OSPF Configuration Guide

First Published: 2025-09-15

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

Read Me First

Read Me First

Only supported features are documented. To confirm or clarify all the supported features for a platform, go to Cisco Feature Navigator.

CONTENTS

CHAPTER 1

OSPF

Feature History for OSPF

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPF: Open Shortest Path First (OSPF) is a link-state routing protocol that is used in Internet Protocol (IP) networks and suitable to be deployed on single autonomous system (AS), such as an enterprise network.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

OSPF

Open Shortest Path First (OSPF) is a link-state routing protocol

- that is used in Internet Protocol (IP) networks,
- is classified as an Interior Gateway Protocol (IGP), and
- suitable to be deployed on single autonomous system (AS), such as an enterprise network.

OSPF versions

OSPF versions and their classifications are as follows:

OSPFv2

OSPF or OSPF Version 2 (OSPFv2) is designed exclusively for IPv4 networks. It exchanges and calculates routes for IPv4 addresses.

OSPFv3

OSPF Version 3 (OSPFv3) was designed exclusively for IPv6 networks and address-family independent routing. While its primary design was for IPv6, OSPFv3 now allows carrying IPv4 routing information along with address families. This means a single OSPFv3 instance can route both IPv4 and IPv6 information.

OSPF for routed access

OSPF for Routed Access refers to a network design in which OSPF is implemented at the access layer of a network using Layer 3 switches. In a routed access design, the Layer 2 functionality, such as VLANs and MAC learning remains active. Layer 3 routing capabilities are deployed on the access layer switches. As a result, Layer 3 switches are configured with IP addresses on their Switched Virtual Interfaces (SVIs) or routed ports and they participate in a routing protocol such as OSPF.

Key components of OSPF

The following are the key components of OSPF:

- Router ID:

  A unique identifier assigned to each OSPF device in the network.

- Areas:

  OSPF divides a large AS into smaller, manageable segments called areas. This limits the scope of LSA flooding so that changes in one area do not necessarily affect the entire AS.

  Each area has a unique Area ID, typically represented as a decimal number, for example 0, 1, 2 or an IP address, for example 1.1.1.1.

  Table lists the types of areas and their classification.

Types of areas	Classification
Backbone Area (Area 0):	This is the central and most critical area in an OSPF domain. All other non-backbone areas must connect directly to Area 0. This ensures a logical and physical path for inter-area communication. It also acts as the transit area for traffic moving between different non-backbone areas.

Types of areas	Classification
Stub Areas	A stub area is designed to reduce the size of routing tables and LSDBs within the area. Information about external routes (routes from outside the OSPF autonomous system) is not sent into a stub area. To allow devices in a stub area to reach external destinations, the ABR connected to the stub area automatically generates and advertises a default external route into that area.
Not-So-Stubby Area (NSSA)	A more flexible type of stub area. While it also aims to reduce LSA flooding, it allows for the injection of external routes. An NSSA does not flood all LSAs from the OSPF backbone (core) into the area. However, unlike a standard stub area, it can import external routes (AS external routes) into the area through redistribution.

- Device roles:

  In OSPF, devices are assigned specific roles based on their deployment and function within the network hierarchy.

  Table lists the device roles and their classifications.

Device roles	Classification
ABR	An OSPF-enabled device is classified as an Area Border Router (ABR) if it has interfaces in more than one OSPF area and at least one interface in Area 0. They connect regular areas to the backbone. ABRs maintain separate LSDBs for each area they are connected to and are responsible for summarizing routing information between areas.
ASBR	A device is classified as an Autonomous System Boundary Router (ASBR) if it has one interface connected to an OSPF network and at least one interface connected to a non-OSPF network (or another OSPF domain if routes are being explicitly redistributed). This external network could be running a different routing protocol like BGP, EIGRP, or RIP or simply consist of static routes that are being injected into OSPF.

- Link-State Advertisement (LSA):

  Packets containing information about a device's neighbors and the state of its links. Devices exchange LSAs to share network topology information.

- Link-State Database (LSDB):

  A database that stores information about the network topology, built from the LSAs received from neighboring devices. Each device in an OSPF area maintains an identical LSDB.

- OSPF tables:

  OSPF devices maintain several tables to store routing and topology information:

  - Neighbor Table (Adjacency Database): Stores information about OSPF neighbors.
  - Topology Table (Link-State Database): Stores the topology structure of a network.
  - Routing Table (Forwarding Database): Stores the best routes to each destination network.

- Hello Packets:

Used by OSPF devices to discover neighbors and establish neighbor relationships.

- Shortest Path First (SPF) Algorithm (Dijkstra's Algorithm):

  Used by each OSPF device to calculate the best path to each destination based on cost, using the information in the LSDB.

- Cost (Metric):

  A parameter used by OSPF to determine the best path, typically based on interface bandwidth.

To learn more about OSPF, refer the additional references section in this chapter.

How OSPF works

Workflow

Before OSPF is configured, the switch must have IP routing enabled to allow it to forward traffic like a router.

The following section details how OSPF is configured and works on access switches:

1. Each VLAN on the access switch will have an SVI configured with an IP address, acting as the default gateway for that VLAN. OSPF is enabled on these SVIs. This means the access switch will advertise the networks (VLAN subnets) directly connected to it into the OSPF domain.

   If there are dedicated links between access switches or links to distribution switches configured as routed ports (Layer 3 ports), OSPF will also run on these physical interfaces.

2. The access layer switches will form OSPF neighbor relationships with their upstream distribution layer switches (which are also running OSPF).

3. The access switches will exchange LSAs (Link-State Advertisements) with the distribution switches, sharing information about their directly connected networks (VLANs). They will build an LSDB (Link-State Database) that reflects the topology of their area.

4. Each access switch will run the SPF algorithm based on its LSDB and populate its IP routing table with the best paths to all known destinations, including routes learned from the distribution layer.

5. The access switch can now perform inter-VLAN routing locally, directly routing traffic between VLANs that terminate on that switch, without needing to send the traffic up to the distribution layer.

   For traffic destined outside its local VLANs, the access switch will use the OSPF-learned routes to forward packets upstream to the distribution layer.

Enable OSPF and configure area ID

Perform this task to enable OSPF, specify the range of IP addresses to associate with the routing process, and assign area IDs to be associated with that range.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **network** *address wild-card-mask* **area** *area-id*

Example:

```
Device(config-router)# network 10.1.1.1 255.240.0.0 area 20
```

Defines an interface on which OSPF runs and the area ID for that interface.

- *address wild-card-mask*: Addresses of the networks that belong to a particular OSPF area. The *wildcard-mask* allows to use a single command to define one or more multiple interfaces to be associated with a specific OSPF area.
- *area-id*: The area identifer. The area ID can be a decimal value or an IP address.

Step 5 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

Step 6 **show ip protocols**

Example:

```
Device# show ip protocols
```

(Optional) Verifies your entries.

Step 7 **copy running-config startup-config**

Example:

```
Device# copy running-config startup-config
```

(Optional) Saves your entries in the configuration file.

Configure OSPF for IPv6

Before you begin

You can customize OSPF for IPv6 for your network. However, the current configuration defaults for OSPF in IPv6 are set to meet the requirements of most customers and features.

Follow these guidelines:

- Be careful when changing the defaults for IPv6 commands. Changing the defaults might adversely affect OSPF for the IPv6 network.
- Before you enable IPv6 OSPF on an interface, you must enable routing by using the **ip routing** command in global configuration mode, enable the forwarding of IPv6 packets by using the **ipv6 unicast-routing** command in global configuration mode, and enable IPv6 on Layer 3 interfaces on which you are enabling IPv6 OSPF.

For more information about configuring OSPF routing for IPv6, see the “Implementing OSPF for IPv6” chapter in the Cisco IOS IPv6 Configuration Library on Cisco.com.

To configure OSPF routing for IPv6, perform this procedure:

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 Choose one of the following:

- **router ospfv3** *process-id*
- **ipv6 router ospf** *process-id*

Example:

```
Device(config)# router ospfv3 15
```

```
OR
Device(config)# ipv6 router ospf 15
```

Enables OSPF routing and enters router configuration mode.

process-id: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

Process ID can be a positive integer from 1 to 65535.

Step 4 **area** *area-id* **range** {*ipv6-prefix* / *prefix-length*} [**advertise** | **not-advertise**] [**cost** *cost*]

Example:

```
Device(config-router)# area .3 range 2001:0DB8::/32 not-advertise
```

(Optional) Consolidates and summarizes routes at an area boundary.

- *area-id*: Identifier of the area about which routes are to be summarized. It can be specified as either a decimal value or as an IPv6 prefix.
- *ipv6-prefix*/*prefix length*: The destination IPv6 network and a decimal value that shows how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark (/) must precede the decimal value.
- **advertise**: (Optional) Sets the address range status to advertise and generate a Type 3 summary link-state advertisement.
- **not-advertise**: (Optional) Sets the address range status to DoNotAdvertise. The Type 3 summary LSA is suppressed, and component networks remain hidden from other networks.
- **cost** *cost*: (Optional) Sets the metric or cost for this summary route, which is use during OSPF SPF calculation to determine the shortest paths to the destination.

  The value can be 0 to 16777215.

Step 5 **maximum-paths** *number-paths*

Example:

```
Device(config-router)# maximum paths 16
```

(Optional) Defines the maximum number of equal-cost routes to the same destination that IPv6 OSPF should enter in the routing table.

The range is from 1 to 32. The default is 16 paths.

Step 6 **exit**

Example:

```
Device(config-router)# exit
```

Returns to global configuration mode.

Step 7 **interface** *interface-id*

Example:

```
Device(config)# interface gigabitethernet 1/0/1
```

Enters interface configuration mode, and specifies the Layer 3 interface to configure.

Step 8 Choose one of the following:

- **router ospfv3** *process-id* **area** *area-id* [**instance** *instance-id*]
- **ipv6 ospf** *process-id* **area** *area-id* [**instance** *instance-id*]

Example:

```
Device(config-if)# router ospfv3 21 area .3
OR
Device(config-if)# ipv6 ospf 21 area .3
```

Enables OSPF for IPv6 on the interface.

instance *instance-id*: (Optional) Instance identifier.

Step 9 **end**

Example:

```
Device(config-if)# end
```

Returns to privileged EXEC mode.

Step 10 **show ipv6 ospf** [*process-id*] [*area-id* **interface** [*interface-id*]

Example:

```
Device# show ipv6 ospf 21 interface gigabitethernet 2/0/1
```

(Optional) Displays information about OSPF interfaces.

Step 11 **show ipv6 ospf** [*process-id*] [*area-id*

Example:

```
Device# show ipv6 ospf 21
```

(Optional) Displays general information about OSPF routing processes.

Step 12 **copy running-config startup-config**

Example:

```
Device# copy running-config startup-config
```

(Optional) Saves your entries in the configuration file.

Configure OSPF interfaces

You can use the **ip ospf** interface configuration commands to modify interface-specific OSPF parameters. You are not required to modify any of these parameters, but some interface parameters (hello interval, dead interval, and authentication key) must be consistent across all devices in an attached network. If you modify these parameters, be sure all devices in the network have compatible values.

Note The **ip ospf interface** configuration commands are all optional.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *interface-id*

Example:

```
Device(config)# interface gigabitethernet 1/0/1
```

Enters interface configuration mode, and specifies the Layer 3 interface to configure.

Step 4 **ip ospf cost** *cost*

Example:

```
Device(config-if)# ip ospf cost 8
```

(Optional) Explicitly specifies the cost of sending a packet on the interface.

Step 5 **ip ospf retransmit-interval** *seconds*

Example:

```
Device(config-if)# ip ospf retransmit-interval 10
```

(Optional) Specifies the number of seconds between link state advertisement transmissions.

The range is 1 to 65535 seconds. The default is 5 seconds.

Step 6 **ip ospf transmit-delay** *seconds*

Example:

```
Device(config-if)# ip ospf transmit-delay 2
```

(Optional) Sets the estimated number of seconds to wait before sending a link state update packet.

The range is 1 to 65535 seconds. The default is 1 second.

Step 7 **ip ospf priority** *number*

Example:

```
Device(config-if)# ip ospf priority 5
```

(Optional) Sets priority to help find the OSPF designated device for a network.

The range is from 0 to 255. The default is 1.

Step 8 **ip ospf hello-interval** *seconds*

Example:

```
Device(config-if)# ip ospf hello-interval 12
```

(Optional) Sets the number of seconds between hello packets sent on an OSPF interface. The value must be the same for all nodes on a network.

The range is 1 to 65535 seconds. The default is 10 seconds.

Step 9 **ip ospf dead-interval** *seconds*

Example:

```
Device(config-if)# ip ospf dead-interval 8
```

(Optional) Sets the number of seconds after the last device hello packet was seen before its neighbors declare the OSPF device to be down. The value must be the same for all nodes on a network.

The range is 1 to 65535 seconds. The default is 4 times the hello interval.

Step 10 **ip ospf authentication-key** *key*

Example:

```
Device(config-if)# ip ospf authentication-key password
```

Optional) Assign a password to be used by neighboring OSPF devices. The password can be any string of keyboard-entered characters up to 8 bytes in length. All neighboring devices on the same network must have the same password to exchange OSPF information.

Step 11 **ip ospf message digest-key** *keyid* **md5** *key*

Example:

```
Device(config-if)# ip ospf message digest-key 16 md5 your1pass
```

(Optional) Enables MD5 authentication.

- *keyid*: An identifier from 1 to 255.
- *key*: An alphanumeric password of up to 16 bytes.

Step 12 **ip ospf database-filter all out**

Example:

```
Device(config-if)# ip ospf database-filter all out
```

(Optional) Block flooding of OSPF LSA packets to the interface. By default, OSPF floods new LSAs over all interfaces in the same area, except the interface on which the LSA arrives.

Step 13 **end**

Example:

```
Device(config-if)# end
```

Returns to privileged EXEC mode.

Step 14 **show ip ospf interface** *inteface-id*

Example:

```
Device# show ip ospf interface gigabitethernet 1/0/1
```

(Optional) Displays OSPF-related interface information.

Step 15 **show ip ospf neighbor detail**

Example:

```
Device# show ip ospf neighbor detail
```

(Optional) Displays NSF awareness status of neighbor switch. The output matches one of these examples:

- ```
  Optionsis 0x52
  LLSOptions is 0x1 (LR)
  ```

  When both of these lines appear, the neighbor switch is NSF aware.

- ```
  LLSOptions is 0x1 (LR)
  ```

  When both of these lines appear, the neighbor switch is NSF aware.

Step 16 **copy running-config startup-config**

Example:

```
Device# copy running-config startup-config
```

(Optional) Saves your entries in the configuration file.

Configure OSPF area parameters

Before you begin

The OSPF area parmeter configurations are all optional.

Perform this procedure to configure the OSPF area parameters.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **area** *area-id* **authentication**

Example:

```
Device(config-router)# area 1 authentication
```

(Optional) Allow password-based protection against unauthorized access to the identified area.

The identifier can be either a decimal value or an IP address.

Step 5 **area** *area-id* **authentication message-digest**

Example:

```
Device(config-router)# area 1 authentication message-digest
```

(Optional) Enables MD5 authentication on the area.

Step 6 **area** *area-id* **stub** [**no-summary**]

Example:

```
Device(config-router)# area 1 stub
```

(Optional) Define an area as a stub area.

The **no-summary** keyword prevents an ABR from sending summary link advertisements into the stub area.

Step 7 **area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate**] [**no-summary**]

Example:

```
Device(config-router)# area 1 nssa default-information originate
```

(Optional) Defines an area as a not-so-stubby-area. Every device within the same area must agree that the area is NSSA. Select one of these keywords:

- **no-redistribution**: Select when the device is an NSSA ABR and you want the redistribute command to import routes into normal areas, but not into the NSSA.
- **default-information-originate**: Select on an ABR to allow importing type 7 LSAs into the NSSA.
- **no-summary**: Select to not send summary LSAs into the NSSA.

Step 8 **area** *area-id* **range** *address-mask*

Example:

```
Device(config-router)# area 1 range 255.240.0.0
```

Optional) Specifies an address range for which a single route is advertised. Use this command only with area border routers.

Step 9 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

Step 10 **show ip ospf** *process-id*

Example:

```
Device# show ip ospf
```

(Optional) Displays information about the OSPF routing process in general or for a specific process ID to verify configuration.

Step 11 **show ip ospf** [*process-id* [*area-id*]] **database**

Example:

```
Device# show ip ospf database
```

(Optional) Displays lists of information related to the OSPF database for a specific device.

Step 12 **copy running-config startup-config**

Example:

```
Device# copy running-config startup-config
```

(Optional) Saves your entries in the configuration file.

Parameters that optimize OSPF

These OSPF features and parameters allow for fine-tuning and optimization of an OSPF network.

OSPF parameters	Features
Authentication	Provides password-based protection to prevent unauthorized access to an OSPF area. This secures the routing updates exchanged within that area.
Route summarization	Consolidates multiple advertised network addresses into a single, summary route. This reduces the number of entries in routing tables and the amount of routing information exchanged. If network numbers are contiguous, an Area Border Router (ABR) can be configured using the **area range** router configuration command to advertise a single summary route that encompasses all networks within that range to other areas.
Route summarization (for redistributed routes)	Reduces the size of the OSPF LSDB and routing tables when external routes (from other protocols) are redistributed into OSPF. Instead of advertising each redistributed route individually as an external LSA (Type 5), the **summary-address** router configuration command allows you to consolidate multiple contiguous redistributed routes into a single summary route.

