



Configure Segment Routing for IS-IS Protocol

Integrated Intermediate System-to-Intermediate System (IS-IS), Internet Protocol Version 4 (IPv4), is a standards-based Interior Gateway Protocol (IGP). The Cisco IOS XR software implements the IP routing capabilities described in International Organization for Standardization (ISO)/International Engineering Consortium (IEC) 10589 and RFC 1995, and adds the standard extensions for single topology and multitopology IS-IS for IP Version 6 (IPv6).

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

- [Enabling Segment Routing for IS-IS Protocol, on page 1](#)
- [Configuring a Prefix-SID on the IS-IS Enabled Loopback Interface, on page 3](#)
- [Weighted Anycast SID-Aware Path Computation, on page 6](#)
- [Configuring an Adjacency SID, on page 11](#)
- [Configuring Bandwidth-Based Local UCMP, on page 17](#)
- [Conditional Prefix Advertisement, on page 18](#)
- [Segment Routing ECMP-FEC Optimization, on page 20](#)

Enabling Segment Routing for IS-IS Protocol

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

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

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

Before you begin

Your network must support the MPLS Cisco IOS XR software feature before you enable segment routing for IS-IS on your router.



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

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	router isis instance-id Example: RP/0/RP0/CPU0:router(config)# router isis isp	Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode. Note You can change the level of routing to be performed by a particular routing instance by using the is-type router configuration command.
Step 3	address-family { ipv4 ipv6 } [unicast] Example: RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast	Specifies the IPv4 or IPv6 address family, and enters router address family configuration mode.
Step 4	metric-style wide [level { 1 2 }] Example: RP/0/RP0/CPU0:router(config-isis-af)# metric-style wide level 1	Configures a router to generate and accept only wide link metrics in the Level 1 area.
Step 5	router-id loopback loopback interface used for prefix-sid Example: RP/0/RP0/CPU0:router(config-isis-af)# router-id loopback0	Configures router ID for each address-family (IPv4/IPv6). IS-IS advertises the router ID in TLVs 134 (for IPv4 address family) and 140 (for IPv6 address family). Required when traffic engineering is used.
Step 6	segment-routing mpls [sr-prefer] Example: RP/0/RP0/CPU0:router(config-isis-af)# segment-routing mpls	Segment routing is enabled by the following actions: • MPLS forwarding is enabled on all interfaces where IS-IS is active. • All known prefix-SIDs in the forwarding plain are programmed, with the

	Command or Action	Purpose
		prefix-SIDs advertised by remote routers or learned through local or remote mapping server. The prefix-SIDs locally configured are advertised. Use the sr-prefer keyword to set the preference of segment routing (SR) labels over label distribution protocol (LDP) labels.
Step 7	exit Example: RP/0/RP0/CPU0:router(config-isis-af)# exit RP/0/RP0/CPU0:router(config-isis)# exit	
Step 8	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.

What to do next

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.

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

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

Before you begin

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

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	router isis instance-id Example: RP/0/RP0/CPU0:router(config)# router isis 1	Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode. You can change the level of routing to be performed by a particular routing instance by using the is-type router configuration command.
Step 3	interface Loopback instance Example: RP/0/RP0/CPU0:router(config-isis)# interface Loopback0	Specifies the loopback interface and instance.
Step 4	address-family { ipv4 ipv6 } [unicast] Example: The following is an example for ipv4 address family: RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast	Specifies the IPv4 or IPv6 address family, and enters router address family configuration mode.
Step 5	prefix-sid [algorithm algorithm-number] {index SID-index absolute SID-value} [n-flag-clear] [explicit-null] Example: RP/0/RP0/CPU0:router(config-isis-if-af)# prefix-sid index 1001	Configures the prefix-SID index or absolute value for the interface. Specify algorithm algorithm-number to configure SR Flexible Algorithm. Specify index SID-index for each node to create a prefix SID based on the lower boundary of the SRGB + the index.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router(config-isis-if-af)# prefix-sid absolute 17001	Specify absolute SID-value for each node to create a specific prefix SID within the SRGB. By default, the n-flag is set on the prefix-SID, indicating that it is a node SID. For specific prefix-SID (for example, Anycast prefix-SID), enter the n-flag-clear keyword. IS-IS does not set the N flag in the prefix-SID sub Type Length Value (TLV). To disable penultimate-hop-popping (PHP) and add explicit-Null label, enter explicit-null keyword. IS-IS sets the E flag in the prefix-SID sub TLV.
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/RP0/CPU0:router# show isis database verbose
```

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

<...>
Metric: 0             IP-Extended 10.0.0.1/32
  Prefix-SID Index: 1001, Algorithm:0, R:0 N:1 P:0 E:0 V:0 L:0

<...>
```

Weighted Anycast SID-Aware Path Computation

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Weighted Anycast SID-Aware Path Computation	Release 7.3.1	This feature extends Anycast SIDs with weighted nodes. Weighted Anycast nodes advertise a cost (weight) along with the Anycast SID. Traffic is then distributed according to the weights. Weighted Anycast SIDs allow for highly available paths with node redundancy and path optimality that provide Fast ReRoute (FRR) for node failure of service provider edge (PE) routers and ABR/ASBRs nodes in multi-domain networks.

