Configure Segment Routing for OSPF Protocol

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
NC57 line cards in Native Mode: SR-MPLS - OSPFv2	Release 7.3.1	This feature is now supported on routers that have the Cisco NC57 line cards installed and operate in native mode. To enable the native mode, use the **hw-module profile npu native-mode-enable** command in the configuration mode. Ensure that you reload the router after configuring the native mode.

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

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

Note For additional information on implementing OSPF on your Cisco NCS 5500 Series Router, see the *Implementing OSPF* module in the *Routing Configuration Guide for Cisco NCS 5500 Series Routers*.

- Enabling Segment Routing for OSPF Protocol, on page 1
- Configuring a Prefix-SID on the OSPF-Enabled Loopback Interface, on page 4
- Configuring an Adjacency SID, on page 6
- Conditional Prefix Advertisement, on page 8
- Segment Routing ECMP-FEC Optimization, on page 10

Enabling Segment Routing for OSPF Protocol

Segment routing on the OSPF control plane supports the following:

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

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

Before you begin

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

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

SUMMARY STEPS

1. **configure**
2. **router ospf** *process-name*
3. **segment-routing mpls**
4. **segment-routing sr-prefer**
5. **area** *area*
6. **segment-routing mpls**
7. **exit**
8. Use the **commit** or **end** command.

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**segment-routing mpls** **Example:** `RP/0/RP0/CPU0:router(config-ospf)# segment-routing mpls`	Enables segment routing using the MPLS data plane on the routing process and all areas and interfaces in the routing process. Enables segment routing fowarding on all interfaces in the routing process and installs the SIDs received by OSPF in the forwarding table.
Step 4	**segment-routing sr-prefer** **Example:** `RP/0/RP0/CPU0:router(config-ospf)# segment-routing sr-prefer`	Sets the preference of segment routing (SR) labels over label distribution protocol (LDP) labels.
Step 5	**area** *area* **Example:** `RP/0/RP0/CPU0:router(config-ospf)# area 0`	Enters area configuration mode.
Step 6	**segment-routing mpls** **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar)# segment-routing mpls`	(Optional) Enables segment routing using the MPLS data plane on the area and all interfaces in the area. Enables segment routing fowarding on all interfaces in the area and installs the SIDs received by OSPF in the forwarding table.

Step 7	**exit**
	Example:
	`RP/0/RP0/CPU0:router(config-ospf-ar)# exit`
	`RP/0/RP0/CPU0:router(config-ospf)# 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 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** [**algorithm** *algorithm-number*] {**index** *SID-index* | **absolute** *SID-value* } [**n-flag-clear**] [**explicit-null**]
6. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**router ospf** *process-name* **Example:** `RP/0/RP0/CPU0:router(config)# router ospf 1`	Enables OSPF routing for the specified routing process, and places the router in router configuration mode.
Step 3	**area** *value* **Example:**	Enters area configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-ospf)# area 0`	
Step 4	**interface Loopback** *interface-instance* **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar)# interface Loopback0 passive`	Specifies the loopback interface and instance.

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-ospf-ar)# prefix-sid index 1001`
	`RP/0/RP0/CPU0:router(config-ospf-ar)# prefix-sid absolute 17001`
		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.
		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. OSPF does not set the `N` flag in the prefix-SID sub Type Length Value (TLV).
		To disable penultimate-hop-popping (PHP) and add an explicit-Null label, enter the `explicit-null` keyword. OSPF 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 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
```

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 OSPF Adj-SID sub-TLV. The P-flag is defined for manually allocated Adj-SIDs.

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|B|V|L|G|P|     |
+-+-+-+-+-+-+-+-+
```

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

Field	Description
P (Persistent)	This flag is set if the Adj-SID is persistent (manually allocated).

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 ospf** *process-name*
3. **area** *area*
4. **interface** *type interface-path-id*
5. **adjacency-sid** {**index** *adj-SID-index* | **absolute** *adj-SID-value*} [**protected**]
6. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**router ospf** *process-name* **Example:** `RP/0/RP0/CPU0:router(config)# router ospf 1`	Enables OSPF routing for the specified routing instance, and places the router in router configuration mode.
Step 3	**area** *area* **Example:** `RP/0/RP0/CPU0:router(config-ospf)# area 0`	Enters area configuration mode.
Step 4	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar)# interface HundredGigE0/0/0/1`	Specifies the interface and enters interface configuration mode.

Step 5	**adjacency-sid** {**index** *adj-SID-index* | **absolute** *adj-SID-value*} [**protected**]
	Example:
	`RP/0/RP0/CPU0:router(config-config-ospf-ar-if)# adjacency-sid index 10`
	`RP/0/RP0/CPU0:router(config-config-ospf-ar-if)# adjacency-sid absolute 15010`
		Configures the Adj-SID index or absolute value for the interface.
		Specify **index** *adj-SID-index* for each link to create an Ajd-SID based on the lower boundary of the SRLB + the index.
		Specify **absolute** *adj-SID-value* for each link to create a specific Ajd-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 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session.

	Command or Action	Purpose
		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 SR-TE policy.

Conditional Prefix Advertisement

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
Segment Routing Conditional Prefix Advertisement for OSPF	Release 7.3.1	In a typical Anycast scenario, if an advertising node becomes unavailable or unreachable while still advertising its Anycast SID, traffic could still be routed to the node and, as a result, get dropped. This feature allows a node to advertise its loopback address when it's connected to the domain, and to track the loopback addresses of the other nodes in the domain. If a node becomes unavailable or unreachable, it stops advertising its loopback address, allowing for a new path to be computed.

In some situations, it's beneficial to make the OSPF 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 OSPF 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 OSPF, 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 OSPF conditionally, use the **advertise prefix route-policy** command under OSPF interface address-family configuration sub-mode.

```
Router(config)# router ospf 1
Router(config-ospf)# area 0
Router(config-ospf-ar)# interface Loopback0
Router(config-ospf-ar-if)# advertise prefix route-policy rpl-name
Router(config-ospf-ar-if)# 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 ospf 1
Router(config-ospf)# area 0
Router(config-ospf-ar)# interface Loopback0
Router(config-ospf-ar-if)# advertise prefix route-policy track_domain-2
Router(config-ospf-ar-if)# 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 ospf 1
 area 0
  interface Loopback0
   advertise prefix route-policy track_domain_2
  !
 !
!
```

Segment Routing ECMP-FEC Optimization

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 during underlay programming for an SR-MPLS network. This feature supports sharing the same ECMP-FEC, regular FEC, and Egress Encapsulation DB (EEDB) entries for all IPv4 and IPv6 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. Other prefixes that meet the rule are candidates for optimization.

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.

For usage guidelines, limitations, and configuration options, see Segment Routing ECMP-FEC Optimization.