OSPF parameters	Features
Virtual links	Maintains backbone continuity in OSPF when a non-backbone area separates a standard area from Area 0, or when Area 0 itself is segmented. Use the **area** *area-id* **virtual-link** *router-id* command to configure a virtual link. This establishes a logical connection between two ABRs through a non-backbone (transit) area, effectively extending the backbone.
Domain Name Server (DNS) names in display	Enhances readability and ease of identification in OSPF **show** command outputs. Allows OSPF displays to show router DNS names instead of just their Router IDs or Neighbor IDs, making it simpler for administrators to identify specific devices.
Default metrics (cost calculation)	Determines the "cost" of a link, which OSPF uses to calculate the shortest path. OSPF calculates the cost of an interface based on its bandwidth using the formula: `cost = reference_bandwidth / interface_bandwidth` The default reference bandwidth is 100 Mbps. For high-bandwidth links (e.g., Gigabit Ethernet), the reference bandwidth can be increased to allow OSPF to differentiate costs more accurately.
Administrative distance	Rates the trustworthiness of routing information learned from different sources. An integer value between 0 and 255, where a lower value indicates higher trustworthiness. OSPF uses different default administrative distances for intra-area, inter-area, and external routes. These values can be changed by the administrator.
Passive interfaces	Prevents OSPF from sending Hello packets out of a specific interface, typically used on interfaces connected to end-user segments where no OSPF neighbors are expected. Once an interface is configured as passive, OSPF stops forming adjacencies on that segment. However, the network connected to the interface can still be advertised by OSPF.
Route calculation timers	Controls the timing of OSPF's SPF algorithm calculations. Allows configuration of a delay time between a topology change and the start of the SPF calculation, and a hold time between consecutive SPF calculations. This helps to prevent excessive CPU utilization during periods of network instability.
Log neighbor changes	Provides real-time visibility into OSPF neighbor state transitions. Configure the OSPF-enabled switch to send a syslog message whenever an OSPF neighbor’s state changes, for example, from Down to Full, or Full to Down. This feature aids network monitoring and troubleshooting.

Configure OSPF parameters

Perform this task to configure a stub area, configure a virtual link, configure DNS look up and so on.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.

- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **summary-address** *address-mask*

Example:

```
Device(config-router)# summary-address 10.1.1.1
```

(Optional) Specifies an address and IP subnet mask for redistributed routes so that only one summary route is advertised.

Step 5 **area** *area-id* **virtual-link** *router-id* [**hello-interval-seconds**] [**retransmit-interval-seconds**] [**trans**] [**authentication-key** *key*] | **message-digest-key** *keyid* **md5** *key*]]

Example:

```
Device(config-router)# area 2 virtual-link 192.168.255.1 hello-interval 5
```

(Optional) Define an area as a stub area. The no-summary keyword prevents an ABR from sending summary link advertisements into the stub area.

Step 6 **area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate**] [**no-summary**]

Example:

```
Device(config-router)# area 1 nssa default-information originate
```

(Optional) Establishes a virtual link and set its parameters.

Step 7 **default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type-value*] [**router-map** *map-name*

Example:

```
Device(config-router)# default information originate metric 100 metric-type 1
```

(Optional) Forces the ASBR to generate a default route into the OSPF routing domain. Parameters are all optional.

Step 8 **ip ospf name-lookup**

Example:

```
Device(config-router)# ip ospf name-lookup
```

(Optional) Configures DNS name lookup. The default is disabled.

Step 9 **auto-cost reference-bandwidth** *ref-bw*

Example:

```
Device(config-router)# auto-cost reference-bandwidth ref-bw
```

(Optional) Specifies the reference bandwidth in Mbps, and is used to automatically calculate the OSPF interface cost. Auto-cost is calculated using `ref-bw/interface-bw`.

- The default ref-bw is 100 Mbps. The range is from 1 to 4294967.
- The interface-bw is based on the interface's operating speed, and is set using the **bandwidth x** command in interface configuration mode.

Step 10 **distance ospf** {[**inter-area** *dist1*] [**inter-area** *dist2*] [**external** *dist3*]}

Example:

```
Device(config)# distance ospf inter-area 150
```

(Optional) Changes the OSPF distance values. The default distance for each type of route is 110. The range is 1 to 255.

Step 11 **passive-interface** *type number*

Example:

```
Device(config)# passive-interface gigabitethernet 1/0/6
```

(Optional) Suppresses the sending of hello packets through the specified interface.

Step 12 **timers throttle spf** *spf-delay spf-holdtime spf-wait*

Example:

```
Device(config)# timers throttle spf 200 100 100
```

(Optional) Configures route calculation timers.

- *spf-delay*—Delay between receiving a change to SPF calculation. The range is from 1 to 600000 miliseconds.
- *spf-holdtime*—Delay between first and second SPF calculation. The range is from 1 to 600000 in milliseconds.

- *spf-wait*—Maximum wait time in milliseconds for SPF calculations. The range is from 1 to 600000 in milliseconds.

Step 13 **ospf log-adj-changes**

Example:

```
Device(config)# ospf log-adj-changes
```

(Optional) Sends syslog message when a neighbor state changes.

Step 14 **end**

Example:

```
Device(config)# end
```

Returns to privileged EXEC mode.

Step 15 **show ip ospf** [*process-id* [*area-id*]] **database**

Example:

```
Device# show ip ospf database
```

Displays lists of information related to the OSPF database for a specific router.

Step 16 **copy running-config startup-config**

Example:

```
Device# copy running-config startup-config
```

(Optional) Saves your entries in the configuration file.

CHAPTER 2

OSPF Area Transit Capability

Feature History for OSPF Area Transit Capability

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPF Area Transit Capability: The OSPF Area Transit Capability is an enhancement feature that allows any OSPF area, other than Area 0 (backbone area), to act as a transit area.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

OSPF Area Transit Capability

The OSPF Area Transit Capability is an enhancement feature that allows any OSPF area, other than Area 0 (backbone area), to act as a transit area. A transit area can forward traffic that does not originate nor terminate within the area.

How OSPF Area Transit Capability works

By default, only Area 0 is used for inter-area transit traffic. Sometimes, devices in non-backbone areas can provide a shorter path for traffic between other areas. With Area Transit Capability enabled, an OSPF Area Border Router (ABR) can recognize and use these shorter transit paths through non-backbone areas. This may provide more optimal routing compared to virtual links, which are normally used to connect non-backbone areas to Area 0.

Disable OSPF Area Transit Capability on an ABR

OSPF area capability is enabled by default. To disable OSPF area transit capability on an ABR, perform this procedure.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **no capability transit**

Example:

```
Device(config-router)# no capability transit
```

Disables OSPF area transit capability on all areas for an OSPF process.

Step 5 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

Configuration example for OSPF Area Transit Capability

This example shows how to disable OSPF Area Transit Capability.

```
Device> enable
Device# configure terminal
Device(config)# router ospf 15
Device(config-router)# no capability transit
```

CHAPTER 3

OSPF Flooding Reduction

Feature History for OSPF Flooding Reduction

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPF Flooding Reduction: OSPF Flooding Reduction is a network optimization feature that minimizes unnecessary OSPF LSA refreshes and network flooding, especially for LSAs that are stable and unchanged.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

OSPF Flooding Reduction

The OSPF Flooding Reduction is a network optimization feature that minimizes unnecessary OSPF Link-State Advertisement (LSA) refreshes and network flooding, especially for LSAs that are stable and unchanged.

How OSPF Flooding Reduction works

OSPF devices periodically refresh LSAs to prevent them from aging out (default every 30 minutes). With flooding reduction enabled, devices set the DoNotAge (DNA) bit in their LSAs. LSAs marked with the DNA bit do not age out, so they are not periodically refreshed or flooded. This reduces control-plane overhead and conserves bandwidth, especially in stable networks with few topology changes.

Configure OSPF Flooding Reduction

Perform this task on each interface you want to configure OSPF flooding reduction.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type number*

Example:

```
Device(config)# interface ethernet 1/0/1
```

Configures an interface.

Step 4 **ip ospf flood-reduction**

Example:

```
Device(config-if)# ip ospf flood-reduction
```

Reduces unnecessary flooding and refreshing of LSAs in stable networks.

Step 5 **end**

Example:

```
Device(config-if)# end
```

Returns to privileged EXEC mode.

Configuration example for OSPF Flooding Reduction

This example shows how to configure OSPF Flooding Reduction.

```
Device> enable
Device# configure terminal
Device(config)# interface ethernet 1/0/1
```

```
Device(config-if)# ip ospf flood-reduction
Device(config-if)# end
```

CHAPTER 4

OSPF Forwarding Address Suppression in Translated Type-5 LSAs

Feature History for OSPF Forwarding Address Suppression in Translated Type-5 LSAs

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPF Forwarding Address Suppression in Translated Type-5 LSAs: The OSPF Forwarding Address Suppression is an enhancement feature that modifies how NSSA ABRs translate Type-7 LSAs to Type-5 LSAs when sending external routes into the OSPF backbone.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

OSPF Forwarding Address Suppression in Translated Type-5 LSAs

The OSPF Forwarding Address Suppression is an enhancement feature that modifies how Not-So-Stubby Area (NSSA) Area Border Routers (ABRs) translate Type-7 LSAs to Type-5 LSAs when sending external routes into the OSPF backbone. Specifically, it suppresses the forwarding address by setting it to 0.0.0.0 in the translated Type-5 LSA.

How OSPF Forwarding Address Suppression in Translated Type-5 LSAs works

In OSPF, NSSA areas use Type-7 LSAs to describe external routes. When an NSSA ABR translates a Type-7 LSA to a Type-5 LSA (so it can be advertised into Area 0), it usually copies the forwarding address from the Type-7 LSA. With Forwarding Address Suppression enabled, the ABR sets the forwarding address in the Type-5 LSA to 0.0.0.0—regardless of the original Type-7 LSA's forwarding address.

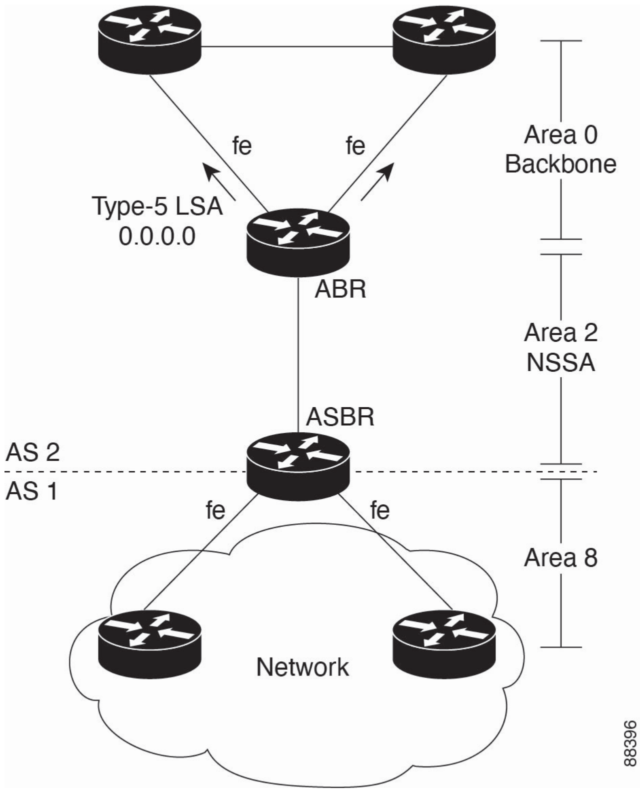
Why to suppress OSPF Forwarding Address in translated Type-5 LSAs

In large OSPF deployments, route summarization is used at area boundaries (ABRs) to reduce the size of routing tables in Area 0 (the backbone) and enhance network scalability. This is typically achieved using the **area range** command on ABRs. This command consolidates multiple more-specific prefixes from a non-backbone area, such as Area 2 and Area 8 in Figure, into a single summary route advertised into the backbone.

When an ASBR in an NSSA redistributes external routes, such as those from a connected network or another routing protocol (see ASBR in Area 2 in Figure), it generates Type-7 LSAs. These Type-7 LSAs are specific to NSSAs and are used to carry information about external routes within the NSSA. The NSSA ABR translates these Type-7 LSAs into Type-5 LSAs (Autonomous System External LSAs) before flooding them into the backbone (Area 0). See ABR connecting Area 2 to Area 0 in Figure. This allows the external routes to be propagated throughout the rest of the OSPF domain.

The Type-7 and Type-5 LSAs include a forwarding address—often the interface address of the ASBR that originated the external route. This address instructs OSPF devices where to forward traffic destined for the external prefix.

When the **area range** command is used, only the summary route is advertised into Area 0—not the specific prefixes or host routes within Area 2. If the forwarding address in a Type-5 LSA is a specific address within Area 2 that has been suppressed by summarization, backbone devices may not have a route to this address. As a result, the external prefix (though present in the routing table) becomes unreachable from the backbone, since traffic cannot be routed to the correct ASBR.



Before configuring this feature, consider the following caution.

Caution Configuring this feature causes the device to be noncompliant with RFC 1587. In addition, suboptimal routing may occur if better paths exist to reach the destination's forwarding address. Configure this feature only after careful consideration and once the network topology is understood.

Suppress OSPF Forwarding Address in Translated Type-5 LSAs

Perform this task to suppress the OSPF forwarding address in translated Type-5 LSAs.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **area** *area-id* **nssa translate type7 suppress-fa**

Example:

```
Device(config-router)# area 10 nssa translate type7 suppress-fa
```

Configures an area as a NSSA and suppresses the forwarding address in translated Type-7 LSAs.

Configuration example of OSPF Forwarding Address Suppression in translated Type-5 LSAs

This example shows how to configure OSPF Forwarding Address Suppression in Translated Type-5 LSAs

```
Device> enable
Device# configure terminal
Device(config)# router ospf 15
Device(config-router)# area 10 nssa translate type7 suppress-fa
```

CHAPTER 5

OSPF Inbound Filtering Using Route Maps with a Distribute List

Feature History for OSPF Inbound Filtering using Route Maps with a Distribute List

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPF Inbound Filtering Using Route Maps with a Distribute List: The OSPF Inbound Filtering Using Route Maps with a Distribute List is an enhancement feature that enables administrators to control which OSPF-learned routes are installed in the local routing table.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

OSPF Inbound Filtering using Route Maps with a Distribute List

The OSPF Inbound Filtering Using Route Maps with a Distribute List is an enhancement feature that enables administrators to control which OSPF-learned routes are installed in the local routing table. With this enhancement, you can apply more granular filtering criteria based on route attributes instead of relying solely on prefix lists or access lists.

Match criteria in route maps and practical use cases

Match Criteria in Route Maps

Users can match on the following route attributes:

- **match interface**: The outgoing interface for the route OSPF is installing.
- **match ip address**: The route's prefix (prefix-lists or access-lists).
- **match ip next-hop**: The next-hop IP address for the route.
- **match ip route-source**: The OSPF Router ID that originated the LSA advertising the prefix.
- **match metric**: The route's OSPF metric.
- **match route-type**: The OSPF route type (e.g., intra-area, inter-area, external Type 1, external Type 2).
- **match tag**: A tag value assigned to the route, often used during redistribution.

Practical use cases

Filter type	Use case
Filtering based on route tag	Assign tags to routes during redistribution (e.g., on an ASBR). Use a route map to deny or permit installation of routes with specific tags.
Filtering based on route type	Match on external Type 1 or Type 2 routes or internal (intra-area or inter-area) routes to filter specific types.
Filtering based on route source	Match on the OSPF Router ID of the LSA originator.
Filtering based on interface	Match on the local interface that OSPF would use to reach the route.
Filtering based on next-hop	Match on the next-hop IP for further control.

How OSPF Inbound Filtering using Route Maps with a Distribute list works

Traditionally, OSPF inbound filtering uses prefix-lists or access-lists to filter routes. With this enhancement, you can use a route map with a distribute-list to filter OSPF routes. Route maps allow for matching on multiple route attributes, such as prefix, next-hop, metric, tag, and more. Only OSPF routes that match the criteria in the route map will be installed in the local routing table; others are discarded.

Configure OSPF Inbound Filtering using a Route Map

Perform this procedure to configure OSPF inbound filtering using a route map.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]

Example:

```
Device(config)# route-map tag-filter deny 10
```

Defines a route map to control filtering.

- *map-tag*: A name to identify the route map (for example, OSPF-INBOUND-FILTER).
- **permit** | **deny**: Specifies whether matched routes will be permitted (allowed into the routing table) or denied (filtered out).
- *sequence-number*: (Optional) Used to order multiple match conditions within the same route map.

Step 4 **match tag** *tag-name*

Example:

```
Device(config-router)# match tag 777
```

Matches routes with a specified name, to be used as the route map is referenced.

tag-value: The numeric tag value assigned to the route (typically set during redistribution).

- At least one **match** command is required, but it need not be this **match** command. This example illustrates one possible configuration.
- The available **match** commands for this type of route map are listed in the **distribute-list in** command reference page.
- This type of route map does not include any **set** commands.