The Weighted Anycast SID feature extends Anycast SIDs with weighted nodes.

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. With the default (unweighted) behavior, the traffic is load-balanced across each node in the group evenly.

Weighted Anycast nodes advertise a cost along with the Anycast SID. This cost serves as a weight. Traffic to the SID is then distributed according to the weights.

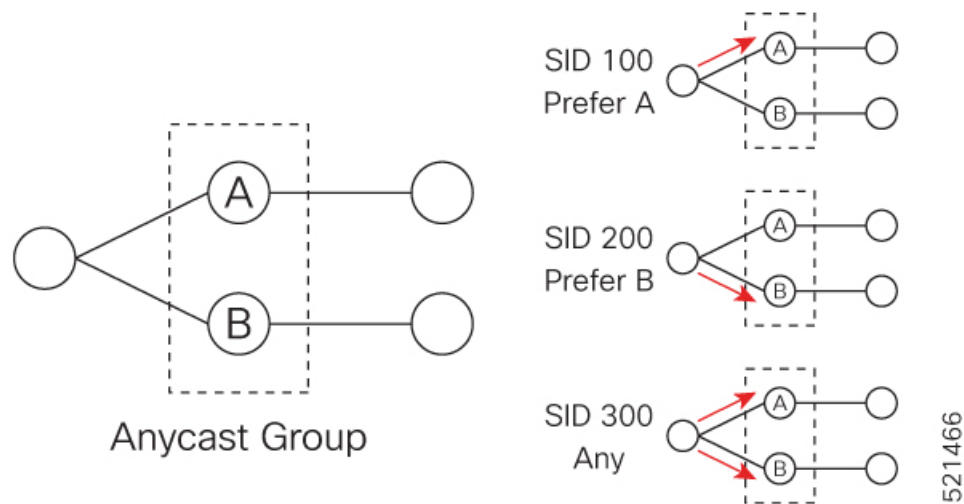
Weighted Anycast SIDs allow for highly available paths with node redundancy and path optimality that provide FRR for node failure of service provider edge (PE) routers and ABR/ASBR nodes in multi-domain networks.

In addition, Weighted Anycast SIDs allow for scaled computation at the PCE of multi-domain paths.

The native SR path computation algorithms are augmented to compute optimum paths relying on Weighted Anycast SIDs during path encoding.

Consider the example depicted below. Nodes A and B are part of the same Anycast groups, represented by different SIDs (100, 200, 300).

- SID 100 sends traffic preferentially to node A
- SID 200 sends traffic preferentially to node B
- SID 300 sends traffic equally to both nodes

