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.

- Enabling Segment Routing for IS-IS Protocol, on page 1
- Configuring a Prefix-SID on the IS-IS Enabled Loopback Interface, on page 4
- Configuring an Adjacency SID, on page 6
- Configuring Bandwidth-Based Local UCMP, on page 12
- IS-IS Prefix Attributes for Extended IPv4 and IPv6 Reachability, on page 13
- IS-IS Multi-Domain Prefix SID and Domain Stitching: Example, on page 16

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. **commit**

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong></td>
</tr>
</tbody>
</table>
| **Step 2** | **router isis instance-id**  
Example:  
RP/0/RSP0/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/RSP0/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/RSP0/CPU0:router(config-isis-af)# metric-style wide level 1  
| Configures a router to generate and accept only wide link metrics in the Level 1 area. |
| **Step 5** | **mpls traffic-eng level**  
Example:  
<p>| Enables RSVP traffic engineering functionality. |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis-af)# mpls traffic-eng level-2-only</td>
<td>Sets the traffic engineering loopback interface.</td>
</tr>
<tr>
<td><strong>Step 6</strong> mpls traffic-eng router-id interface</td>
<td>Configures router ID for each address-family (ipv4/ipv6).</td>
</tr>
</tbody>
</table>
| **Example:**
RP/0/RSP0/CPU0:router(config-isis-af)# mpls traffic-eng router-id Loopback0 | |
| **Step 7** router-id loopback loopback interface used for prefix-sid | Segment routing is enabled by the following actions: |
| **Example:**
RP/0/(config-isis-af)#router-id loopback0 | • MPLS forwarding is enabled on all interfaces where IS-IS is active. |
| | • All known prefix-SIDs in the forwarding plain are programmed, with the prefix-SIDs advertised by remote routers or learned through local or remote mapping server. |
| | • The prefix-SIDs locally configured are advertised. |
| **Step 8** segment-routing mpls | 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. |
| **Example:**
RP/0/RSP0/CPU0:router(config-isis-af)# segment-routing mpls | |
| **Step 9** exit | What to do next |
| **Example:**
RP/0/RSP0/CPU0:router(config-isis-af)# exit | Configure the prefix SID. |
| RP/0/RSP0/CPU0:router(config-isis)# exit | |
| **Step 10** mpls traffic-eng | |
| **Example:**
RP/0/RSP0/CPU0:router(config)# mpls traffic-eng | |
| **Step 11** commit | |

**What to do next**

Configure the prefix SID.
Configuring a Prefix-SID on the IS-IS Enabled Loopback Interface

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

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 SR Router 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.

Note

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

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. commit

DETAILED STEPS

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

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.
Configure Segment Routing for IS-IS Protocol

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>interface Loopback instance</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RSP0/CPU0:router(config-isis)# interface Loopback0</td>
<td>Specifies the loopback interface and instance.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>**address-family { ipv4</td>
<td>ipv6 } [ unicast ]**&lt;br&gt;<strong>Example:</strong>&lt;br&gt;The following is an example for ipv4 address family:&lt;br&gt;RP/0/RSP0/CPU0:router(config-isis-if)# address-family ipv4 unicast</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>**prefix-sid [strict-spf</td>
<td>algorithm algorithm-number] [index SID-index</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>commit</strong></td>
<td>Verify the prefix-SID configuration:&lt;br&gt;RP/0/RSP0/CPU0:router# show isis database verbose&lt;br&gt;IS-IS 1 (Level-2) Link State Database&lt;br&gt;LSPID LSP Seq Num LSP Checksum LSP Holdtime ATT/P/OL&lt;br&gt;router.00-00 * 0x0000039b 0xfc27 1079 0/0/0</td>
</tr>
</tbody>
</table>
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.
Table 1: Adjacency Segment Identifier (Adj-SID) Flags Sub-TLV Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S (Set)</td>
<td>This flag is set if the same Adj-SID value has been provisioned on multiple interfaces.</td>
</tr>
<tr>
<td>P (Persistent)</td>
<td>This flag is set if the Adj-SID is persistent (manually allocated).</td>
</tr>
</tbody>
</table>

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. `commit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure</code></td>
<td>Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.</td>
</tr>
</tbody>
</table>
| Step 2 | `router isis instance-id`  
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 type interface-path-id`  
Example:  
`RP/0/RSP0/CPU0:router(config-isis)# interface GigabitEthernet0/0/7` | Specifies the interface and enters interface configuration mode. |
| Step 4 | `point-to-point`  
Example: | Specifies the interface is a point-to-point interface. |
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>RP/0/RSP0/CPU0:router(config-isis-if)# point-to-point</td>
<td>Specifies the IPv4 or IPv6 address family, and enters router address family configuration mode.</td>
</tr>
</tbody>
</table>

**Step 5**

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

**Step 6**

adjacency-sid { index adj-SID-index | absolute adj-SID-value } [ protected ]

**Example:**
RP/0/RSP0/CPU0:router(config-isis-if-af)# adjacency-sid index 10

RP/0/RSP0/CPU0:router(config-isis-if-af)# adjacency-sid absolute 15010

**Step 7**

commit

Verify the Adj-SID configuration:

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):