Step 5 Repeat Steps 3 and 4 with other route-map and match commands if you choose.

Step 6 **exit**

Example:

```
Device(config-router)# exit
```

Exits router configuration mode.

Step 7 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 8 **distribute-list route-map** *map-tag* **in**

Example:

```
Device(config-router)# distribute-list route-map tag-filter in
```

Enables filtering based on an OSPF route map.

Step 9 **end**

Example:

```
Device(config-router)# end
```

Exits router configuration mode.

Configuration example for OSPF Route-Map based Giltering

This example shows how to configure OSPF route-map based filtering.

In this example, the device is set up to filter OSPF external routes based on their tag values. A route map called *tag-filter* is created with two statements:

- the first denies any route with a tag value of 777, and
- the second permits all other routes by default since it has no specific match conditions.

This route map is then applied as an inbound distribute-list to OSPF process 1. As a result, when OSPF routes are received, any external route tagged with 777 is filtered out and not installed in the routing table, while all other routes are permitted. This approach ensures that only routes without the specific tag are allowed, effectively controlling which external OSPF prefixes are accepted by the device.

```
Device> enable
Device# configure terminal
Device(config)# route-map tag-filter deny 10
Device(config-router)# match tag 777
Device(config-router)# route-map tag-filter permit 20
Device(config-router)# exit
Device(config)# router ospf 1
Device(config-router)# router-id 10.0.0.2
Device(config-router)# log-adjacency-changes
Device(config-router)# network 172.16.2.1 0.0.0.255 area 0
Device(config-router)# distribute-list route-map tag-filter in
```

CHAPTER 6

OSPF Incremental SPF Support

Feature History for OSPF Incremental SPF Support

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPF incremental SPF Support: The OSPF Incremental SPF feature is a network optimization feature that allows a device to use incremental SPF algorithm for calculating the shortest path first SPF routes.	Cisco C9610 Series Smart Switches

OSPF Incremental SPF Support

OSPF Incremental SPF Support is a network optimization feature that allows a device to use incremental shortest path first (SPF) algorithm for calculating the SPF routes.

How OSPF Incremental SPF works

Normally, OSPF uses Dijkstra's SPF algorithm to compute the shortest path tree (SPT). During the computation of the SPT, the shortest path to each node is discovered. The topology tree populates the routing table with routes to IP networks.

When changes to a Type-1 or Type-2 link-state advertisement (LSA) occur in an area, the entire SPT is recomputed. In many cases, the entire SPT need not be recomputed because most of the tree remains unchanged.

Incremental SPF allows the system to recompute only the affected part of the tree. Recomputing only a portion of the tree, rather than the entire tree, results in faster OSPF convergence and saves CPU resources. If the change to a Type-1 or Type-2 LSA occurs in the calculating device itself, the system performs a full SPT.

Incremental SPF is scheduled in the same way as the full SPF. Devices enabled with and without incremental SPF can function in the same network.

Enable Incremental SPF

Perform this procedure to enable Incremental SPF.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **ispf**

Example:

```
Device(config-router)# ispf
```

Enables incremental SPF.

Step 5 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

Configuration example for OSPF Incremental SPF

This example shows how to enable Incremental SPF:

```
Device> enable
Device# configure terminal
Device(config)# router ospf 1
Device(config-router)# ispf
Device(config-router)# end
```

CHAPTER 7

OSPF Limit on Number of Redistributed Routes

Feature History for OSPF Limit on Number of Redistributed Routes

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPF Limit on Number of Redistributed Routes: OSPF Limit on Number of Redistributed Routes is a network enhancement feature that limits the maximum number of prefixes (routes) that can be redistributed into OSPF from other protocols or OSPF processes.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

OSPF Limit on Number of Redistributed Routes

OSPF Limit on Number of Redistributed Routes is a network enhancement feature that limits the maximum number of prefixes (routes) that can be redistributed into OSPF from other protocols or OSPF processes.

Without such a limit, an access switch acting as an Autonomous System Boundary Router (ASBR) could potentially receive a massive number of routes from another routing domain, for example a full BGP routing table from the internet, or a large number of routes from an EIGRP domain, and then redistribute all of them into OSPF.

How OSPF Limit on Number of Redistributed Routes works

When redistribution occurs, for example from BGP, EIGRP, or another OSPF process, the device will only inject up to the specified number of external routes into the OSPF domain. If the configured limit is exceeded, additional routes are not redistributed, helping to protect the OSPF domain from route table overload.

Guidelines to configure OSPF Limit on Number of Redistributed Routes

- This feature is supported only for the IPv6 address family.
- You must have OSPF configured in your network either along with another protocol, or another OSPF process for redistribution.

Configure an OSPF Limit on the Number of Redistributed Routes

Follow the steps in each of these tasks to configure an OSPF limit on the number of redistributed routes

Procedure

Step 1 Limit the number of OSPF redistributed routes

Step 2 Request a warning message about the number of routes redistributed into OSPF

Step 3 Limit the number of OSPFv3 redistributed routes

Step 4 Request a warning message about the number of routes redistributed into OSPFv3

Limit the number of OSPF redistributed routes

This task limits the number of OSPF redistributed routes. If the maximum configured value is reached, no more routes are distributed.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **redistribute** *protocol* [*process-id*] [*as-number*] [**include-connected** {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** *type-value*] [**nssa-only**] [**tag** *tag-value*] [**route-map** *map-tag*]

Example:

```
Device(config-router)# redistribute eigrp 10
```

Redistributes routes from one routing domain into another routing domain.

- *protocol*: Specifies the source of the routes to be redistributed. Common options include:
  - **connected**: Redistributes directly connected networks that are not already part of the OSPF process.
  - **static**: Redistributes static routes configured on the router.
  - **bgp**: Redistributes routes learned via BGP.
  - **eigrp**: Redistributes routes learned via EIGRP.
  - **rip**: Redistributes routes learned via RIP.
  - **ospf**: Redistributes routes from another OSPF process (e.g., if running multiple OSPF instances or different process IDs).
  - **isis**: Redistributes routes learned via IS-IS.
- *process-id*: (Optional) If the source protocol (EIGRP, OSPF, RIP) uses a process ID, you specify it here to identify the specific instance from which routes should be taken.
- *as-number*: (Optional) If the source protocol (BGP or EIGRP) uses an Autonomous System (AS) number, you specify it here. This is often used interchangeably with process-id for protocols that use AS numbers.
- **include-connected** {**level-1** | **level-1-2** | **level-2**}: (Optional) This parameter is typically specific to IS-IS redistribution into OSPF, or sometimes OSPF into IS-IS, or even OSPF into OSPF (though less common). It refers to the IS-IS routing levels:
  - level-1: Redistributes Level-1 IS-IS routes.
  - level-2: Redistributes Level-2 IS-IS routes.

    - level-1-2: Redistributes both Level-1 and Level-2 IS-IS routes.
    - This part of the syntax suggests a context where IS-IS is the source protocol.

- **metric** *metric-value*: (Optional) Specifies the metric (cost) that OSPF will assign to the redistributed routes. This value is used by OSPF to calculate the path cost to these external destinations. If not specified, a default metric (often 1 or 20, depending on the source protocol and platform) will be used.
- **metric-type** *type-value*: (Optional) For OSPF, this is crucial for how external routes are handled:
    - type-1 (E1): The cost to the external destination is the sum of the external metric and the internal OSPF cost to reach the ASBR. This is often preferred for more accurate path selection.
    - type-2 (E2): The cost to the external destination is only the external metric, regardless of the internal OSPF cost to reach the ASBR. This is the default if not specified. Type 2 routes are always preferred over Type 1 routes if their external metric is the same.
- **nssa-only**: (Optional) This keyword is specifically used when redistributing routes into a Not-So-Stubby Area (NSSA). When nssa-only is specified, the redistributed routes will be advertised as Type 7 LSAs within the NSSA, and then translated to Type 5 LSAs by the NSSA ABR for propagation to other areas. Without this, redistribution into an NSSA might not be allowed or might behave differently
- **tag** *tag-value*: (Optional) Assigns a numerical tag to the redistributed routes. This tag can be used later in route maps or distribute lists for filtering or policy-based routing.
- **route-map** *map-tag*: (Optional) A very powerful and commonly used parameter. It references a `route-map` that provides granular control over which routes are redistributed and how their attributes (like metric, metric-type, tag) are modified during the redistribution process.

Step 5 **redistribute maximum-prefix** *maximum* [*threshold*]

Example:

```
Device(config-router-af)# redistribute maximum-prefix 100 80
```

Sets a maximum number of IP prefixes that are allowed to be redistributed into OSPF.

- *maximum*: The maximum value.

  The default value is set at 10240 routes.

- *threshold*: The threshold value defaults to 75 percent.

  Note
  If the warning-only keyword is configured in this command, no limit is enforced; a warning message is logged.

Step 6 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

Request a warning message about the number of routes redistributed into OSPF

This task limits the number of OSPF redistributed routes. If the maximum configured value is reached, a warning message is displayed but routes are still redistributed.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **redistribute** *protocol* [*process-id*] [*as-number*] [**include-connected** {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** *type-value*] [**nssa-only**] [**tag** *tag-value*] [**route-map** *map-tag*]

Example:

```
Device(config-router)# redistribute eigrp 10
```

Redistributes routes from one routing domain into another routing domain.

- *protocol*: Specifies the source of the routes to be redistributed. Common options include:
  - **connected**: Redistributes directly connected networks that are not already part of the OSPF process.
  - **static**: Redistributes static routes configured on the router.
  - **bgp**: Redistributes routes learned via BGP.

  - **eigrp**: Redistributes routes learned via EIGRP.
  - **rip**: Redistributes routes learned via RIP.
  - **ospf**: Redistributes routes from another OSPF process (e.g., if running multiple OSPF instances or different process IDs).
  - **isis**: Redistributes routes learned via IS-IS.

- *process-id*: (Optional) If the source protocol (EIGRP, OSPF, RIP) uses a process ID, you specify it here to identify the specific instance from which routes should be taken.
- *as-number*: (Optional) If the source protocol (BGP or EIGRP) uses an Autonomous System (AS) number, you specify it here. This is often used interchangeably with process-id for protocols that use AS numbers.
- **include-connected** {**level-1** | **level-1-2** | **level-2**}: (Optional) This parameter is typically specific to IS-IS redistribution into OSPF, or sometimes OSPF into IS-IS, or even OSPF into OSPF (though less common). It refers to the IS-IS routing levels:
  - level-1: Redistributes Level-1 IS-IS routes.
  - level-2: Redistributes Level-2 IS-IS routes.
  - level-1-2: Redistributes both Level-1 and Level-2 IS-IS routes.
  - This part of the syntax suggests a context where IS-IS is the source protocol.
- **metric** *metric-value*: (Optional) Specifies the metric (cost) that OSPF will assign to the redistributed routes. This value is used by OSPF to calculate the path cost to these external destinations. If not specified, a default metric (often 1 or 20, depending on the source protocol and platform) will be used.
- **metric-type** *type-value*: (Optional) For OSPF, this is crucial for how external routes are handled:
  - type-1 (E1): The cost to the external destination is the sum of the external metric and the internal OSPF cost to reach the ASBR. This is often preferred for more accurate path selection.
  - type-2 (E2): The cost to the external destination is only the external metric, regardless of the internal OSPF cost to reach the ASBR. This is the default if not specified. Type 2 routes are always preferred over Type 1 routes if their external metric is the same.
- **nssa-only**: (Optional) This keyword is specifically used when redistributing routes into a Not-So-Stubby Area (NSSA). When nssa-only is specified, the redistributed routes will be advertised as Type 7 LSAs within the NSSA, and then translated to Type 5 LSAs by the NSSA ABR for propagation to other areas. Without this, redistribution into an NSSA might not be allowed or might behave differently
- **tag** *tag-value*: (Optional) Assigns a numerical tag to the redistributed routes. This tag can be used later in route maps or distribute lists for filtering or policy-based routing.
- **route-map** *map-tag*: (Optional) A very powerful and commonly used parameter. It references a `route-map` that provides granular control over which routes are redistributed and how their attributes (like metric, metric-type, tag) are modified during the redistribution process.

Step 5 **redistribute maximum-prefix** *maximum* [*threshold*] [**warning-only**]

Example:

```
Device(config-router-af)# redistribute maximum-prefix 100 80 warning-only
```

Causes a warning message to be logged when the maximum number of IP prefixes have been redistributed to OSPF.

- Because the **warning-only** keyword is included, no limit is imposed on the number of redistributed prefixes into OSPF.
- The *threshold* value defaults to 75 percent.
- This example causes two warnings:
  - one at 80 percent of 1000 (800 routes redistributed) and
  - another at 1000 routes redistributed.

Step 6 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

Limit the number of OSPFv3 redistributed routes

This task limits the number of OSPFv3 redistributed routes. If the maximum configured value is reached, no more routes are distributed.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **address-family ipv6** [**unicast**]

Example:

```
Device(config-router)# address-family ipv6 unicast
```

Enters IPv6 address family configuration mode.

unicast: Specifies the configuration is for IPv6 unicast routing.

Step 5 **redistribute** *protocol* [*process-id*] [*as-number*] [**include-connected** {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** *type-value*] [**nssa-only**] [**tag** *tag-value*] [**route-map** *map-tag*]

Example:

```
Device(config-router-af)# redistribute eigrp 10
```

Redistributes routes from one routing domain into another routing domain.

- *protocol*: Specifies the source of the routes to be redistributed. Common options include:
  - **connected**: Redistributes directly connected networks that are not already part of the OSPF process.
  - **static**: Redistributes static routes configured on the router.
  - **bgp**: Redistributes routes learned via BGP.
  - **eigrp**: Redistributes routes learned via EIGRP.
  - **rip**: Redistributes routes learned via RIP.
  - **ospf**: Redistributes routes from another OSPF process (e.g., if running multiple OSPF instances or different process IDs).
  - **isis**: Redistributes routes learned via IS-IS.
- *process-id*: (Optional) If the source protocol (EIGRP, OSPF, RIP) uses a process ID, you specify it here to identify the specific instance from which routes should be taken.
- *as-number*: (Optional) If the source protocol (BGP or EIGRP) uses an Autonomous System (AS) number, you specify it here. This is often used interchangeably with process-id for protocols that use AS numbers.
- **include-connected** {**level-1** | **level-1-2** | **level-2**}: (Optional) This parameter is typically specific to IS-IS redistribution into OSPF, or sometimes OSPF into IS-IS, or even OSPF into OSPF (though less common). It refers to the IS-IS routing levels:
  - level-1: Redistributes Level-1 IS-IS routes.
  - level-2: Redistributes Level-2 IS-IS routes.
  - level-1-2: Redistributes both Level-1 and Level-2 IS-IS routes.
  - This part of the syntax suggests a context where IS-IS is the source protocol.

- **metric** *metric-value*: (Optional) Specifies the metric (cost) that OSPF will assign to the redistributed routes. This value is used by OSPF to calculate the path cost to these external destinations. If not specified, a default metric (often 1 or 20, depending on the source protocol and platform) will be used.
- **metric-type** *type-value*: (Optional) For OSPF, this is crucial for how external routes are handled:
  - type-1 (E1): The cost to the external destination is the sum of the external metric and the internal OSPF cost to reach the ASBR. This is often preferred for more accurate path selection.
  - type-2 (E2): The cost to the external destination is only the external metric, regardless of the internal OSPF cost to reach the ASBR. This is the default if not specified. Type 2 routes are always preferred over Type 1 routes if their external metric is the same.
- **nssa-only**: (Optional) This keyword is specifically used when redistributing routes into a Not-So-Stubby Area (NSSA). When nssa-only is specified, the redistributed routes will be advertised as Type 7 LSAs within the NSSA, and then translated to Type 5 LSAs by the NSSA ABR for propagation to other areas. Without this, redistribution into an NSSA might not be allowed or might behave differently
- **tag** *tag-value*: (Optional) Assigns a numerical tag to the redistributed routes. This tag can be used later in route maps or distribute lists for filtering or policy-based routing.
- **route-map** *map-tag*: (Optional) A very powerful and commonly used parameter. It references a `route-map` that provides granular control over which routes are redistributed and how their attributes (like metric, metric-type, tag) are modified during the redistribution process.

Step 6 **redistribute maximum-prefix** *maximum* [*threshold*] [**warning-only**]

Example:

```
Device(config-router-af)# redistribute maximum-prefix 100 80 warning-only
```

Causes a warning message to be logged when the maximum number of IP prefixes have been redistributed to OSPFv3.

- Because the **warning-only** keyword is included, no limit is imposed on the number of redistributed prefixes into OSPF.
- The *threshold* value defaults to 75 percent.
- This example causes two warnings:
  - one at 80 percent of 1000 (800 routes redistributed) and
  - another at 1000 routes redistributed.

Step 7 **exit-address-family**

Example:

```
Device(config-router-af)# exit-address-family
```

Exits IPv6 address family configuration mode.

Step 8 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

Request a warning message about the number of routes redistributed into OSPFv3

This task limits the number of OSPFv3 redistributed routes. If the maximum configured value is reached, no more routes are distributed.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **address-family ipv6** [**unicast**]

Example:

```
Device(config-router)# address-family ipv6 unicast
```

Enters IPv6 address family configuration mode.

unicast: Specifies the configuration is for IPv6 unicast routing.