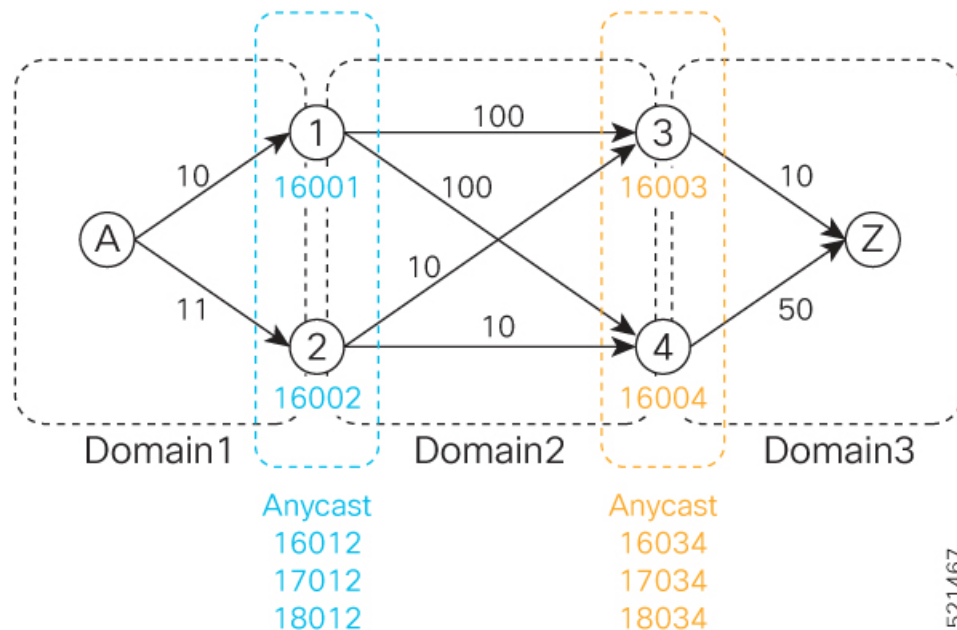


The Anycast replacement algorithm runs after an SR-TE path has been computed. It examines the prefix SIDs in the path and swaps them with Anycast SIDs that contain the same node. The new paths are checked against the original constraints and kept if suitable.

If a node is part of multiple Anycast groups, the algorithm considers them according to their weights.

Example

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. The link delays are also shown.



ABRs 1 and 2 share the following Anycast SIDs:

- 16012 – sends traffic to either Node 1 or 2 (the topologically nearest node)
- 17012 – sends traffic preferentially to Node 1

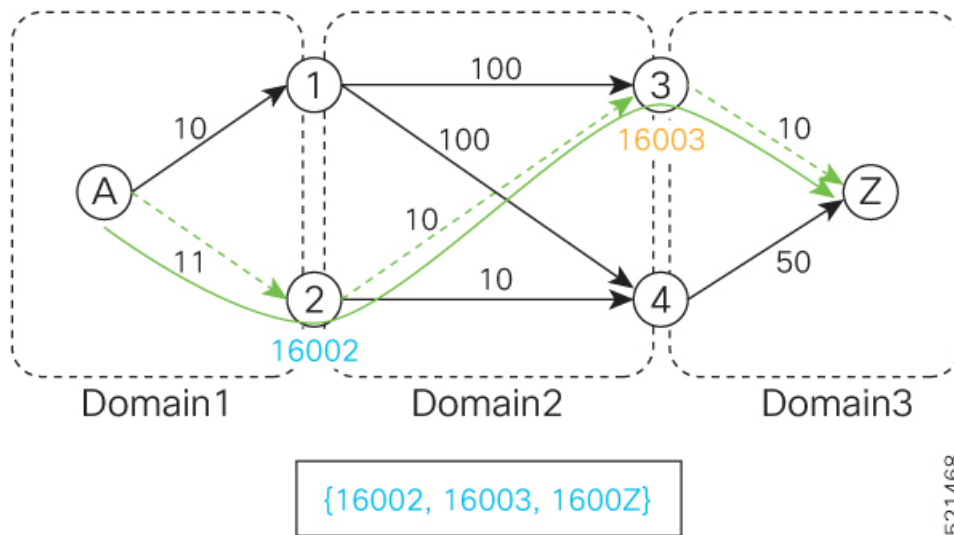
- 18012 – sends traffic preferentially to Node 2

ABRs 3 and 4 share the following Anycast SIDs:

- 16034 – sends traffic either Node 3 or 4 (the topologically nearest node)
- 17034 – sends traffic preferentially to Node 3
- 18034 – sends traffic preferentially to Node 4

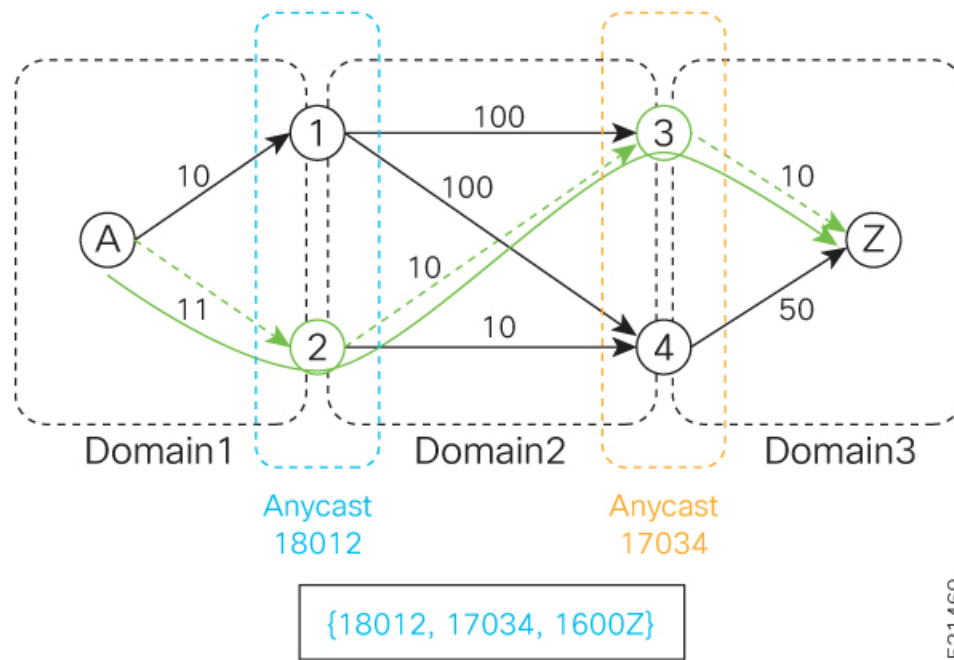
Consider the case where routers A and Z are provider edge (PE) routers in the same VPN. Router A receives a VPN route with BGP next-hop to router Z. Router A resolves the SR path to router Z using SR-ODN or SR-PCE.

