be validated since there are no Adjacency SID labels.

• Adjacency validation only validates physical and bundle links, including broadcast links.

• Prefix Validation—Prefix validation identifies any forwarding inconsistency of any IGP Prefix SID reachable from the device. The validation is done for all upstream and downstream neighbor combinations of each prefix SID, and identifies inconsistencies in the downstream neighbor. The prefix validation phase simulates customer traffic path by validating both ingress and egress forwarding chain at the DPM processing node.

Since prefix validation is localized to a device running DPM as well as its immediate neighbors, it does not suffer from scale limitations of end-to-end monitoring.

Prefix validation builds on top of adjacency validation by using special MPLS echo requests that travel to the upstream node, return to the DPM-processing node, and time-to-live (TTL) expire at the immediate downstream node, thus exercising entire forwarding path towards the downstream.

Note
Observe the following limitations for prefix validation:

• Because prefix validation builds on top of adjacency validation, if a link is not part of adjacency validation, it is not used in prefix validation.

• If all adjacencies are marked as “Faulty” during adjacency validation, prefix validation is not performed.

• If a node only has downstream links at a specific node, but no upstream node (possible in certain PE node scenarios), Prefix Validation is not performed.

• Prefix validation does not support TI-LFA.

DPM maintains a database of all prefixes and adjacencies being monitored.

The prefix database is populated by registering as a redistribution client to RIB, which enables DPM to keep the database up-to-date whenever IGP pushes a new prefix SID to RIB, deletes an existing prefix SID, or when the path of an existing prefix SID is modified.

DPM maintains the following prefix data:

• IPv4 Prefix
• Prefix Length
• Prefix SID label
• Error stats

DPM also maintains a list of all local adjacencies. DPM maintains a database that contains local links, their respective local and remote adjacency labels and IP addresses, and error stats.

**SR-DPM Operation: Example**

The following SR-DPM operation example use the following scenarios:

*Figure 8: Test Iteration A Path*

Test forwarding entry to node 9
Node 2 generates Probe 1 to hash on north link to Node 3 with TTL of 2

Result: **PASS**
Line card connected to 1 has a valid forwarding entry on the first ECMP path towards 9

*Figure 9: Test Iteration B Path*

Test forwarding entry to node 9
Node 2 generates Probe 2 to hash on south link to Node 3 with TTL of 2

Result: **FAIL**
Line card connected to 1 has an invalid forwarding entry on the second ECMP path towards 9
Node 2 is a DPM-capable device. DPM is enabled in proactive mode to perform forwarding consistency tests for all prefix-SIDs in the network. For each destination prefix, the router identifies the directly connected upstream and downstream neighbors used to reach a given destination.

Using node 9 as the prefix under test (prefix-SID = 16009), node 1 is designated as the upstream node and node 3 as the downstream nodes with 2 ECMPs.

1. Node 2 generates test traffic (MPLS OAM ping with source_ip of node 2) to test its forwarding for every upstream/downstream combination. In this case, two combinations exist:
   - Prefix-SID node 9 - test iteration A path = Node 2 to Node 1 to Node 2 to Node 3 (via top ECMP)
   - Prefix-SID node 9 - test iteration B path = Node 2 to Node 1 to Node 2 to Node 3 (via bottom ECMP)

2. Node 2 adds a label stack in order to enforce the desired path for the test traffic. For example, two labels are added to the packet for test iterations A and B:
   - The top label is equal to the adjacency-SID on node 1 for the interface facing node 2 (adjacency SID = 24012). The bottom label is the prefix-SID under test (16009). The test traffic is sent on the interface facing node 1.
   - The top label (after being POPed at node 1) causes the test traffic to come back to node 2. This returning traffic is completely hardware-switched based on the forwarding entry for the prefix-SID under test (16009). Note that the labeled test traffic has a time-to-live (TTL) of 2 and it will never be forwarded beyond the downstream router(s).
   - When test traffic reaches node 3, a TTL expired response is sent back to node 2. If the response packet arrives over the expected interface (top ECMP link) then the forwarding verification on node 2 for the first iteration towards node 9 is considered to be a success.
   - The difference between the test traffic for test iteration A and B in this example is the destination_ip of the MPLS OAM ping. Node 2 calculates them in this order to exercise a given ECMP path (if present). Thus, test traffic for iteration A is hashed onto the top ECMP and test traffic for iteration B is hashed onto the bottom ECMP link.

3. The DPM tests are then repeated for the remaining prefix-SIDs in the network

**Configure SR DPM**

To configure SR-DPM, complete the following configurations:

- Enable SR DPM
- Configure SR DPM interval timer
- Configure SR DPM rate limit

**Enable SR DPM**

Use the `mpls oam dpm` command to enable SR DPM and enter MPLS OAM DPM command mode.

```
Router(config)# mpls oam dpm
Router(config-oam-dpm)#
```
Configure SR DPM Interval Timer

Use the `interval minutes` command in MPLS OAM DPM command mode to specify how often to run DPM scan. The range is from 1 to 3600 minutes. The default is 30 minutes.

```
Router(config-oam-dpm)# interval 240
Router(config-oam-dpm)#
```

Configure SR DPM Rate Limit

Use the `pps pps` command in MPLS OAM DPM command mode to rate limit the number of echo request packets per second (PPS) generated by DPM. The range is from 1 to 250 PPS. The default is 50 PPS.

```
Router(config-oam-dpm)# pps 45
Router(config-oam-dpm)#
```

Verification

Router# `show mpls oam dpm summary`  
Displays the overall status of SR-DPM from the last run.
Router# `show mpls oam dpm adjacency summary`  
Displays the result of DPM adjacency SID verification for all local interfaces from the last run.
Router# `show mpls oam dpm adjacency interface`  
Displays the result of DPM adjacency SID verification for all adjacencies for the specified local interface.
Router# `show mpls oam dpm counters`  
Outputs various counters for DPM from last run as well as since the start of DPM process.
Router# `show mpls oam dpm prefix summary`  
Displays the result of DPM prefix SID verification for all reachable IGP prefix SIDs from the last run.
Router# `show mpls oam dpm prefix prefix`  
Displays the result of DPM prefix SID verification for the specified prefix including all upstream and downstream combinations.
Router# `show mpls oam dpm trace`  
Returns logged traces for DPM.

In addition, the existing `show mpls oam` command is extended to specify DPM counters.

```
Router# show mpls oam counters packet dpm
```