Step 5 **redistribute** *protocol* [*process-id*] [*as-number*] [**include-connected** {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** *type-value*] [**nssa-only**] [**tag** *tag-value*] [**route-map** *map-tag*]

Example:

```
Device(config-router-af)# redistribute eigrp 10
```

Redistributes routes from one routing domain into another routing domain.

- *protocol*: Specifies the source of the routes to be redistributed. Common options include:
  - **connected**: Redistributes directly connected networks that are not already part of the OSPF process.
  - **static**: Redistributes static routes configured on the router.
  - **bgp**: Redistributes routes learned via BGP.
  - **eigrp**: Redistributes routes learned via EIGRP.
  - **rip**: Redistributes routes learned via RIP.
  - **ospf**: Redistributes routes from another OSPF process (e.g., if running multiple OSPF instances or different process IDs).
  - **isis**: Redistributes routes learned via IS-IS.
- *process-id*: (Optional) If the source protocol (EIGRP, OSPF, RIP) uses a process ID, you specify it here to identify the specific instance from which routes should be taken.
- *as-number*: (Optional) If the source protocol (BGP or EIGRP) uses an Autonomous System (AS) number, you specify it here. This is often used interchangeably with process-id for protocols that use AS numbers.
- **include-connected** {**level-1** | **level-1-2** | **level-2**}: (Optional) This parameter is typically specific to IS-IS redistribution into OSPF, or sometimes OSPF into IS-IS, or even OSPF into OSPF (though less common). It refers to the IS-IS routing levels:
  - level-1: Redistributes Level-1 IS-IS routes.
  - level-2: Redistributes Level-2 IS-IS routes.
  - level-1-2: Redistributes both Level-1 and Level-2 IS-IS routes.
  - This part of the syntax suggests a context where IS-IS is the source protocol.
- **metric** *metric-value*: (Optional) Specifies the metric (cost) that OSPF will assign to the redistributed routes. This value is used by OSPF to calculate the path cost to these external destinations. If not specified, a default metric (often 1 or 20, depending on the source protocol and platform) will be used.
- **metric-type** *type-value*: (Optional) For OSPF, this is crucial for how external routes are handled:
  - type-1 (E1): The cost to the external destination is the sum of the external metric and the internal OSPF cost to reach the ASBR. This is often preferred for more accurate path selection.
  - type-2 (E2): The cost to the external destination is only the external metric, regardless of the internal OSPF cost to reach the ASBR. This is the default if not specified. Type 2 routes are always preferred over Type 1 routes if their external metric is the same.
- **nssa-only**: (Optional) This keyword is specifically used when redistributing routes into a Not-So-Stubby Area (NSSA). When nssa-only is specified, the redistributed routes will be advertised as Type 7 LSAs within the NSSA, and then translated to Type 5 LSAs by the NSSA ABR for propagation to other areas. Without this, redistribution into an NSSA might not be allowed or might behave differently

- **tag** *tag-value*: (Optional) Assigns a numerical tag to the redistributed routes. This tag can be used later in route maps or distribute lists for filtering or policy-based routing.
- **route-map** *map-tag*: (Optional) A very powerful and commonly used parameter. It references a `route-map` that provides granular control over which routes are redistributed and how their attributes (like metric, metric-type, tag) are modified during the redistribution process.

Step 6 **redistribute maximum-prefix** *maximum* [*threshold*]

Example:

```
Device(config-router-af)# redistribute maximum-prefix 100 80
```

Sets a maximum number of IP prefixes that are allowed to be redistributed into OSPF.

- *maximum*: The maximum value.

  The default value is set at 10240 routes.
- *threshold*: The threshold value defaults to 75 percent.

Note
If the warning-only keyword is configured in this command, no limit is enforced; a warning message is logged.

Step 7 **exit-address-family**

Example:

```
Device(config-router-af)# exit-address-family
```

Exits IPv6 address family configuration mode.

Step 8 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

Configuration examples for OSPF Limit on Number of Redistributed Routes

The following sections provide configuration examples for OSPF Limit on Number of Redistributed Routes.

Example: OSPF Limit on Number of Redistributed Routes

This example shows how to set a maximum of 1200 prefixes that can be redistributed into the OSPF process 1. Prior to reaching the limit, when the number of prefixes that are redistributed reaches 80 percent of 1200 (960 prefixes), a warning message is logged. Another warning message is logged when the limit is reached and no more routes are redistributed.

```
Device> enable
Device# configure terminal
```

```
Device(config)# router ospf 1
Device(config-router-af)# redistribute static subnets
Device(config-router-af)# redistribute maximum-prefix 1200 80
```

This example shows how to set a maximum of 1200 prefixes that can be redistributed into the OSPFv3 process 1.

```
Device> enable
Device# configure terminal
Device(config)# router ospfv3 1
Device(config-router)# address-family ipv6
Device(config-router-af)# redistribute static subnets
Device(config-router-af)# redistribute maximum-prefix 1200 80
```

Example: Requesting a warning message about the number of redistributed routes

This example shows how to enable two warning messages to be logged, the first if the number of prefixes that are redistributed reaches 85 percent of 600 (510 prefixes), and the second if the number of redistributed routes reaches 600. However, the number of redistributed routes is not limited.

```
Device> enable
Device# configure terminal
Device(config)# router ospf 11
Device(config-router-af)# redistribute eigrp 10 subnets
Device(config-router-af)# redistribute maximum-prefix 600 85 warning-only
```

This example shows how to enable two warnings to be logged for an OSSPv3 process.

```
Device> enable
Device# configure terminal
Device(config)# router ospfv3 11
Device(config-router)# address-family ipv6
Device(config-router-af)# redistribute eigrp 10 subnets
Device(config-router-af)# redistribute maximum-prefix 600 85 warning-only
```

CHAPTER 8

OSPF Link-State Database Overload Protection

Feature History for OSPF link state database overload protection

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPF link state database overload protection: The OSPF link state database overload protection is an enhancement feature that limits the number of non-self-generated LSAs for an OSPF or OSPFv3 process.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

OSPF Link-State Database Overload Protection

The OSPF Link-State Database Overload Protection is an enhancement feature that limits the number of non-self-generated Link-State Advertisements (LSAs) for an OSPF or OSPFv3 process.

This feature safeguards the device from CPU and memory exhaustion by limiting the number of non-self-generated LSAs it will accept and process.

How OSPF link-state database overload protection works

OSPF devices maintain a link-state database (LSDB) containing LSAs generated by themselves and other devices in the OSPF domain. If a device receives an excessive number of non-self-generated LSAs (often due to misconfiguration, route leaks, or excessive redistribution), its memory and CPU can become overwhelmed. With LSDB overload protection enabled, the device is configured with a maximum threshold for non-self-generated LSAs. If this threshold is exceeded, the device enters an overload state, stopping further OSPF processing and advertising its overload condition to peers.

Limit the number of non self-generated LSAs for an OSPF process

You can perform this task to configure the maximum number of non self-generated LSAs the switch can receive.

When the configured maximum number of LSAs is exceeded:

- the switch sends a notification and stops accepting any new LSAs. If the count of received LSAs is still higher than the configured maximum, then the OSPF process takes down all adjacencies, clears the OSPF database, and enters the IGNORE state.

  You can configure the **ignore-time** *minutes* to set the time for which the OSPF process can remain in the IGNORE state.

- Each time the OSPF process enters the IGNORE state, a counter is incremented.

  You can set the count on the number of times the OSPF process has entered the IGNORE state using the **ignore-count** *count-number*. When the configured count is exceeded, the OSPF process remains in the IGNORE state. You must restart the OSPF process to restore normal operation.

- If the OSPF process has returned to its normal state of operation, you can configure the **reset-time** *minutes* to specify the duration to wait before the IGNORE state counter is reset.

Additionally if you require warning messages to be displayed, you can use *threshold-percentage* and **warning-only**.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **router-id** *ip-address*

Example:

```
Device(config-router)# router-id 10.0.0.1
```

Specifies a fixed router ID for an OSPF process.

ip-address: The IP address to identify the device in the routing domain.

Step 5 **log-adjacency-changes** [**detail**]

Example:

```
Device(config-router)# log-adjacency-changes
```

Configures the device to send a syslog message when an OSPF neighbor goes up or down.

detail: Logs all adjacency state changes, including states like DOWN, INIT, 2EXCHANGE, LOADING, and so on.

Step 6 **max-lsa** *maximum-number* [*threshold-percentage*] [**warning-only** [**ignore-time** *minutes*] [**ignore-count***count-number*] [**reset-time** *minutes*]

Example:

```
Device(config-router)# max-lsa 12000
```

Limits the number of non self-generated LSAs that an OSPF routing process can keep in the OSPF link-state database (LSDB).

- *maximum-number*: Maximum number of non-self-generated LSAs allowed in the LSDB.
- *threshold-percentage*: (Optional) Percentage of the maximum number at which a warning message is logged. Default is 75%.
- **warning-only**: (Optional) If specified, only a warning message is logged when the limit is exceeded; the OSPF process does not enter the ignore state. Disabled by default.
- **ignore-time** *minutes*: (Optional) Time in minutes to ignore all neighbors after the maximum LSA limit is exceeded. Default is 5 minutes.
- **ignore-count***count-number*: (Optional) Number of times the OSPF process can consecutively enter the ignore state. Default is 5 times.
- **reset-time** *minutes*: (Optional) Time in minutes after which the ignore count is reset to zero. Default is 10 minutes (or 2 times ignore-time in some platforms).

Step 7 **network** *ip-address wildcard-mask* **area** *area-id*

Example:

```
Device(config-router)# network 10.1.1.1 255.240.0.0 area 20
```

Defines an interface on which OSPF runs and the area ID for that interface.

- *address wild-card-mask*: Addresses of the networks that belong to a particular OSPF area. The wildcard-mask allows you to use a single command to define one or more multiple interfaces to be associated with a specific OSPF area.

  area-id: The area identifer. The area identifier can be a decimal value or an IP address.

Step 8 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

Step 9 **show ip ospf** [*process-id area-id*]] **database** *database-summary*

Example:

```
Device(config)# show ip ospf 2000 database database-summary
```

Displays lists of information related to the OSPF database for a specific device.

Use this command to verify the number of non self-generated LSAs on a device.

Configuration example set a limit for LSA generation

In this example, the device is configured to stop accepting non self-generated LSAs after the maximum of 14,000 has been exceeded:

```
Device(config)# router ospf 1
Device(config-router)# router-id 192.168.0.1
Device(config-router)# log-adjacency-changes
Device(config-router)# max-lsa 14000
Device(config-router)# area 33 nssa
Device(config-router)# network 192.168.0.10.0.0.0 area 1
Device(config-router)# network 192.168.5.10.0.0.0 area 1
Device(config-router)# network 192.168.2.10.0.0.0 area 0
```

In this example, the device is configured to stop accepting non self-generated LSAs once a maximum of 12,000 has been exceeded for an OPSFv3 process:

```
Device> enable
Device# configure terminal
Device(config)# router ospfv3 1
Device(config-router)# router-id 10.0.0.1
Device(config-router)# log-adjacency-changes
Device(config-router)# max-lsa 12000
```

In this example, the **show ip ospf** command is entered to confirm the configuration:

```
Device# show ip ospf 1
Routing Process "ospf1" with ID 192.168.0.1
Supports only single TOS(TOS0)routes
Supports opaque LSA
Supports Link-local Signaling(LLS)
```

```
Supports area transit capability
Maximum number of nonself-generated LSA allowed 14000
Threshold for warning message75%
Ignore-time 5minutes,reset-time 10minutes
Ignore-countallowed 5,currentignore-count 0
```

In this example, the output is displayed when the **show ip ospf** command is entered when the device is in the ignore state:

```
Device# show ip ospf 1
Routing Process "ospf1" with ID 192.168.0.1
Supports only single TOS(TOS0)routes
Supports opaque LSA
Supports Link-local Signaling(LLS)
Supports area transit capability
Maximum number of nonself-generated LSA allowed 14000
Threshold for warning message 75%
Ignore-time 5minutes, reset-time 10minutes
Ignore-count allowed 5,current ignore-count 1
Ignoring all neighbors due to max-lsa limit, time remaining: 00:04:52
```

This output is displayed when the **show ip ospf** command is entered after the device left the ignore state:

```
Device# show ip ospf 1
Routing Process "ospf 1" with ID 192.168.0.1
Supports only single TOS(TOS0) routes
Supports opaque LSA Supports Link-local Signaling (LLS)
Supports area transit capability
Maximum number of non self-generated LSA allowed 14000
Threshold for warning message 75%
Ignore-time 5 minutes, reset-time 10 minutes
Ignore-count allowed 5, current ignore-count 1- time remaining: 00:09:51
```

This output is displayed when the **show ip ospf** command is entered for a device that is permanently in the ignore state:

```
Device# show ip ospf 1
Routing Process "ospf 1" with ID 192.168.0.1
Supports only single TOS(TOS0) routes
Supports opaque LSA Supports Link-local Signaling (LLS)
Supports area transit capability
Maximum number of non self-generated LSA allowed 14000
Threshold for warning message 75%
Ignore-time 5 minutes, reset-time 10 minutes
Ignore-count allowed 5, current ignore-count 6
Permanently ignoring all neighbors due to max-lsa limit
```

CHAPTER 9

OSPF Per-Interface Link-Local Signaling

Feature History for OSPF link-local signaling per interface basis

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPF link-local signaling per interface basis: The OSPF link-local signaling per interface basis is an enhancement feature that allows you to selectively enable or disable link-local signaling for a specific interface regardless of the device 's global OSPF link-local signaling configuration.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

OSPF link-local signaling per-interface basis

OSPF link-local signaling per-interface basis is an enhancement feature that allows you to selectively enable or disable link-local signaling for a specific interface regardless of the device 's global OSPF link-local signaling configuration.

When you enable link-local signaling globally in OSPF, it becomes active on all OSPF-enabled interfaces by default. The OSPF link-local signaling per-interface basis feature lets you override the global setting for individual interfaces. You can enable or disable link-local signaling on specific interfaces as needed. For example, disabling link-local signaling on an interface connected to a non-Cisco device, that may be noncompliant with RFC 2328, can prevent problems with the forming of OSPF neighbors in the network.

Purpose of link-local signaling

Link-local signaling provides a mechanism to append additional data to OSPF hello and database description packets by using a dedicated link-local signaling data block (a TLV structure). This extra space allows OSPF devices to exchange more information during adjacency formation and maintenance, without changing the base OSPF packet formats. For example, the OSPF Nonstop Forwarding (NSF) Awareness feature uses this functionality to let customer premises equipment (CPE) devices that are NSF-aware help NSF-capable devices provide nonstop forwarding of packets.

Enable or disable link-local signaling on a per interface basis

Perform this procedure to enable or disable link-local signaling on a per-interface basis.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *interface-id*

Example:

```
Device(config)# interface gigabitethernet 1/0/1
```

Enters interface configuration mode, and specifies the Layer 3 interface to configure.

Step 4 **ip address** *ip-address* [*mask*] [**secondary**]

Example:

```
Device(config-if)# ip address 10.2.145.20 255.255.255.0
```

Sets a primary or secondary IP address for an interface.

- *ip-address*: The IPv4 address you want to assign to the interface.
- *mask*: The subnet mask for the IP address.
- **secondary**: (Optional) Specifies that the address is a secondary IP address, allowing the interface to have multiple IP addresses from different subnets.

Step 5 **no ip directed-broadcast**[*access-list-number* | *extended access-list-number*]

Example:

```
Device(config-if)# no ip directed-broadcast
```

Drops directed broadcasts destined for the subnet to which that interface is attached, rather than broadcasting them.

access-list-number | *extended-access-list-number*: (Optional) Allows you to specify an access list to filter which directed broadcasts are affected.

Step 6 **ip ospf message-digest-key** *key-id encryption-type* **md5** *key*

Example:

```
Device(config-if)# ip ospf message-digest-key 100 md5 testing
```

Enables OSPF Message Digest 5 (MD5) algorithm authentication.

- *key-id*: A unique identifier (number) for the key on this interface. The range is from 1 to 255.
- *encryption-type*: The type of encryption
  - 0 means the password is in plain text.
  - 7 means the password is encrypted in Cisco's type 7 encryption.
- **md5**: Specifies that MD5 authentication is being used.
- *key*: The actual key string (password) used for authentication.

Step 7 [**no** | **default**] **ip ospf lls** [**disable**]

Example:

```
Device(config-if)# ip ospf lls disable
```

Enables or disables link-local signaling on an interface, regardless of the global setting.

- **ip ospf lls**: Enables OSPF link-local signaling on the interface (overrides global settings if configured).
- **disable**: Disables link-local signaling on this specific interface, even if link-local signaling is enabled globally.
- **no**: Removes the **ip ospf lls** configuration or disables link-local signaling on the interface.
- **default**: Resets the command to its default state on the interface.

Step 8 **end**

Example:

```
Device(config-if)# end
```

Returns to privileged EXEC mode.

Configuration example for OSPF link-local signaling per-interface basis

These sections provide configuration examples for OSPF link-local signaling per-interface basis.