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 $16002 > 16003 > 1600Z$.

Using the weighted Anycast-encoded SID list, the optimized computed path from router A to router Z is $18012 > 17034 > 1600Z$. This path has a cumulative delay of 31.



Using node SIDs, failures inside each domain (for example, links) benefit from fast TI-LFA convergence. However, failures of the ABR nodes would be dependent on SR-PCE reoptimization.

Using weighted Anycast SIDs, failures of the ABR nodes and failures inside each domain benefit from fast TI-LFA convergence.

Configuration

Based on the topology in Figure NN, this example shows the Weighted Anycast SID configuration of ABRs 1 and 2.

ABR 1 Configuration

```
RP/0/RSP0/CPU0:ios(config)# router isis 1
RP/0/RSP0/CPU0:ios(config-isis)# interface Loopback0
RP/0/RSP0/CPU0:ios(config-isis-if)# address-family ipv4 unicast
RP/0/RSP0/CPU0:ios(config-isis-if-af)# prefix-sid absolute 16001
RP/0/RSP0/CPU0:ios(config-isis-if-af)# exit
RP/0/RSP0/CPU0:ios(config-isis-if)# exit
RP/0/RSP0/CPU0:ios(config-isis)# interface Loopback1
RP/0/RSP0/CPU0:ios(config-isis-if)# prefix-attributes anycast
RP/0/RSP0/CPU0:ios(config-isis-if)# address-family ipv4 unicast
RP/0/RSP0/CPU0:ios(config-isis-if-af)# prefix-sid absolute 16012
RP/0/RSP0/CPU0:ios(config-isis-if-af)# exit
RP/0/RSP0/CPU0:ios(config-isis-if)# exit
RP/0/RSP0/CPU0:ios(config-isis)# interface Loopback2
RP/0/RSP0/CPU0:ios(config-isis-if)# prefix-attributes anycast
RP/0/RSP0/CPU0:ios(config-isis-if)# address-family ipv4 unicast
RP/0/RSP0/CPU0:ios(config-isis-if-af)# metric 1
RP/0/RSP0/CPU0:ios(config-isis-if-af)# prefix-sid absolute 17012
RP/0/RSP0/CPU0:ios(config-isis-if-af)# exit
RP/0/RSP0/CPU0:ios(config-isis-if)# exit
RP/0/RSP0/CPU0:ios(config-isis)# interface Loopback3
RP/0/RSP0/CPU0:ios(config-isis-if)# prefix-attributes anycast
RP/0/RSP0/CPU0:ios(config-isis-if)# address-family ipv4 unicast
RP/0/RSP0/CPU0:ios(config-isis-if-af)# metric 100000
```

```
RP/0/RSP0/CPU0:ios(config-isis-if-af)# prefix-sid absolute 18012
```

Running Config

```
router isis 1
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid absolute 16001 // Node SID
  !
 !
 interface Loopback1
  prefix-attributes anycast
  address-family ipv4 unicast
   prefix-sid absolute 16012 //Anycast SID - (prefer node 1 or 2)
  !
 !
 interface Loopback2
  prefix-attributes anycast
  address-family ipv4 unicast
   metric 1
   prefix-sid absolute 17012 // Weighted Anycast SID (prefer node 1)
  !
 !
 interface Loopback3
  prefix-attributes anycast
  address-family ipv4 unicast
   metric 100000
   prefix-sid absolute 18012 // Weighted Anycast SID (prefer node 2)
  !
 !
end
```