RP/0/RSP0/CPU0:router# show mpls forwarding labels 15010
Mon Jun 12 02:50:12.172 PDT

<table>
<thead>
<tr>
<th>Local</th>
<th>Outgoing</th>
<th>Prefix</th>
<th>Outgoing</th>
<th>Next Hop</th>
<th>Bytes</th>
<th>Switched</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Label or ID</td>
<td>Interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>-------</td>
<td>----------</td>
</tr>
<tr>
<td>15010</td>
<td>Pop</td>
<td>SRLB (idx 10)</td>
<td>G10/0/0/3</td>
<td>10.0.3.3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Pop</td>
<td>SRLB (idx 10)</td>
<td>G10/0/0/7</td>
<td>10.1.0.5</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16004</td>
<td>Pop</td>
<td>SRLB (idx 10)</td>
<td>G10/0/0/7</td>
<td>10.1.0.5</td>
<td>0</td>
<td>(!)</td>
</tr>
</tbody>
</table>
What to do next
Configure the SR-TE policy.

Configuring 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 ]
### Configuring a Layer 2 Adjacency SID

1. **l2-adjacency sid** `{index adj-SID-index | absolute adj-SID-value} [next-hop {ipv4_address | ipv6_address}]` 
2. **commit** 
3. **end** 
4. **router isis instance-id** 
5. **address-family {ipv4 | ipv6} [unicast]** 
6. **segment-routing bundle-member-adj-sid**

#### 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 segment routing configuration mode.</td>
</tr>
</tbody>
</table>
| **Step 2** segment-routing  
  Example:  
  Router(config)# segment-routing | Enters adjacency SID configuration mode. |
| **Step 3** adjacency-sid  
  Example:  
  Router(config-sr)# adjacency-sid | Specifies the interface and enters interface configuration mode. |
| **Step 4** interface `type interface-path-id`  
  Example:  
  Router(config-sr-adj)# interface GigabitEthernet0/0/0/3 | Specifies the IPv4 or IPv6 address family, and enters router address family configuration mode. |
| **Step 5** address-family `{ipv4 | ipv6} [unicast]`  
  Example:  
  Router(config-sr-adj-intf)# address-family ipv4 unicast | Configures the Adj-SID index or absolute value for the interface. |
| **Step 6** l2-adjacency sid `{index adj-SID-index | absolute adj-SID-value} [next-hop {ipv4_address | ipv6_address}]`  
  Example:  
  Router(config-sr-adj-intf-af)# l2-adjacency sid absolute 15015 next-hop 10.1.1.4 | Specify `index adj-SID-index` for each link to create an Adj-SID based on the lower boundary of the SRLB + the index. |
<p>| | Specify <code>absolute adj-SID-value</code> 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. |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>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>
<td></td>
</tr>
</tbody>
</table>

**Step 7**

**commit**

**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
```

**Purpose**

Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.

Specifies the IPv4 or IPv6 address family, and enters router address family configuration mode.

Programs the dynamic Layer 2 Adj-SIDs, and advertises both manual and dynamic Layer 2 Adj-SIDs.

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
  l2-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. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enables IS-IS routing for the specified routing instance, and places the router in router 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>Example:</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>RP/0/RSP0/CPU0:router(config)# router isis 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> address-family { ipv4</td>
<td>ipv6 } [ unicast ]</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>The following is an example for ipv4 address family:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis)# address-family ipv4 unicast</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> apply-weight ecmp-only bandwidth</td>
<td>Enables UCMP functionality locally between ECMP paths based on the bandwidth of the local links.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis-af)# apply-weight ecmp-only bandwidth</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>
</tbody>
</table>
N (Node Flag)
For prefixes that are propagated from another level:
1. Copy the N-flag from the prefix attribute sub-TLV, if present in the source level.
2. Copy the N-flag from the prefix-SID sub-TLV, if present in the source level.
3. Otherwise, set to 0.

For connected prefixes:
1. Set to 0 if `isis prefix-attributes n-flag-clear` is configured (see Configuring Prefix Attribute N-flag-clear).
2. Set to 0 if `prefix-SID n-flag-clear` is configured (see Configuring a Prefix-SID on the IS-IS Enabled Loopback Interface).
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.

Note: If the flag is set and the prefix length is not a host prefix, then the flag must be ignored.

### 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 2: Source Router Sub-TLV Format

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

<table>
<thead>
<tr>
<th>IPv6 Source Router ID</th>
<th>Type: 12</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Length: 16</td>
</tr>
<tr>
<td></td>
<td>Value: IPv6 Router ID of the source of the prefix advertisement</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. interface Loopback instance
3. isis prefix-attributes n-flag-clear [Level-1 | Level-2]
4. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 interface Loopback instance</td>
<td>Specifies the loopback interface and instance.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# interface Loopback0</td>
<td></td>
</tr>
<tr>
<td>Step 3 isis prefix-attributes n-flag-clear [Level-1</td>
<td>Level-2]</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-if)# isis prefix-attributes n-flag-clear</td>
<td></td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
</tbody>
</table>

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
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
```
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 IOS XR Traffic Controller (XTC)).

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**
```bash
router isis Domain1
  distribute link-state id
```

**Example: Node 14**
```bash
router isis Domain2
  distribute link-state 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.