Example: Configure OSPF link-local signaling per-interface basis

In this example, link-local signalling has been enabled on Ethernet interface 1/0 and disabled on Ethernet interface 2/0:

```
Device> enable
Device# configure terminal
Device(config)# interface Ethernet1/0
Device(config-if)# ip address 10.2.145.2 255.255.255.0
Device(config-if)# no ip directed-broadcast
Device(config-if)# ip ospf message-digest-key 1 md5 testing
Device(config-if)# ip ospf lls
Device(config-if)# exit

Device(config)# interface Ethernet2/0
Device(config-if)# ip address 10.1.145.2 255.255.0.0
Device(config-if)# no ip directed-broadcast
Device(config-if)# ip ospf message-digest-key 1 md5 testing
Device(config-if)# ip ospf lls disable
Device(config-if)# exit

Device(config)# interface Ethernet3/0
Device(config-if)# ip address 10.3.145.2 255.255.255.0
Device(config-if)# no ip directed-broadcast

Device(config-if)# router ospf 1
Device(config-router)# log-adjacency-changes detail
Device(config-router)# area 0 authentication message-digest
Device(config-router)# redistribute connected subnets
Device(config-router)# network 10.0.0.0 0.255.255.255 area 1
Device(config-router)# network 10.2.3.0 0.0.0.255 area 1
```

Example: Verify OSPF link-local signaling per-interface basis configuration

In the following example, the **show ip ospf interface** command has been entered to verify that link-local signaling has been enabled for Ethernet interface 1/0 and disabled for interface Ethernet 2/0

```
Device# show ip ospf interface
    Ethernet1/0 is up, line protocol is up
    Internet Address 10.2.145.2/24, Area 1
    Process ID 1, Router ID 10.22.222.2, Network Type BROADCAST, Cost: 10
    Transmit Delay is 1 sec, State BDR, Priority 1
    Designated Router (ID) 10.2.2.3, Interface address 10.2.145.1
    Backup Designated router (ID) 10.22.222.2, Interface address 10.2.145.2
    Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:00
    ! Supports Link-local Signaling (LLS)
Index 1/1, flood queue length 0
    Next 0x0(0)/0x0(0)
    Last flood scan length is 2, maximum is 8
    Last flood scan time is 0 msec, maximum is 0 msec
    Neighbor Count is 1, Adjacent neighbor count is 1
```

```
    Adjacent with neighbor 10.2.2.3 (Designated Router)
    Suppress hello for 0 neighbor(s)
Ethernet2/0 is up, line protocol is up
    Internet Address 10.1.145.2/16, Area 1
    Process ID 1, Router ID 10.22.222.2, Network Type BROADCAST, Cost: 10
    Transmit Delay is 1 sec, State BDR, Priority 1
    Designated Router (ID) 10.2.2.3, Interface address 10.1.145.1
    Backup Designated router (ID) 10.22.222.2, Interface address 10.1.145.2
    Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:04
    ! Does not support Link-local Signaling (LLS)
Index 2/2, flood queue length 0
    Next 0x0(0)/0x0(0)
    Last flood scan length is 2, maximum is 11
    Last flood scan time is 0 msec, maximum is 0 msec
    Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 45.2.2.3 (Designated Router)
    Suppress hello for 0 neighbor(s)
```

CHAPTER 10

OSPF Stub Router Advertisement

Feature History for OSPF Stub Router Advertisement

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPF Stub Router Advertisement: The OSPF Stub Router Advertisement is an enhancement feature that enables a device running OSPF to temporarily advertise itself as an undesirable path, also known as stub, by setting its OSPF interface metrics to maximum or infinite.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

OSPF Stub Router Advertisement

The OSPF Stub Router Advertisement is an enhancement feature that enables a device running OSPF to temporarily advertise itself as an undesirable path, also known as stub, by setting its OSPF interface metrics to maximum or infinite.

This is useful in two main scenarios:

- Bringing up a new device:

  Prevents immediate OSPF transit traffic through the new device until it is fully operational.

- Graceful shutdown or reload:

  Allows a device to withdraw itself from being a transit path, preventing packet loss when the device is about to go down.

OSPF Stub Router configuration options

This feature provides three configuration options:

- Default route only advertisement

  The device only advertises a default route with a maximum metric. All other routes are withdrawn from OSPF advertisements.

- Maximum metric for all routes

  The device advertises all its routes, but with a maximum metric. This effectively makes the device the least preferred path for all networks.

- Maximum metric for summary and external routes only

  Only summary and external routes (LSA Type 3, 4, and 5) are advertised with a maximum metric.

Startup maximum metric options

This options prevent neighbor routers from immediately sending traffic through a new or reloaded router, allowing time for BGP or other routing tables to fully converge. This prevents packet loss due to incomplete routing information during initial startup.

1. Timer-Based Maximum Metric Advertisement

   When the device starts or reloads, it advertises a maximum metric for a user-configured period (timer). Neighbor devices avoid using this device as a transit path until the timer expires. After timer expiration, the device advertises normal OSPF metrics, and can be used for forwarding traffic.

2. BGP-Dependent Maximum Metric Advertisement

   The device advertises a maximum metric at startup and continues to do so until either:

   - BGP routing tables have converged (detected via BGP protocol events)
   - The default timer expires (600 seconds), whichever comes first

   Ensures that the device does not forward transit traffic until BGP is ready, preventing packet drops. The device advertises normal OSPF metrics.

OSPF maximum metric and graceful shutdown

This enables a device to gracefully withdraw itself from being a transit path in the network before being shut down, reloaded, or taken out of service. This minimizes the risk of dropped packets and service disruptions.

Before shutdown, the device is configured to advertise a maximum metric (infinite cost) for all its OSPF links. Neighboring devices recalculate their OSPF paths and select alternate routes to avoid the device scheduled for shutdown. The device does not receive new transit traffic, so it can finish processing any local traffic before being removed from the network, minimizing packet loss. The device can be powered down, rebooted, or reconfigured without impacting end-to-end connectivity for other devices.

How OSPF Stub Router Advertisement works

Workflow

1. The device advertises a maximum (or infinite) cost for all its OSPF routes.
2. Other devices will avoid using this device as a transit path.
3. Once the device is ready or has completed its shutdown, normal metrics are restored.

Configure OSPF Stub Router Advertisement

These sections provide configuration information on OSPF stub router advertisement.

Note All tasks are optional and should be individually configured.

Configure advertisement on startup

Perform this task to configure advertisement on startup.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.
- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **max-metric router-lsa on-startup** *announce-time*

Example:

```
Device(config-router)# max-metric router-lsa on-startup 100
```

Configures OSPF to advertise a maximum metric during startup for a configured period of time.

The *announce-time* argument is a configurable timer that must follow the on-startup keyword to be configured. The configurable time range is from 5 to 86,400 seconds. There is no default timer value.

Step 5 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

Configure advertisement until routing tables converge

Perform this task to configure advertisement until routing tables converge.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.
- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **max-metric router-lsa on-startup wait-for-bgp**

Example:

```
Device(config-router)# max-metric router-lsa on-startup 100
```

Configures OSPF to advertise a maximum metric until BGP routing tables have converged or until the default timer has expired. The **wait-for-bgp** keyword must follow the **on-startup** keyword to be configured. The default timer value is 600 seconds.

Step 5 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

Configure advertisement for a graceful shutdown

Perform this task to configure advertisement for a graceful shutdown.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.
- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **max-metric router-lsa**

Example:

```
Device(config-router)# max-metric router-lsa
```

Configures OSPF to advertise a maximum metric until the router is shut down.

Step 5 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

Step 6 **show ip ospf**

Example:

```
Device# show ip ospf
```

Use the **show ip ospf** command to verify that the **max-metric router-lsa** command has been enabled before the device is shut down or reloaded.

Configuration examples of OSPF Stub Router Advertisement

This section provides configuration example of OSPF stub router advertisement.

Example: Advertisement on startup

In the following example, a device that is running OSPF is configured to advertise a maximum metric at startup for 300 seconds:

```
Device> enable
Device# configure terminal
Device(config)# router ospf 100
Device(config-router)# max-metric router-lsa on-startup 300
Device(config-router)# end
```

Example: Advertisement until routing tables converge

In the following example, a router that is running OSPF is configured to advertise a maximum metric until BGP routing tables converge or until the default timer expires (600 seconds):

```
Device> enable
Device# configure terminal
```

```
Device(config)# router ospf 100
Device(config-router)# max-metric router-lsa on-startup wait-for-bgp
Device(config-router)# end
```

Example: Graceful shutdown

In the following example, a device that is running OSPF is configured to advertise a maximum metric until the device is shut down:

```
Device> enable
Device# configure terminal
Device(config)# router ospf 100
Device(config-router)# max-metric router-lsa
Device(config-router)# end
```

Example: Verify the advertisement of a maximum metric

To verify that the advertisement of a maximum metric has been configured correctly, use the **show ip ospf** or **show ip ospf database** command.

The output of the **show ip ospf** command will display the condition, state, and remaining time delay of the advertisement of a maximum metric, depending on which options were configured with the **max-metric router-lsa** command.

The following sample output is similar to the output that will be displayed when the **on-startup** keyword and *announce-time* argument are configured with the **max-metric router-lsa** command:

```
Device# show ip ospf
Routing Process "ospf 1998" with ID 10.18.134.155
     Supports only single TOS(TOS0) routes
     Supports opaque LSA
     It is an area border and autonomous system boundary router
     Redistributing External Routes from,
        static, includes subnets in redistribution
     Originating router-LSAs with maximum metric, Time remaining: 00:01:18
        Condition: on startup for 300 seconds, State: active
     SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
     Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
     Number of external LSA 7. Checksum Sum 0x47261
     Number of opaque AS LSA 0. Checksum Sum 0x0
     Number of DCbitless external and opaque AS LSA 0
     Number of DoNotAge external and opaque AS LSA 0
     Number of areas in this router is 2. 1 normal 0 stub 1 nssa
     External flood list length 0
        Area BACKBONE(0)
           Number of interfaces in this area is 1
           Area has no authentication
           SPF algorithm executed 3 times
           Area ranges are
           Number of LSA 8. Checksum Sum 0x474AE
           Number of opaque link LSA 0. Checksum Sum 0x0
```

The following sample output is similar to the output that will be displayed when the **on-startup** and **wait-for-bgp** keywords are configured with the **max-metric router-lsa** command:

```
Device# show ip ospf
Routing Process "ospf 1998" with ID 10.18.134.155
     Supports only single TOS(TOS0) routes
     Supports opaque LSA
     It is an area border and autonomous system boundary router
```

```
     Redistributing External Routes from,
        static, includes subnets in redistribution
     Originating router-LSAs with maximum metric, Time remaining: 00:01:18
        Condition: on startup while BGP is converging, State: active
     SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
     Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
     Number of external LSA 7. Checksum Sum 0x47261
     Number of opaque AS LSA 0. Checksum Sum 0x0
     Number of DCbitless external and opaque AS LSA 0
     Number of DoNotAge external and opaque AS LSA 0
     Number of areas in this router is 2. 1 normal 0 stub 1 nssa
     External flood list length 0
        Area BACKBONE(0)
           Number of interfaces in this area is 1
           Area has no authentication
           SPF algorithm executed 3 times
           Area ranges are
           Number of LSA 8. Checksum Sum 0x474AE
           Number of opaque link LSA 0. Checksum Sum 0x0
```

The following sample output is similar to the output that will be displayed when the **max-metric router-lsa** command is configured without any keywords or arguments:

```
Device# show ip ospf
Routing Process "ospf 1998" with ID 10.18.134.155
     Supports only single TOS(TOS0) routes
     Supports opaque LSA
     It is an area border and autonomous system boundary router
     Redistributing External Routes from,
        static, includes subnets in redistribution
     Originating router-LSAs with maximum metric
        Condition: always, State: active
     SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
     Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
     Number of external LSA 7. Checksum Sum 0x47261
     Number of opaque AS LSA 0. Checksum Sum 0x0
     Number of DCbitless external and opaque AS LSA 0
     Number of DoNotAge external and opaque AS LSA 0
     Number of areas in this router is 2. 1 normal 0 stub 1 nssa
     External flood list length 0
        Area BACKBONE(0)
           Number of interfaces in this area is 1
           Area has no authentication
           SPF algorithm executed 3 times
           Area ranges are
           Number of LSA 8. Checksum Sum 0x474AE
           Number of opaque link LSA 0. Checksum Sum 0x0
```

The output of the **show ip ospf database** command will display information about OSPF LSAs and indicate if the router is announcing maximum cost links. The following sample output is similar to the output that will be displayed when any form of the **max-metric router-lsa** command is configured:

```
Device# show ip ospf database
      Exception Flag: Announcing maximum link costs
      LS age: 68
      Options: (No TOS-capability, DC)
      LS Type: Router Links
      Link State ID: 172.18.134.155
      Advertising Router: 172.18.134.155
      LS Seq Number: 80000002
      Checksum: 0x175D
      Length: 60
      Area Border Router
```

```
AS Boundary Router
Number of Links: 3

  Link connected to: a Transit Network
  (Link ID) Designated Router address: 192.168.1.11
  (Link Data) Router Interface address: 192.168.1.14
   Number of TOS metrics: 0
    TOS 0 Metrics: 65535 (metric used for local calculation: 10)

  Link connected to: a Transit Network
  (Link ID) Designated Router address: 10.1.145.11
  (Link Data) Router Interface address: 10.1.145.14
   Number of TOS metrics: 0
    TOS 0 Metrics: 65535 (metric used for local calculation: 10)

  Link connected to: a Stub Network
  (Link ID) Network/subnet number: 10.11.12.0
  (Link Data) Network Mask: 255.255.255.0
   Number of TOS metrics: 0
    TOS 0 Metrics: 1
```

Monitor and maintain OSPF Stub Router Advertisement

Command	Purpose
show ip ospf	Displays general information about OSPF routing processes and provides information about the configuration settings and status of the OSPF Stub Router Advertisement feature.
show ip ospf database router	Displays information about device LSAs, and indicates if a device is announcing maximum link costs.

CHAPTER 11

OSPF Update Packet-Pacing Configurable Timers

Feature History for OSPF Update Packet-Pacing Configurable Timers

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPF Update Packet-Pacing Configurable Timers: OSPF Update Packet-Pacing Configurable Timers is an enhancement feature that allows to configure the rate at which OSPF LSA flood pacing, retransmission pacing, and group pacing updates occur.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

OSPF Update Packet-Pacing Configurable Timers

OSPF Update Packet-Pacing Configurable Timers is an enhancement feature that allows to configure the rate at which Open Shortest Path First (OSPF) link-state advertisement (LSA) flood pacing, retransmission pacing, and group pacing updates occur. In rare situations, you might need to change OSPF packet-pacing default timers to mitigate CPU or buffer utilization issues associated with flooding very large numbers of LSAs.

Types of OSPF Packet-Pacing Configurable Timers

There are three types of OSPF packet-pacing configurable timers available:

- OSPF Flood Pacing Timers:

  These timers control the inter-packet spacing between consecutive link-state update packets as they are sent out from the OSPF transmission queue.

  By spacing out the flood of new LSAs, the device can manage its resources more effectively. This approach prevents a sudden surge of traffic that could overwhelm the CPU or buffers.

- OSPF Retransmission Pacing Timers:

  These timers govern the inter-packet spacing between consecutive link-state update packets that are being retransmitted from the OSPF retransmission queue.

  Similar to flood pacing, this helps prevent a device from being overwhelmed by retransmissions, especially in unstable network environments where LSAs might be lost and need to be resent. It ensures a more controlled and gradual retransmission process.

- OSPF Group Pacing Timers (Cisco IOS XE Specific):

  Cisco IOS XE software introduces this feature to group the periodic refresh of LSAs. This grouping aims to improve the LSA packing density. More LSAs can be sent in fewer packets during refreshes, which is beneficial in large network topologies.

Guidelines to configure OSPF Update Packet-Pacing Configurable Timers

Do not change the packet-pacing timers unless all other options to meet OSPF packet flooding requirements have been exhausted. Network operators should prefer summarization, stub area usage, queue tuning, and buffer tuning before changing the default timers.

There are no guidelines for changing timer values because each OSPF deployment is unique and should be considered on a case-by-case basis. The network operator assumes the risks associated with changing the default timer values.

How to configure OSPF Packet-Pacing Timers

This section provides information on configuring OSPF packet-pacing timers.

Configure OSPF Packet-Pacing Timers

Perform this task to configure OSPF packet-pacing timers.

Before you begin

Caution The default settings for OSPF packet-pacing timers are suitable for the majority of OSPF deployments. You should change the default timers only as a last resort.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **timers pacing flood** *milliseconds*

Example:

```
Device(config-router)# timers pacing flood 15
```

Configures a flood packet-pacing timer delay, in milliseconds.

Step 5 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

(Optional) Configure a Retransmission Packet-Pacing Timer

Perform this task to configure a retransmission packet-pacing timer.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **timers pacing retransmission** *milliseconds*

Example:

```
Device(config-router)# timers pacing retransmission 15
```

Configures a retransmission packet-pacing timer delay, in milliseconds.

Step 5 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

(Optional) Configure a Group Packet-Pacing Timer

Perform this task to configure a group packet-pacing timer.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **timers pacing lsa-group** *seconds*

Example:

```
Device(config-router)# timers pacing lsa-group 15
```

Configures an LSA group packet-pacing timer delay, in milliseconds.