ABR 2 Configuration

```
RP/0/RSP0/CPU0:ios(config)# router isis 1
RP/0/RSP0/CPU0:ios(config-isis)# interface Loopback0
RP/0/RSP0/CPU0:ios(config-isis-if)# address-family ipv4 unicast
RP/0/RSP0/CPU0:ios(config-isis-if-af)# prefix-sid absolute 16002
RP/0/RSP0/CPU0:ios(config-isis-if-af)# exit
RP/0/RSP0/CPU0:ios(config-isis-if)# exit
RP/0/RSP0/CPU0:ios(config-isis)# interface Loopback1
RP/0/RSP0/CPU0:ios(config-isis-if)# prefix-attributes anycast
RP/0/RSP0/CPU0:ios(config-isis-if)# address-family ipv4 unicast
RP/0/RSP0/CPU0:ios(config-isis-if-af)# prefix-sid absolute 16012
RP/0/RSP0/CPU0:ios(config-isis-if-af)# exit
RP/0/RSP0/CPU0:ios(config-isis-if)# exit
RP/0/RSP0/CPU0:ios(config-isis)# interface Loopback2
RP/0/RSP0/CPU0:ios(config-isis-if)# prefix-attributes anycast
RP/0/RSP0/CPU0:ios(config-isis-if)# address-family ipv4 unicast
RP/0/RSP0/CPU0:ios(config-isis-if-af)# metric 100000
RP/0/RSP0/CPU0:ios(config-isis-if-af)# prefix-sid absolute 17012
RP/0/RSP0/CPU0:ios(config-isis-if-af)# exit
RP/0/RSP0/CPU0:ios(config-isis-if)# exit
RP/0/RSP0/CPU0:ios(config-isis)# interface Loopback3
RP/0/RSP0/CPU0:ios(config-isis-if)# prefix-attributes anycast
RP/0/RSP0/CPU0:ios(config-isis-if)# address-family ipv4 unicast
RP/0/RSP0/CPU0:ios(config-isis-if-af)# metric 1
RP/0/RSP0/CPU0:ios(config-isis-if-af)# prefix-sid absolute 18012
```

Running Config

```

router isis 1
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid absolute 16002 // Node SID
  !
 !
 interface Loopback1
  prefix-attributes anycast
  address-family ipv4 unicast
   prefix-sid absolute 16012 // Anycast SID (prefer any)
  !
 !
 interface Loopback2
  prefix-attributes anycast
  address-family ipv4 unicast
   metric 100000
   prefix-sid absolute 17012 // Weighted Anycast SID (prefer node 1)
  !
 !
 interface Loopback3
  prefix-attributes anycast
  address-family ipv4 unicast
   metric 1
   prefix-sid absolute 18012 // Weighted Anycast SID (prefer node 2)
  !
 !
end

```

Configuring an Adjacency SID

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

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

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

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

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

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

0 1 2 3 4 5 6 7

```

+---+---+---+---+---+---+
| F | B | V | L | S | P |   |
+---+---+---+---+---+---+

```

Table 2: Adjacency Segment Identifier (Adj-SID) Flags Sub-TLV Fields

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

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

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

Before you begin

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

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

Procedure

	Command or Action	Purpose
Step 1	configure Example: <code>RP/0/RP0/CPU0:router# configure</code>	Enters mode.
Step 2	router isis instance-id Example: <code>RP/0/RP0/CPU0:router(config)# router isis 1</code>	Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode. You can change the level of routing to be performed by a particular routing instance by using the is-type router configuration command.
Step 3	interface type interface-path-id Example: <code>RP/0/RP0/CPU0:router(config-isis)# interface GigabitEthernet0/0/0/7</code>	Specifies the interface and enters interface configuration mode.
Step 4	point-to-point Example: <code>RP/0/RP0/CPU0:router(config-isis-if)# point-to-point</code>	Specifies the interface is a point-to-point interface.

	Command or Action	Purpose
Step 5	address-family { ipv4 ipv6 } [unicast] Example: The following is an example for ipv4 address family: <pre>RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast</pre>	Specifies the IPv4 or IPv6 address family, and enters router address family configuration mode.
Step 6	adjacency-sid {index adj-SID-index absolute adj-SID-value } [protected] Example: <pre>RP/0/RP0/CPU0:router(config-isis-if-af)# adjacency-sid index 10</pre> <pre>RP/0/RP0/CPU0:router(config-isis-if-af)# adjacency-sid absolute 15010</pre>	Configures the Adj-SID index or absolute value for the interface. Specify index adj-SID-index for each link to create an Adj-SID based on the lower boundary of the SRLB + the index. Specify absolute adj-SID-value for each link to create a specific Adj-SID within the SRLB. Specify if the Adj-SID is protected . For each primary path, if the Adj-SID is protected on the primary interface and a backup path is available, a backup path is installed. By default, manual Adj-SIDs are not protected.
Step 7	Use the commit or end command.	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:

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

IS-IS 1 Manual Adjacency SID Table

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

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

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

```
RP/0/RP0/CPU0:router# show mpls forwarding labels 15010
Mon Jun 12 02:50:12.172 PDT
Local   Outgoing   Prefix      Outgoing    Next Hop    Bytes
Label   Label      or ID       Interface   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          (!)
```

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 3 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 3 bundle link goes up and down.
- Manually configured Layer 2 Adj-SIDs are persistent if the Layer 3 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.