Step 5 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

Configuration examples of OSPF Update Packet-Pacing Timers

This section provides configuration examples of OSPF update packet-pacing timers.

Example: LSA Flood Pacing

The following example configures LSA flood pacing updates to occur in 50-millisecond intervals for OSPF routing process 1:

```
Device> enable
Device# configure terminal
Device(config)# router ospf 1
Device(config-router)# timers pacing flood 50
```

Example: LSA Retransmission Pacing Timer

The following example configures LSA retransmission pacing updates to occur in 100-millisecond intervals for OSPF routing process 1:

```
Device> enable
Device# configure terminal
Device(config)# router ospf 1
Device(config-router)# timers pacing retransmission 100
```

Example: LSA Group Pacing

The following example configures OSPF group pacing updates between LSA groups to occur in 75-second intervals for OSPF routing process 1:

```
Device> enable
Device# configure terminal
Device(config)# router ospf 1
Device(config-router)# timers pacing lsa-group 75
```

Verify OSPF Packet-Pacing Timers

To verify that OSPF packet pacing has been configured, use the **show ip ospf** privileged EXEC command. The output of the **show ip ospf** command will display the type and delay time of the configurable pacing timers (flood, retransmission, group).

The following sample output is from the **show ip ospf** command:

```
Router# show ip ospf
  Routing Process "ospf 1" with ID 10.0.0.1 and Domain ID 10.20.0.1
  Supports only single TOS(TOS0) routes
  Supports opaque LSA
  SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
  Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
  LSA group pacing timer 100 secs
  Interface flood pacing timer 55 msecs
  Retransmission pacing timer 100 msecs
  Number of external LSA 0. Checksum Sum 0x0
  Number of opaque AS LSA 0. Checksum Sum 0x0
  Number of DCbitless external and opaque AS LSA 0
```

```
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 2. 2 normal 0 stub 0 nssa
External flood list length 0
   Area BACKBONE(0)
       Number of interfaces in this area is 2
       Area has message digest authentication
       SPF algorithm executed 4 times
       Area ranges are
       Number of LSA 4. Checksum Sum 0x29BEB
       Number of opaque link LSA 0. Checksum Sum 0x0
       Number of DCbitless LSA 3
       Number of indication LSA 0
       Number of DoNotAge LSA 0
       Flood list length 0
   Area 172.16.26.0
       Number of interfaces in this area is 0
       Area has no authentication
       SPF algorithm executed 1 times
       Area ranges are
          192.168.0.0/16 Passive Advertise
       Number of LSA 1. Checksum Sum 0x44FD
       Number of opaque link LSA 0. Checksum Sum 0x0
       Number of DCbitless LSA 1
       Number of indication LSA 1
       Number of DoNotAge LSA 0
       Flood list length 0
```

Monitor and maintain OSPF Packet-Pacing Timers

Command	Purpose
show ip ospf	Displays general information about OSPF routing processes.
show ip ospf neighbor	Displays OSPF neighbor information on a per-interface basis.
clear ip ospf redistribution	Clears route redistribution based on the OSPF routing process ID.

CHAPTER 12

OSPFv2 NSSA Option

Feature History for OSPFv2 NSSA

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPFv2 NSSA: The OSPFv2 NSSA is a network optimization feature that allows a central site that runs on OSPF protocol to connect to a remote site that runs on any other routing protocol.	Cisco C9610 Series Smart Switches

OSPFv2 NSSA

The OSPFv2 NSSA is a network optimization feature that allows a central site that runs on OSPF protocol to connect to a remote site that runs on any other routing protocol.

Without OSPFv2 NSSA

Without OSPFv2 NSSA, the following happens:

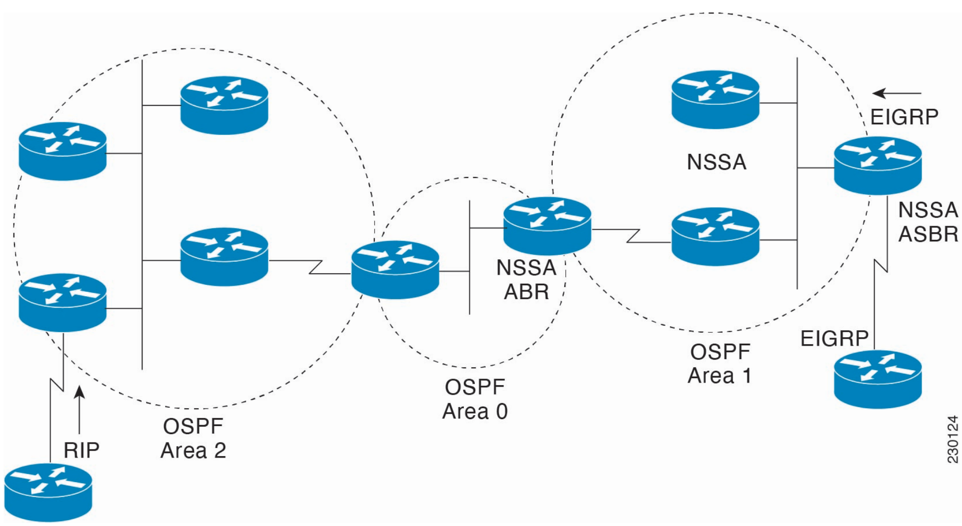
- OSPF stub areas do not allow the redistribution from other protocols like RIP (Routing Information Protocol).
- You must maintain two routing protocols: OSPF in the core and RIP at the edge.
- The stub area cannot import external routes from the remote network.

Key components in OSPFv2 NSSA

- Autonomous System Border Router (ASBR): ASBRs connect an OSPF network and a non-OSPF network (or another OSPF domain if routes are being explicitly redistributed). The ASBR redistributes routes from RIP (or another protocol) into OSPF NSSA as type 7 LSAs.
- Area Border Router (ABR): ABRs connect regular areas to the backbone (Area 0). An ABR converts type 7 LSAs to type 5 LSAs, allowing the rest of the OSPF domain to learn about the external routes.
- Route summarization: Route summarization is the process of combining a group of contiguous network addresses into a single summary address. This reduces the size of routing tables and improves routing efficiency. In OSPF, summarization is performed at Area Border Routers (ABRs) and Autonomous System Boundary Routers (ASBRs). If an OSPF area contains multiple contiguous subnets, the ABR can be configured to advertise a single summary route to other areas instead of multiple individual routes.

How OSPFv2 NSSA works

Figure 1: OSPF NSSA



The figure shows a network diagram that contains

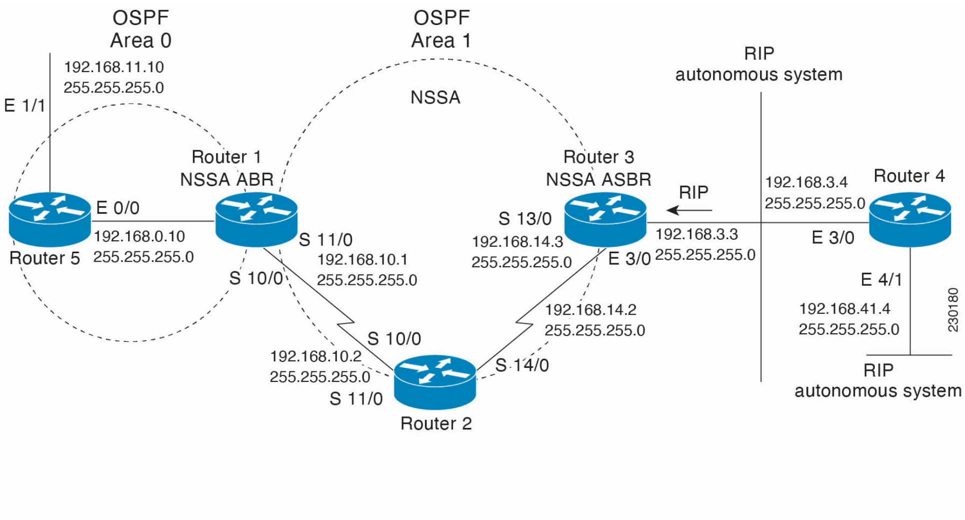
- OSPF Area 0 (Backbone Area): This is the central OSPF area, acting as the backbone that connects other OSPF areas. It contains at least two devices, one of which is an NSSA ABR.
- OSPF Area 2: This area connects to OSPF Area 0 via an ABR (Area Border Router). It also connects to an external network running RIP (Routing Information Protocol).

- OSPF Area 1 (NSSA - Not-So-Stubby Area): This is a special type of OSPF area. It connects to OSPF Area 0 via an NSSA ABR and also connects to an external network running EIGRP (Enhanced Interior Gateway Routing Protocol) via an NSSA ASBR.
- ABR (Area Border Router): Devices that connect one OSPF area to another (e.g., the router between OSPF Area 2 and OSPF Area 0, and the NSSA ABR between OSPF Area 0 and OSPF Area 1). ABRs maintain separate link-state databases for each area they are connected to. They also summarize routes between areas.
- NSSA ABR (Not-So-Stubby Area ABR): This ABR connects the NSSA (Area 1) to the backbone (Area 0). It is responsible for translating Type 7 LSAs (generated by the NSSA ASBR for external routes) into Type 5 LSAs, which are then flooded into the backbone area.
- NSSA ASBR (Autonomous System Boundary Router): This switch is located within the NSSA (Area 1) and connects the OSPF domain to an external routing domain (EIGRP in this case). It imports external EIGRP routes into OSPF as Type 7 LSAs.
- IP (Routing Information Protocol): An external network running RIP is connected to OSPF Area 2. Routes from this RIP domain would be redistributed into OSPF by the ASBR (which is also an ABR in this case) connecting Area 2 to the RIP network.
- EIGRP (Enhanced Interior Gateway Routing Protocol): An external network running EIGRP is connected to OSPF Area 1 via the NSSA ASBR. This NSSA ASBR redistributes EIGRP routes into OSPF as Type 7 LSAs within Area 1.

How it works

- Devices within each OSPF area exchange link-state information and build their respective topological databases.
- ABRs summarize routes between areas. For example, Area 2 routes are summarized and sent to Area 0, and Area 0 routes are sent to Area 2.
- Routes from the RIP domain are imported into OSPF Area 2 and then propagated through the OSPF backbone.
- EIGRP routes are imported by the NSSA ASBR into OSPF Area 1 as Type 7 LSAs. The NSSA ABR then converts these Type 7 LSAs into Type 5 LSAs, which are then flooded into OSPF Area 0 (the backbone) and subsequently to other areas if needed. This mechanism allows external routes to be injected into a stub-like area without making it a full external area.

Figure 2: OSPF NSSA Network with NSSA ABR and ASBR Devices



The following table shows information about the above figure.

OSPF Area 0 (Backbone Area)	Switch 5	• Connects to OSPF Area 0. • Interface E 1/1 has IP address 192.168.11.10 with a subnet mask of 255.255.255.0. • Interface E 0/0 connects to Switch 1 with IP address 192.168.0.10/24.
	Switch 1 (NSSA ABR - Not-So-Stubby Area Area Border Router)	• Acts as an Area Border Router, connecting OSPF Area 0 to OSPF Area 1. • Interface E 0/0 connects to Switch 5 with IP address 192.168.0.10/24. • Interfaces S 10/0 (192.168.10.1/24) and S 11/0 (implied, but shares the 192.168.10.1/24 network with S 10/0 on Switch 2) connect to Switch 2, extending into OSPF Area 1.

OSPF Area 1 (NSSA - Not-So-Stubby Area)	Switch 2	• Part of OSPF Area 1. • Interface S 11/0 connects to Switch 1 with IP address 192.168.10.2/24. • Interface S 14/0 connects to Switch 3 with IP address 192.168.14.2/24.
	Switch 3 (NSSA ASBR - Not-So-Stubby Area Autonomous System Boundary Router)	• Acts as an Autonomous System Boundary Router, connecting OSPF Area 1 to the RIP autonomous system. • Interface S 13/0 connects to Switch 2 with IP address 192.168.14.3/24. • Interface E 3/0 connects to Switch 4, participating in both OSPF (within Area 1) and RIP (with the RIP autonomous system) with IP address 192.168.3.3/24.
RIP Autonomous System	Switch 4	• Part of the RIP autonomous system. • Interface E 3/0 connects to Switch 3 with IP address 192.168.3.4/24. • Interface E 4/1 has IP address 192.168.41.4/24, representing another network within the RIP domain.

How it works

- Devices within each OSPF area exchange link-state information and build their respective topological databases.
- RIP routes are imported by the NSSA ASBR (Switch 2) into OSPF Area 1 as Type 7 LSAs. The NSSA ABR (Switch 2) then converts these Type 7 LSAs into Type 5 LSAs, which are then flooded into OSPF Area 0 (the backbone). This mechanism allows external routes to be injected into a stub-like area without making it a full external area.

How to configure OSPFv2 NSSA

This section provides information on how to configure OSPFv2 NSSA.

Configure an OSPFv2 NSSA area and its parameters

Perform this procedure to configure an OSPFv2 NSSA area and it parameters.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **redistribute** *protocol* [*process-id*] [*as-number*] [**include-connected** {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value* | **transparent**] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**subnets**] [**nssa-only**] [**tag** *tag-value*] [**route-map** *map-tag*]

Example:

```
Device(config-router)# redistribute rip subnets
```

Redistributes routes from one routing domain into another routing domain.

- *protocol*: Specifies the source of the routes to be redistributed. Common options include:
  - **connected**: Redistributes directly connected networks that are not already part of the OSPF process.
  - **static**: Redistributes static routes configured on the router.
  - **bgp**: Redistributes routes learned via BGP.
  - **eigrp**: Redistributes routes learned via EIGRP.
  - **rip**: Redistributes routes learned via RIP.
  - **ospf**: Redistributes routes from another OSPF process (e.g., if running multiple OSPF instances or different process IDs).
  - **isis**: Redistributes routes learned via IS-IS.
- *process-id*: (Optional) If the source protocol (e.g., EIGRP, OSPF, RIP) uses a process ID, you specify it here to identify the specific instance from which routes should be taken.

- *as-number*: (Optional) If the source protocol (e.g., BGP or EIGRP) uses an Autonomous System (AS) number, you specify it here. This is often used interchangeably with process-id for protocols that use AS numbers.
- **include-connected** {**level-1** | **level-1-2** | **level-2**}: (Optional) This parameter is typically specific to IS-IS redistribution into OSPF, or sometimes OSPF into IS-IS, or even OSPF into OSPF (though less common). It refers to the IS-IS routing levels:
    - level-1: Redistributes Level-1 IS-IS routes.
    - level-2: Redistributes Level-2 IS-IS routes.
    - level-1-2: Redistributes both Level-1 and Level-2 IS-IS routes.
    - This part of the syntax suggests a context where IS-IS is the source protocol.
- **metric** {*metric-value* | **transparent**}: This is a critical parameter for redistribution. When routes are redistributed from one protocol to another, they need to be assigned a metric that is meaningful to the destination protocol.
    - *metric-value*: A specific numerical value to assign as the metric. The meaning of this value depends on the destination protocol (e.g., hop count for RIP, cost for OSPF).
    - **transparent**: This attempts to carry the original metric information across, but its effectiveness can vary depending on the protocols involved.
- **metric-type** *type-value*: (Optional) For OSPF, this is crucial for how external routes are handled:
    - type-1 (E1): The cost to the external destination is the sum of the external metric and the internal OSPF cost to reach the ASBR. This is often preferred for more accurate path selection.
    - type-2 (E2): The cost to the external destination is only the external metric, regardless of the internal OSPF cost to reach the ASBR. This is the default if not specified. Type 2 routes are always preferred over Type 1 routes if their external metric is the same.
- **match** {**internal** | **external 1** | **external 2**}: This parameter is specific to OSPF when redistributing OSPF routes into another protocol. It allows you to specify which types of OSPF routes to redistribute:
    - **internal**: Redistributes OSPF intra-area (Type 1) and inter-area (Type 2) routes.
    - **external 1**: Redistributes OSPF external Type 1 routes.
    - **external 2**: Redistributes OSPF external Type 2 routes.
- **nssa-only**: (Optional) This keyword is specifically used when redistributing routes into a Not-So-Stubby Area (NSSA). When **nssa-only** is specified, the redistributed routes will be advertised as Type 7 LSAs within the NSSA, and then translated to Type 5 LSAs by the NSSA ABR for propagation to other areas. Without this parameter, redistribution into an NSSA might not be allowed or might behave differently
- **tag** *tag-value*: (Optional) Assigns a numerical tag to the redistributed routes. This tag can be used later in route maps or distribute lists for filtering or policy-based routing.
- **route-map** *map-tag*: (Optional) A very powerful and commonly used parameter. It references a `route-map` that provides granular control over which routes are redistributed and how their attributes (like metric, metric-type, tag) are modified during the redistribution process.

- **subnets**: This parameter is specific to OSPF. By default, OSPF only redistributes classful networks. To redistribute subnetted networks (which is almost always desired in modern networks), you must include the **subnets** keyword. Without it, only major network numbers (e.g., 10.0.0.0/8, 192.168.1.0/24) would be redistributed.