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	segment-routing Example: RP/0/RP0/CPU0:Router(config)# segment-routing	Enters segment routing configuration mode.
Step 3	adjacency-sid Example: RP/0/RP0/CPU0:Router(config-sr)# adjacency-sid	Enters adjacency SID configuration mode.
Step 4	interface type interface-path-id Example: RP/0/RP0/CPU0:Router(config-sr-adj)# interface GigabitEthernet0/0/0/3	Specifies the interface and enters interface configuration mode.
Step 5	address-family { ipv4 ipv6 } [unicast] Example: RP/0/RP0/CPU0:Router(config-sr-adj-intf)# address-family ipv4 unicast	Specifies the IPv4 or IPv6 address family, and enters router address family configuration mode.
Step 6	l2-adjacency sid {index adj-SID-index absolute adj-SID-value } [next-hop {ipv4_address ipv6_address }] Example: RP/0/RP0/CPU0:Router(config-sr-adj-intf-af)# l2-adjacency sid absolute 15015 next-hop 10.1.1.4	Configures the Adj-SID index or absolute value for the interface. Specify index adj-SID-index for each link to create an Adj-SID based on the lower boundary of the SRLB + the index. Specify absolute adj-SID-value for each link to create a specific Adj-SID within the SRLB. 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.

	Command or Action	Purpose
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	end	
Step 9	router isis <i>instance-id</i> Example: RP/0/RP0/CPU0:Router(config) # router isis isp	Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.
Step 10	address-family { ipv4 ipv6 } [unicast] Example: RP/0/RP0/CPU0:Router(config-isis) # address-family ipv4 unicast	Specifies the IPv4 or IPv6 address family, and enters router address family configuration mode.
Step 11	segment-routing bundle-member-adj-sid Example: RP/0/RP0/CPU0:Router(config-isis-af) # segment-routing bundle-member-adj-sid	Programs the dynamic Layer 2 Adj-SIDs, and advertises both manual and dynamic Layer 2 Adj-SIDs. Note This command is not required to program manual L2 Adj-SID, but is required to program the dynamic Layer 2 Adj-SIDs and to advertise both manual and dynamic Layer 2 Adj-SIDs.

Verify the configuration:

```
Router# show mpls forwarding detail | i "Pop|Outgoing Interface|Physical Interface"
Tue Jun 20 06:53:51.876 PDT
. . .
15001  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 15015 next-hop 10.1.1.4
!
!
!
!

```

Configuring Bandwidth-Based Local UCMP

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

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

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

Before you begin

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	router isis <i>instance-id</i> Example: RP/0/RP0/CPU0:router(config)# router isis 1	Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode. You can change the level of routing to be performed by a particular routing instance by using the is-type router configuration command.
Step 3	address-family { ipv4 ipv6 } [unicast] Example: The following is an example for ipv4 address family:	Specifies the IPv4 or IPv6 address family, and enters IS-IS address family configuration mode.

	Command or Action	Purpose
	<code>RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast</code>	
Step 4	apply-weight ecmp-only bandwidth Example: <code>RP/0/RP0/CPU0:router(config-isis-af)# apply-weight ecmp-only bandwidth</code>	Enables UCMP functionality locally between ECMP paths based on the bandwidth of the local links.
Step 5	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session. 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.

Conditional Prefix Advertisement

In some situations, it's beneficial to make the IS-IS prefix advertisement conditional. For example, an Area Border Router (ABR) or Autonomous System Boundary Router (ASBR) that has lost its connection to one of the areas or autonomous systems (AS) might keep advertising a prefix. If an ABR or ASBR advertises the Segment Routing (SR) SID with this prefix, the label stack of the traffic routed toward the disconnected area or AS might use this SID, which would result in dropped traffic at the ABR or ASBR.

ABRs or ASBRs are often deployed in pairs for redundancy and advertise a shared Anycast prefix SID. Conditional Prefix Advertisement allows an ABR or an ASBR to advertise its Anycast SID only when connected to a specific area or domain. If an ABR or ASBR becomes disconnected from the particular area or AS, it stops advertising the address for a specified interface (for example, Loopback).

Configure the conditional prefix advertisement under a specific interface. The prefix advertisement on this interface is associated with the route-policy that tracks the presence of a set of prefixes (prefix-set) in the Routing Information Base (RIB).