Step 5 **network** *ip-address wildcard-mask* **area** *area-id*

Example:

```
Device(config-router)# redistribute rip subnets
```

Defines the interfaces on which OSPF runs and defines the area ID for those interfaces.

- *ip-address*: The network address of the interface you want to enable OSPF on.
- *wildcard-mask*: Inverse mask that, together with the ip-address, selects a range of IP addresses.
- *area-id*: The OSPF area (in decimal or dotted-decimal format) to which the interface will be assigned.

Step 6 **area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate**] [**metric** *metric-type*] [**no-summary**] [**nssa-only**]

Example:

```
Device(config-router)# area 1 nssa
```

(Optional) Defines an area as a NSSA. Every device within the same area must agree that the area is NSSA. Select one of these keywords:

- **no-redistribution**: Select when the router is an NSSA ABR and you want the redistribute command to import routes into normal areas, but not into the NSSA.
- **default-information-originate**: Select on an ABR to import type 7 LSAs into the NSSA.
- **metric** *metric-type*: (Optional) Sets the metric for the default route.

  (Optional) Specifies the type (1 or 2) for the default route.
- **no-summary**: Select to not send summary LSAs into the NSSA.
- **nssa**: Disallows the area from functioning as a normal area if the NSSA configuration is removed (rarely used in practice).

Step 7 **summary-address** *prefix-mask* [**not-advertise**] [**tag** *tag*] [**nssa-only**]

Example:

```
Device(config-router)# summary-address 10.1.0.0 255.255.0.0 not-advertise
```

Controls the route summarization and filtering during the translation and limits the summary to NSSA areas

- *prefix-mask*: The summarized network address and the network mask for the summary route.
- **not-advertise**: (Optional) Suppresses the summary route advertisement.
- **tag** *tag*: (Optional) Associates a tag with the summary route for policy or routing decisions.
- **nssa-only**: (Optional) Advertises the summary only as a Type 7 LSA for NSSA areas.

Step 8 **end**

Example:

```
Device(config-router)# end
```

Exits router configuration mode and returns to privileged EXEC mode.

Configure an NSSA ABR as a forced NSSA LSA translator

Perform this procedure to configure an NSSA ABR as a forced NSSA LSA translator.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.

- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **area** *area-id* **nssa translate type7 always**

Example:

```
Device(config-router)# area 10 nssa translate type7 always
```

Configures a NSSA ABR device as a forced NSSA LSA translator.

Note
This command can be used if RFC 3101 is disabled and RFC 1587 is used.

- *area-id*: The OSPF area number or address in dotted-decimal format.

- **always**: Manually force this ABR to always be the translator, ensuring consistent LSA translation and avoiding translator role changes due to device restarts or network topology changes.

Step 5 **area** *area-id* **nssa translate type7 suppress-fa**

Example:

```
Device(config-router)# area 10 nssa translate type7 suppress-fa
```

Allows ABR to suppress the forwarding address in translated Type-5 LSA.

Step 6 **end**

Example:

```
Device(config-router)# end
```

Exits router configuration mode and returns to privileged EXEC mode.

Disable RFC 3101 compatibility and enable RFC 1587 compatibility

Perform this task to disable RFC 3101 compatibility and enable RFC 1587 compatibility.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **compatible rfc1587**

Example:

```
Device(config-router)# compatible rfc1587
```

Enables the device to be RFC 1587 compliant.

Step 5 **end**

Example:

```
Device(config-router)# end
```

Enables the device to be RFC 1587 compliant.

Configuration examples for OSPFv2 NSSA

In the following example, an OSPF stub network is configured to include OSPF Area 0 and OSPF Area 1, using five devices. Device 3 is configured as the NSSA ASBR. Device 2 configured to be the NSSA ABR. OSPF Area 1 is defined as a NSSA.

Device 1

```
Device# hostname Device1
!
interface Loopback1
    ip address 10.1.0.1 255.255.255.255
!
interface Ethernet0/0
    ip address 192.168.0.1 255.255.255.0
    ip ospf 1 area 0
    no cdp enable
!
interface Serial10/0
    description Device2 interface s11/0
    ip address 192.168.10.1 255.255.255.0
    ip ospf 1 area 1
    serial restart-delay 0
    no cdp enable
!
router ospf 1
    area 1 nssa
!
end
```

Device 2

```
Device# hostname Device2
!!
interface Loopback1
    ip address 10.1.0.2 255.255.255.255
!
    interface Serial10/0
    description Device1 interface s11/0
    no ip address
    shutdown
    serial restart-delay 0
    no cdp enable
```

```
!
interface Serial11/0
    description Device1 interface s10/0
    ip address 192.168.10.2 255.255.255.0
    ip ospf 1 area 1
    serial restart-delay 0
    no cdp enable
!
interface Serial14/0
    description Device3 interface s13/0
    ip address 192.168.14.2 255.255.255.0
    ip ospf 1 area 1
    serial restart-delay 0
    no cdp enable
!
router ospf 1
    area 1 nssa
!
end
```

Device 3

```
Device# hostname Device3
!
interface Loopback1
    ip address 10.1.0.3 255.255.255.255
!
interface Ethernet3/0
    ip address 192.168.3.3 255.255.255.0
    no cdp enable
!
interface Serial13/0
    description Device2 interface s14/0
    ip address 192.168.14.3 255.255.255.0
    ip ospf 1 area 1
    serial restart-delay 0
    no cdp enable
!
router ospf 1
    log-adjacency-changes
    area 1 nssa
    redistribute rip subnets
!
router rip
    version 2
    redistribute ospf 1 metric 15
    network 192.168.3.0
end
```

Device 4

```
Device# hostname Device4
!
interface Loopback1
    ip address 10.1.0.4 255.255.255.255
!
interface Ethernet3/0
    ip address 192.168.3.4 255.255.255.0
    no cdp enable
!
interface Ethernet4/1
    ip address 192.168.41.4 255.255.255.0
!
    router rip
    version 2
```

```
    network 192.168.3.0
    network 192.168.41.0
!
end
```

Device 5

```
Device# hostname Device5
!
interface Loopback1
    ip address 10.1.0.5 255.255.255.255
!
interface Ethernet0/0
    ip address 192.168.0.10 255.255.255.0
    ip ospf 1 area 0
    no cdp enable
!
interface Ethernet1/1
    ip address 192.168.11.10 255.255.255.0
    ip ospf 1 area 0
!
router ospf 1
!
end
```

Example: Verify OSPF NSSA

The following is sample output from the **show ip ospf** command. The output displays that OSPF Area 1 is an NSSA area.

```
Device2#show ip ospf

 Routing Process "ospf 1" with ID 10.1.0.2
 Start time: 00:00:01.392, Time elapsed: 12:03:09.480
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Supports Link-local Signaling (LLS)
 Supports area transit capability
 Router is not originating router-LSAs with maximum metric
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Incremental-SPF disabled
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 1. 0 normal 0 stub 1 nssa
 Number of areas transit capable is 0
 External flood list length 0
    Area 1
  Number of interfaces in this area is 2
! It is a NSSA area
  Area has no authentication
  SPF algorithm last executed 11:37:58.836 ago
  SPF algorithm executed 3 times
  Area ranges are
```

```
  Number of LSA 7. Checksum Sum 0x045598
  Number of opaque link LSA 0. Checksum Sum 0x000000
  Number of DCbitless LSA 0
  Number of indication LSA 0
  Number of DoNotAge LSA 0
  Flood list length 0


Device2#show ip ospf data

            OSPF Router with ID (10.1.0.2) (Process ID 1)
  Router Link States (Area 1)
Link ID         ADV Router      Age         Seq#       Checksum Link count
10.1.0.1        10.1.0.1        1990        0x80000016 0x00CBCB 2
10.1.0.2        10.1.0.2        1753        0x80000016 0x009371 4
10.1.0.3        10.1.0.3        1903        0x80000016 0x004149 2

  Summary Net Link States (Area 1)
Link ID         ADV Router      Age         Seq#       Checksum
192.168.0.0     10.1.0.1        1990        0x80000017 0x00A605
192.168.11.0    10.1.0.1        1990        0x80000015 0x009503

  Type-7 AS External Link States (Area 1)
Link ID         ADV Router      Age         Seq#       Checksum Tag
192.168.3.0     10.1.0.3        1903        0x80000015 0x00484F 0
192.168.41.0    10.1.0.3        1903        0x80000015 0x00A4CC 0
```

The following is sample output from the **show ip ospf database data** command. The output displays additional information about redistribution between Type 5 and Type 7 LSAs for routes that are injected into the NSSA area and then flooded through the OSPF network.

```
Device2#show ip ospf database data

            OSPF Router with ID (10.1.0.2) (Process ID 1)
Area 1 database summary
  LSA Type      Count    Delete   Maxage
  Router        3        0        0
  Network       0        0        0
  Summary Net   2        0        0
  Summary ASBR  0        0        0
  Type-7 Ext    2        0        0

Prefixes redistributed in Type-7  0
  Opaque Link   0        0        0
  Opaque Area   0        0        0
  Subtotal      7        0        0

Process 1 database summary
  LSA Type      Count    Delete   Maxage
  Router        3        0        0
  Network       0        0        0
  Summary Net   2        0        0
  Summary ASBR  0        0        0
  Type-7 Ext    2        0        0
  Opaque Link   0        0        0
  Opaque Area   0        0        0
  Type-5 Ext    0        0        0

Prefixes redistributed in Type-5  0
  Opaque AS     0        0        0
  Total         7        0        0
```

The following is sample output from the **show ip ospf database nssa** command. The output displays detailed information for Type 7 to Type 5 translations:

```
Device2#show ip ospf database nssa

  OSPF Router with ID (10.1.0.2) (Process ID 1)
  Type-7 AS External Link States (Area 1)
  Routing Bit Set on this LSA
  LS age: 1903
  Options: (No TOS-capability, Type 7/5 translation, DC)
  LS Type: AS External Link
  Link State ID: 192.168.3.0 (External Network Number )
  Advertising Router: 10.1.0.3
  LS Seq Number: 80000015
  Checksum: 0x484F
  Length: 36
  Network Mask: /24
  Metric Type: 2 (Larger than any link state path)
  TOS: 0
  Metric: 20
  Forward Address: 192.168.14.3
  External Route Tag: 0
  Routing Bit Set on this LSA
  LS age: 1903
! Options: (No TOS-capability, Type 7/5 translation, DC)
  LS Type: AS External Link
  Link State ID: 192.168.41.0 (External Network Number )
  Advertising Router: 10.1.0.3
  LS Seq Number: 80000015
  Checksum: 0xA4CC
  Length: 36
  Network Mask: /24
  Metric Type: 2 (Larger than any link state path)
  TOS: 0
  Metric: 20
  Forward Address: 192.168.14.3
  External Route Tag: 0
```

The following sample output from the **show ip ospf** command displays that the device is acting as an ASBR and OSPF Area 1 is configured as an NSSA area:

```
Device3#show ip ospf

 Routing Process "ospf 1" with ID 10.1.0.3
 Start time: 00:00:01.392, Time elapsed: 12:02:34.572
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Supports Link-local Signaling (LLS)
 Supports area transit capability
!It is an autonomous system boundary router
 Redistributing External Routes from,
    rip, includes subnets in redistribution
 Router is not originating router-LSAs with maximum metric
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Incremental-SPF disabled
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
```

```
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 1. 0 normal 0 stub 1 nssa
 Number of areas transit capable is 0
 External flood list length 0
    Area 1
  Number of interfaces in this area is 1
! It is a NSSA area
  Area has no authentication
  SPF algorithm last executed 11:38:13.368 ago
  SPF algorithm executed 3 times
  Area ranges are
  Number of LSA 7. Checksum Sum 0x050CF7
  Number of opaque link LSA 0. Checksum Sum 0x000000
  Number of DCbitless LSA 0
  Number of indication LSA 0
  Number of DoNotAge LSA 0
  Flood list length 0
```

The table below describes the significant fields shown in the **show ip ospf** command output.

Table 1: show ip ospf Field Descriptions

Field	Description
Routing process "ospf 1" with ID 10.1.0.3	Process ID and OSPF router ID.
Supports ...	Number of types of service supported (Type 0 only).
Summary Link update interval	Specifies summary update interval in hours:minutes:seconds, and time until next update.
External Link update interval	Specifies external update interval in hours:minutes:seconds, and time until next update.
Redistributing External Routes from	Lists of redistributed routes, by protocol.
SPF calculations	Lists start, hold, and maximum wait interval values in milliseconds.
Number of areas	Number of areas in router, area addresses, and so on.
SPF algorithm last executed	Shows the last time an SPF calculation was performed in response to topology change event records.
Link State Update Interval	Specifies router and network link-state update interval in hours:minutes:seconds, and time until next update.
Link State Age Interval	Specifies max-aged update deletion interval, and time until next database cleanup, in hours:minutes:seconds.

Example: OSPF NSSA Area with RFC 3101 Disabled and RFC 1587 Active

In the following example, the output for the **show ip ospf** and **show ip ospf database nssa** commands shows an Open Shortest Path First Not-So-Stubby Area (OSPF NSSA) area where RFC 3101 is disabled, RFC 1587 is active, and an NSSA Area Border Router (ABR) device is configured as a forced NSSA LSA translator. If RFC 3101 is disabled, the forced NSSA LSA translator remains inactive.

```
Device#show ip ospf

Routing Process "ospf 1" with ID 10.0.2.1
Start time: 00:00:25.512, Time elapsed: 00:01:02.200
Supports only single TOS(TOS0) routes
Supports opaque LSA
Supports Link-local Signaling (LLS)
Supports area transit capability
Supports NSSA (compatible with RFC 1587)
Event-log enabled, Maximum number of events: 1000, Mode: cyclic
Router is not originating router-LSAs with maximum metric
Initial SPF schedule delay 5000 msecs
Minimum hold time between two consecutive SPFs 10000 msecs
Maximum wait time between two consecutive SPFs 10000 msecs
Incremental-SPF disabled
Minimum LSA interval 5 secs
Minimum LSA arrival 1000 msecs
LSA group pacing timer 240 secs
Interface flood pacing timer 33 msecs
Retransmission pacing timer 66 msecs
Number of external LSA 0. Checksum Sum 0x000000
Number of opaque AS LSA 0. Checksum Sum 0x000000
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 0 normal 0 stub 1 nssa
Number of areas transit capable is 0
External flood list length 0
IETF NSF helper support enabled
Cisco NSF helper support enabled
Reference bandwidth unit is 100 mbps
Area 1
Number of interfaces in this area is 1
It is a NSSA area
Configured to translate Type-7 LSAs, inactive (RFC3101 support
disabled)
Area has no authentication
SPF algorithm last executed 00:00:07.160 ago
SPF algorithm executed 3 times
Area ranges are
Number of LSA 3. Checksum Sum 0x0245F0
Number of opaque link LSA 0. Checksum Sum 0x000000
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0
```

The table below describes the significant fields shown in the **show ip ospf** command output.

Table 2: show ip ospf Field Descriptions

Field	Description
Supports NSSA (compatible with RFC 1587)	Specifies that RFC 1587 is active or that the OSPF NSSA area is RFC 1587 compatible.
Configured to translate Type-7 LSAs, inactive (RFC3101 support disabled)	Specifies that OSPF NSSA area has an ABR device configured to act as a forced translator of Type 7 LSAs. However, it is inactive because RFC 3101 is disabled

```
Device2#show ip ospf database nssa

Router Link States (Area 1)
LS age: 28
Options: (No TOS-capability, DC)
LS Type: Router Links
Link State ID: 10.0.2.1
Advertising Router: 10.0.2.1
LS Seq Number: 80000004
Checksum: 0x5CA2
Length: 36
Area Border Router
AS Boundary Router
Unconditional NSSA translator
Number of Links: 1
Link connected to: a Stub Network
(Link ID) Network/subnet number: 192.0.2.5
(Link Data) Network Mask: 255.255.255.0
Number of MTID metrics: 0
TOS 0 Metrics: 10
```

The table below describes the significant fields shown in the **show ip ospf database nssa** command output.

Table 3: show ip ospf database nssa Field Description

Field	Description
Unconditional NSSA translator	Specifies that NSSA ASBR device is a forced NSSA LSA translator

Additional references

Table 4: Standards and RFCs

Standard/RFC	Title
RFC 3101	The OSPF Not-So-Stubby Area (NSSA) Option
RFC 1587	The OSPF NSSA Option

CHAPTER 13

OSPFv3 Authentication Trailer

Feature History for OSPFv3 Authentication Trailer

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPFv3 Authentication Trailer: The OSPFv3 Authentication Trailer is a security feature that provides an alternative mechanism, to IPsec, to authenticate OSPFv3 protocol packets.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

OSPFv3 Authentication Trailer

The OSPFv3 Authentication Trailer is a security feature that provides an alternative mechanism, to IPsec, to authenticate Open Shortest Path First version 3 (OSPFv3) protocol packets. Prior to the OSPFv3 authentication trailer, OSPFv3 IPsec was the only mechanism for authenticating protocol packets. The OSPFv3 authentication trailer is also platform independent.