For faster convergence, the route-policy used for conditional prefix advertisement uses the new event-based **rib-has-route async** condition to notify IS-IS of the following situations:

- When the last prefix from the prefix-set is removed from the RIB.
- When the first prefix from the prefix-set is added to the RIB.

Configuration

To use the conditional prefix advertisement in IS-IS, create a prefix-set to be tracked. Then create a route policy that uses the prefix-set.

```
Router(config)# prefix-set prefix-set-name
Router(config-pfx)# prefix-address-1/length[, prefix-address-2/length,,
prefix-address-16/length]
Router(config-pfx)# end-set

Router(config)# route-policy rpl-name
Router(config-rpl)# if rib-has-route async prefix-set-name then
Router(config-rpl-if)# pass
Router(config-rpl-if)# endif
Router(config-rpl)# end-policy
```

To advertise the loopback address in IS-IS conditionally, use the **advertise prefix route-policy** command under IS-IS interface address-family configuration sub-mode.

```
Router(config)# router isis 1
Router(config-isis)# interface Loopback0
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# advertise prefix route-policy rpl-name
Router(config-isis-if-af)# commit
```

Example

```
Router(config)# prefix-set domain_2
Router(config-pfx)# 2.3.3.3/32, 2.4.4.4/32
Router(config-pfx)# end-set
Router(config)# route-policy track_domain_2
Router(config-rpl)# if rib-has-route async domain_2 then
Router(config-rpl-if)# pass
Router(config-rpl-if)# endif
Router(config-rpl)# end-policy
Router(config)# router isis 1
Router(config-isis)# interface Loopback0
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# advertise prefix route-policy track_domain-2
Router(config-isis-if-af)# commit
```

Running Configuration

```
prefix-set domain_2
  2.3.3.3/32,
  2.4.4.4/32
end-set
!
route-policy track_domain_2
  if rib-has-route async domain_2 then
    pass
  endif
end-policy
!
router isis 1
  interface Loopback0
    address-family ipv4 unicast
      advertise prefix route-policy track_domain_2
    !
  !
```

Segment Routing ECMP-FEC Optimization

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
SR ECMP-FEC Optimization L2 and L3 Recursive Services	Release 7.4.1	This feature adds support for L2VPN service Label Edge Router (LER) and BGP PIC for Layer 3 BGP services when SR ECMP-FEC Optimization is enabled.

ECMP-FECs are used for any ECMP programming on the system, such as MPLS LSP ECMP, VPN multipath, and EVPN multi-homing.

The SR ECMP-FEC optimization solution minimizes ECMP-FEC resource consumption and duplication, during underlay programming for an SR-MPLS network. This feature supports sharing the same ECMP-FEC, regular FEC, and Egress Encapsulation DB (EEDB) entries among Segment Routing prefixes with the same set of next hops.

ECMP-FEC optimization is triggered when all the out_labels associated with the ECMP paths for a given prefix have the same value. If this rule is not met, then the prefix is programmed with a dedicated ECMP-FEC.

Segment Routing Label Edge Router (LER) ECMP-FEC Optimization enables ECMP-FEC optimization originally developed for Label Switched Router (LSR) nodes (MPLS P) to be enabled on LER (Layer 3 MPLS PE) routers.

Usage Guidelines and Limitations

- For the labeled prefixes with ECMP across a combination of labeled and unlabeled (PHP) paths, the SR ECMP-FEC Optimization cannot be triggered since the paths associated with the prefix do not have the same outgoing label and/or label action.
- For prefixes with LFA backup paths, the SR ECMP-FEC Optimization is possible since these backup paths do not require an extra label to be pushed; all paths associated with the prefix (primary and backup) have the same outgoing label value.
- For prefixes with TI-LFA backup paths requiring extra labels to be pushed on to the backup, the SR ECMP-FEC Optimization is not possible since all the paths associated with the prefix do not have the same outgoing label value.
- For the duration of time that prefixes are programmed to avoid microloops (when SR MicroLoop Avoidance is triggered), SR ECMP-FEC Optimization is not possible since all the paths associated with the prefix do not have the same outgoing label value. After removal of the microloop-avoidance programming, the SR ECMP-FEC Optimization might be possible again.
- For scenarios with prefixes where the SR ECMP-FEC Optimization is not possible, dedicated ECMP-FEC is allocated per prefix. This could potentially lead to ECMP FEC out-of-resource (OOR) considering the baseline usage of ECMP FEC resources at steady state. During ECMP-FEC OOR, prefixes with multiple paths are programmed with a single path in order to avoid traffic disruption.

- SR ECMP-FEC optimization is applicable in the following instances:
 - Label Switched Router (LSR) nodes (MPLS P)
 - L3VPN Label Edge Router (LER) nodes
 - L2VPN LER nodes
 - ASBR node with BGP-LU swap
- BGP PIC is supported
- SR ECMP-FEC optimization should not be enabled in the following instances:
 - L2VPN/L3VPN LER nodes with VPN over BGP-LU over SR
- For the labeled prefixes, transitioning from TI-LFA to SR ECMP-FEC optimization can cause ECMP-FEC OOR due to different output labels (ECMP label vs backup path's label) at make-before-break. This results in a few second traffic loss depending on route scale

Enable SR ECMP-FEC Optimization

To enable SR ECMP-FEC optimization, use the **hw-module fib mpls label lsr-optimized** command in global configuration mode. After enabling this feature, reload the line card. For more information about the command, see the *MPLS Label Distribution Protocol Commands* chapter in the *MPLS Command Reference Guide*.

```
Router(config)# hw-module fib mpls label lsr-optimized
Router(config)# commit

LC/0/0/CPU0:Oct 11 20:19:12.540 UTC: fia_driver[185]:
%FABRIC-FIA_DRV-4-MPLS_HW_PROFILE_MISMATCH :
Mismatch found, reload LC to activate the new mpls profile

Router# reload location 0/0/CPU0

Proceed with reload? [confirm]
Reloading node 0/0/CPU0
```

Verify the SR ECMP-FEC Optimization

From Cisco IOS XR Release 24.2.1, in addition to IPv4 prefixes, you can view the ECMP-FEC optimization details for IPv6 prefixes using the **show controllers npu resources** command. In earlier releases, ECMP-FEC optimization for IPv4 prefixes was available.

The following example shows NPU usage before enabling SR ECMP-FEC optimization for IPv4 and IPv6 prefixes:

```
Router#show controllers npu resources ecmpfec location 0/0/CPU0
Tue May 14 17:17:40.306 UTC
HW Resource Information
  Name                               : ecmp_fec
  Asic Type                           : JerichoPlus

NPU-0
  OOR Summary
    Estimated Max Entries             : 4096
    Red Threshold                     : 95 %
    Yellow Threshold                   : 80 %
```

```

        OOR State                : Green
        High Water Mark          : 216
        High Water Mark Time     : 2024.May.14 17:10:46 UTC
        Bank Info                 : ECMP

OFA Table Information
(May not match HW usage)
    ipnhgroup                    : 111
    ip6nhgroup                   : 105

Current Hardware Usage
Name: ecmp_fec
    Estimated Max Entries        : 4096
    Total In-Use                : 216          (5 %)
    OOR State                    : Green
    High Water Mark              : 216
    High Water Mark Time        : 2024.May.14 17:10:46 UTC
    Bank Info                    : ECMP

Name: hier_0
    Estimated Max Entries        : 4096
    Total In-Use                : 216
    OOR State                    : Green
    High Water Mark              : 216
    High Water Mark Time        : 2024.May.14 17:10:46 UTC
    Bank Info                    : ECMP

```

The following example shows NPU usage after enabling SR ECMP-FEC optimization for IPv4 and IPv6 prefixes. After enabling ECMP-FEC optimization, the overall hardware usage has dropped to 16%, and the group of next-hop addresses used for forwarding IPv4 and IPv6 is significantly reduced.

```
Router#show controllers npu resources ecmpfec location 0/0/CPU0
```

```
Tue May 14 17:07:56.266 UTC
```

```
HW Resource Information
```

```

    Name                : ecmp_fec
    Asic Type            : JerichoPlus

```

```
NPU-0
```

```
OOB Summary
```

```

    Estimated Max Entries    : 4096
    Red Threshold            : 95 %
    Yellow Threshold         : 80 %
    OOR State                : Green
    High Water Mark          : 16
    High Water Mark Time    : 2024.May.14 17:04:47 UTC
    Bank Info                : ECMP

```

```
OFA Table Information
```

```
(May not match HW usage)
```

```

    ipnhgroup              : 11
    ip6nhgroup             : 5

```

```
Current Hardware Usage
```

```

Name: ecmp_fec
    Estimated Max Entries    : 4096
    Total In-Use                : 16          (0 %)
    OOR State                : Green
    High Water Mark          : 16
    High Water Mark Time    : 2024.May.14 17:04:47 UTC
    Bank Info                : ECMP

```

```
Name: hier_0
  Estimated Max Entries      : 4096
  Total In-Use              : 16
  OOR State                 : Green
  High Water Mark           : 16
  High Water Mark Time      : 2024.May.14 17:04:47 UTC
  Bank Info                 : ECMP
```