Key aspects of OSPFv3 Authentication Trailer

The OSPFv3 authentication trailer feature, also referred to as non-IPsec cryptographic authentication feature, includes the following key aspects:

- a special data block, the authentication trailer, is attached to the end of the OSPFv3 packet,

- the length of the authentication trailer is included in the IPv6 payload length, but not the in the length of the OSPFv3 packet,
- for OSPFv3 hello packets and database description packets, in the OSPFv3 Options field, the
  - the Link-Local Signaling (LLS) block is established by setting the L-bit. If present, the LLS data block is included in the cryptographic authentication computation along with the OSPFv3 packet.
  - the authentication trailer bit is set. The authentication trailer bit indicates that the authentication trailer is present.
- The authentication trailer bit setting from the OSPFv3 hello and database description packets is preserved in the OSPFv3 neighbor data structure. For other OSPFv3 packet types (that do not include the OSPFv3 options field), the authentication trailer is determined by using the neighbor data structure.

OSPFv3 authentication trailer components

To authenticate outgoing and incoming OSPFv3 packets, OSPFv3 authentication trailer uses

- cryptographic keys: OSPFv3 authentication trailer uses Cisco IOS key chains to create and manage the keys.
- message digest: A fixed-size cryptographic hash value generated from the contents of a message or data packet. It acts as a unique fingerprint of the data, ensuring data integrity and authenticity.
- Authentication algorithm: A cryptographic method used to generate and verify the message digest attached to OSPFv3 packets.
- Security Association ID: An identifier that links the authentication algorithm and the secret key needed to generate and verify the message digest for authenticating OSPFv3 packets.

How OSPFv3 packet authentication works with OSPFv3 Authentication Trailer

Workflow

With OSPFv3 authentication trailer

1. the Security Association ID links the authentication algorithm with the secret key to calculate the message digest.
2. This digest is attached to the OSPFv3 packet to verify that the packet has not been altered in transit and that it comes from a trusted source.
3. When a device receives the packet, it uses the same Security Association ID (to link the same algorithm with the same secret key) to recalculate the message digest and compares it to the received digest.
4. If they match, the packet is considered authentic and unmodified; otherwise, it is discarded to maintain secure communication.

Key selection for outgoing and incoming packet

When sending OSPFv3 packets (outgoing),

- the device selects the key from the key chain based on these rules:
  - Choose the key that will expire last (i.e., the one with the latest stop time).
  - If two keys share the same expiration time, select the key with the highest key ID.
- If the authentication is configured but the last valid key has expired, packets are still sent using the key, and a syslog message is also generated.
- If no valid key is available, the packet is sent without the authentication trailer.

When a packet is received (incoming), its key ID is used to find the corresponding key in the key chain.

- If the key ID is not found or the security association is invalid, the packet is dropped.
- If the corresponding key is found, the packet is verified using the configured algorithm and key for that key ID.

Key chain rollover

Key chains support rollover using key lifetimes. You can add a new key to a key chain with a future send start time. This setting allows the new key to be configured on all the devices before the keys are used.

Sequence number

The OSPFv3 authentication trailer feature also provides packet replay protection through sequence number.

OSPFv3 hello packets have higher priority than other OSPFv3 packets, so they can get reordered on the outgoing interface. This reordering can create problems with sequence number verification on neighboring devices. To prevent sequence mismatch, OSPFv3 verifies the sequence number separately for each packet type.

See RFC 7166 for more details on the authentication procedure.

Deployment of authentication trailer on the network

During the initial rollout of the OSPFv3 authentication trailer feature, the deployment mode allows adjacency to be maintained between devices that have the authentication trailer configured and those that do not yet have it configured. In deployment mode, devices process packets differently to achieve this compatability.

- Outgoing packets: The OSPF checksum is calculated even if the authentication trailer is configured. This ensures compatibility with devices not yet using the authentication trailer.
- Incoming packets: Packets that either lack the authentication trailer or have an incorrect authentication hash are dropped to maintain security.
- The command **show ospfv3 neighbor detail** in deployment mode displays the authentication status of the last received packet, which helps verify whether the authentication trailer feature is functioning correctly before switching to normal mode.

Once verified, the mode can be changed to normal using the **authentication mode normal** command, where the authentication trailer is fully enforced for all packets.

Configure the OSPFv3 Authentication Trailer

To configure OSPFv3 authentication trailer, perform this procedure:

Before you begin

An authentication key is required for configuring OSPFv3 authentication trailer.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type number*

Example:

```
Device(config)# interface ethernet 1/0/1
```

Configures an interface.

Step 4 **ospfv3** [*process-id*] [**ipv4** | **ipv6**] **authentication** {**key-chain** *chain-name* | **null**}

Example:

```
Device(config-if)# ospfv3 1 ipv6 authentication key-chain ospf-1
```

Specifies the authentication type for an OSPFv3 instance.

- *process-id*: The process ID is an internally used, identification parameter that is locally assigned. Each OSPF has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **keychain** : This option enables keychain authentication. Keychain authentication is a more robust and flexible method for managing authentication keys. It allows you to define multiple keys within a named keychain, along with their lifetimes, enabling automatic time-based key rotation. This enhances security by regularly changing the cryptographic keys used for authentication.

- *chain-name*: Refers to a pre-configured keychain that contains the actual authentication keys.

- **null**: this option configures "null authentication," meaning that no authentication is applied to OSPFv3 packets. While it effectively disables authentication, it is a configurable state. This might be used in specific scenarios where authentication is not required on a particular interface or area, even if other parts of the OSPFv3 domain use authentication.

Step 5 **router ospfv3** [*process-id*]

Example:

```
Device(config-if)# router ospfv3 1
```

Enters OSPFv3 router configuration mode.

process-id: The process ID is an internally used, identification parameter that is locally assigned. Each OSPF has a unique process ID.

Process ID can be a positive integer from 1 to 65535.

Step 6 **address-family ipv6 unicast**

Example:

```
Device(config-router)# address-family ipv6 unicast
```

Configures the IPv6 address family in the OSPFv3 process and enters IPv6 address family configuration mode.

unicast: Specifies the configuration is for IPv6 unicast routing.

Step 7 **area** *area-id* **authentication** {**key-chain** *chain-name* | **null**}

Example:

```
Device(config-router-af)# area 1 authentication key-chain ospf-chain-1
```

Configures the authentication trailer on all interfaces in the OSPFv3 area.

- *area-id*: Identifier of the area about which routes are to be summarized. It can be specified as either a decimal value or as an IPv6 prefix.
- **keychain** : This option enables keychain authentication. Keychain authentication is a more robust and flexible method for managing authentication keys. It allows you to define multiple keys within a named keychain, along with their lifetimes, enabling automatic time-based key rotation. This enhances security by regularly changing the cryptographic keys used for authentication.
- *chain-name*: Refers to a pre-configured keychain that contains the actual authentication keys.
- **null**: this option configures "null authentication," meaning that no authentication is applied to OSPFv3 packets. While it effectively disables authentication, it is a configurable state. This might be used in specific scenarios where authentication is not required on a particular interface or area, even if other parts of the OSPFv3 domain use authentication.

Step 8 **area** *area-id* **virtual-link** *router-id* **authentication key-chain** *chain-name*

Example:

```
Device(config-router-af)# area 1 virtual-link 1.1.1.1 authentication key-chain ospf-chain-1
```

Configures the authentication for virtual links.

- *area-id* **virtual-link** *router-id*: Specifies the OSPF area ID and the router ID of the router at the other end of the virtual link.
- **keychain** : This option enables keychain authentication. Keychain authentication is a more robust and flexible method for managing authentication keys. It allows you to define multiple keys within a named keychain, along with their lifetimes, enabling automatic time-based key rotation. This enhances security by regularly changing the cryptographic keys used for authentication.

- *chain-name*: Refers to a pre-configured keychain that contains the actual authentication keys.

Step 9 **area** *area-id* **sham-link** *source-address destination-address* **authentication key-chain** *chain-name*

Example:

```
Device(config-router-af)# area 1 sham-link 1.1.1.1 1.1.1.0 authentication key-chain
ospf-chain-1
```

Configures the authentication for sham-links.

- *area-id* **sham-link**: This specifies the OSPF area ID to which the sham link belongs. The sham link must be configured within an existing OSPF area, and typically, this is the same area that the connected Customer Edge (CE) routers belong to.
- *source-address*: This is the IP address of the local PE router's endpoint for the sham link. This is usually a loopback interface address that is part of the VPN routing and forwarding (VRF) instance and is advertised via BGP, not OSPF.
- *destination-address*: This is the IP address of the remote PE router's endpoint for the sham link. This address must also be a loopback interface address on the remote PE, part of the same VRF, and advertised via BGP.
- **keychain** : This option enables keychain authentication. Keychain authentication is a more robust and flexible method for managing authentication keys. It allows you to define multiple keys within a named keychain, along with their lifetimes, enabling automatic time-based key rotation. This enhances security by regularly changing the cryptographic keys used for authentication.
- *chain-name*: Refers to a pre-configured keychain that contains the actual authentication keys.

Step 10 **authentication mode** {**deployment** | **normal**}

Example:

```
Device(config-router-af)# authentication mode deployment
```

(Optional) Specifies the type of authentication used for the OSPFv3 instance.

- **deployment**: This mode is typically used during the initial setup, testing, or troubleshooting of an authentication system.
- **normal**: This is the standard operational mode once the authentication system has been thoroughly tested and validated.

Step 11 **end**

Example:

```
Device(config-router-af)# end
```

Exits IPv6 address family configuration mode and returns to privileged EXEC mode.

Step 12 **show ospfv3 interface**

Example:

```
Device# show ospfv3
```

(Optional) Displays OSPFv3-related interface information.

Step 13 **show ospfv3 neighbor** [*detail*]

Example:

```
Device# show ospfv3 neighbor detail
```

(Optional) Displays OSPFv3 neighbor information on a per-interface basis.

detail: This is an optional keyword that, when added, provides more extensive and detailed information about each OSPFv3 neighbor.

Step 14 **debug ospfv3**

Example:

```
Device# debug ospfv3
```

(Optional) Displays debugging information for OSPFv3.

Configuration examples for the OSPFv3 Authentication Trailer

The following sections provide examples on how to configure the OSPFv3 authentication trailer and how to verify the OSPFv3 authentication trailer configuration.

Example: Configure the OSPFv3 Authentication Trailer

The following example shows how to define authentication trailer on GigabitEthernet interface 1/0/1:

```
Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# ospfv3 1 ipv6 authentication key-chain ospf-1
Device(config-if)# router ospfv3 1
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# area 1 authentication key-chain ospf-1
Device(config-router-af)# area 1 virtual-link 1.1.1.1 authentication key-chain ospf-1
Device(config-router-af)# area 1 sham-link 1.1.1.1 authentication key-chain ospf-1
Device(config-router-af)# authentication mode deployment
Device(config-router-af)# end
Device(config)# key chain ospf-1
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string ospf
Device(config-keychain-key)# cryptographic-algorithm hmac-sha-256
```

Example: Verify OSPFv3 Authentication Trailer

The following example shows the output of the **show ospfv3** command.

```
Device# show ospfv3
 OSPFv3 1 address-family ipv6
 Router ID 1.1.1.1
…
RFC1583 compatibility enabled
 Authentication configured with deployment key lifetime
 Active Key-chains:
  Key chain ospf-1: Send key 1, Algorithm HMAC-SHA-256, Number of interfaces 1
    Area BACKBONE(0)
```

CHAPTER 14

OSPFv3 Demand Circuit Ignore

Feature History for OSPFv3 Demand Circuit Ignore

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPFv3 Demand Circuit Ignore: The OSPFv3 Demand Circuit Ignore is a device enhancement feature to prevent an interface on an OSPFv3-configured device from accepting demand-circuit requests from other OSPF-configured devices.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

OSPFv3 Demand Circuit Ignore

The OSPFv3 Demand Circuit Ignore is a device enhancement feature to prevent an interface on an OSPFv3-configured access switch from accepting demand-circuit requests from other OSPF-configured devices (routers or access-switches)

How OSPFv3 Demand Circuit Ignore works

A demand circuit enables devices to reduce OSPF routing traffic on certain types of links by suppressing periodic hello and link-state update messages. These messages are replaced by updates that are sent only when there is a change in the network topology. A demand circuit request is a negotiation attempts by one device to establish demand circuit behavior with another device on the interface.

When a device receives a demand circuit request, it can either accept or ignore it. Configuring the **ignore** keyword in the ipv6 **ospf demand-circuit** command instructs the device to stop accepting these demand circuit negotiation requests from other devices. This approach is useful, for example, on point-to-multipoint interfaces of a hub router to control OSPF behavior.

Configure OSPFv3 Demand Circuit Ignore Support

Perform this task to configure OSPFv3 demand circuit ignore.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *interface-id*

Example:

```
Device(config)# interface gigabitethernet 1/0/1
```

Enters interface configuration mode, and specifies the Layer 3 interface to configure.

Step 4 Choose one of the following:

- **ipv6 ospf demand-circuit ignore**
- **ospfv3 demand-circuit ignore**

Example:

```
Device(config-if)# ipv6 ospf demand-circuit ignore
OR

Device(config-if)# ospfv3 demand-circuit ignore
```

Prevents an interface from accepting demand-circuit requests from other devices.

Step 5 **end**

Example:

```
Device(config-if)# end
```

Returns to privileged EXEC mode.

Step 6 **show ospfv3** *process-id* [*area-id*] [*address-family*] [**vrf** {*vrf-name* | *****}] **interface** [*type-number*] [**brief**]

Example:

```
Device(config)# end
```

(Optional) Displays OSPFv3-related interface information.

- *process-id*: (Optional) The process ID is an internally used, identification parameter that is locally assigned. Each OSPF has a unique process ID.
- *address-family*: (Optional) Specifies the address family (IPv4 or IPv6).
- **vrf** {*vrf-name* | *****}: (Optional) Specifies the VRF instance by name or all VRFs with `*`.
- **interface** [*type-number*: (Optional) Specifies the interface type and number to display information.
- **brief**: (Optional) Displays a brief overview of OSPFv3 interface information.

Configuration example for OSPFv3 Demand Circuit Ignore

The following example shows how to configure demand circuit ignore support for OSPFv3:

```
Device> enable
Device# configure terminal
Device(config)# interface Serial0/0
Device(config-if)# ip address 6.1.1.1 255.255.255.0
Device(config-if)# ipv6 enable
Device(config-if)# ospfv3 network point-to-multipoint
Device(config-if)# ospfv3 demand-circuit ignore
Device(config-if)# ospfv3 1 ipv6 area 0
Device(config-if)# end
```

CHAPTER 15

OSPFv3 External Path Preferences

Feature History for OSPFv3 External Path Preference

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPFv3 External Path Preference: OSPFv3 External Path Preference is a network optimization feature that allows a device to choose the best path to an ASBR or a forwarding address when multiple intra-AS paths are available.	Cisco C9610 Series Smart Switches

OSPFv3 External Path Preference

OSPFv3 External Path Preference is a network optimization feature that allows a device to choose the best path to an Autonomous System Boundary Router (ASBR) or a forwarding address when multiple intra-AS paths are available. This feature is based on RFC 5340.

When an OSPF device needs to reach an ASBR or a forwarding address, and there are multiple paths available within its own Autonomous System (intra-AS), RFC 5340 specifies the following preference order:

1. Intra-area paths using non-backbone areas are always the most preferred.

   - This means if a path to an ASBR or forwarding address is entirely contained within a single non-backbone OSPF area (e.g., Area 1, Area 2, etc.), that path will be chosen over any other type of intra-AS path.

2. Intra-area backbone paths and inter-area paths are of equal preference.
   - If the most preferred path (intra-area non-backbone) is not available, OSPF will then consider paths that are either:
     - Entirely within the backbone area (Area 0).
     - Inter-area paths, meaning they traverse multiple areas, typically via an Area Border Router (ABR) and the backbone.
   - Between these two types, there is no inherent preference; the path with the lowest cost will be selected.

How OSPFv3 External Path Preference works

It's crucial to note that these specific path preference rules from RFC 5340 only apply when RFC 1583 compatibility is disabled.

These rules are applied in two primary scenarios:

- Same ASBR, Multiple Areas:

  When a single ASBR is reachable through interfaces in different OSPF areas. In this case, all paths lead to the same ASBR, but the OSPF device must decide which path (via which area) is optimal.

- Deciding Between AS-External-LSAs:

  When the OSPF device needs to choose between several AS-external-LSAs (Type 5 LSAs). These LSAs may originate from different ASBRs or point to different forwarding addresses. The rules help determine which external route is preferred based on how the ASBRs or forwarding addresses are reached internally.

In both scenarios, each potential path is represented as a separate entry in the routing table, and these preference rules guide the selection of the best entry.

Enable OSPFv3 External Path Preference

Perform this task to enable OSPFv3 external path reference by disabling RFC 1583.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospfv3** *process-id*

Example:

```
Device(config)# router ospf 15
```

Enables OSPFv3 routing and enters router configuration mode.

process-id: The process ID is an internally used, identification parameter that is locally assigned. Each OSPF has a unique process ID.

Process ID can be a positive integer from 1 to 65535.

Step 4 **no compatible rfc1583**

Example:

```
Device(config-router)# no compatible rfc1583
```

Changes the method used to calculate external path preferences per RFC 5340.

Configuration example to verify OSPFv3 External Path Preference

The following example shows how to verify if RFC 1583 is disabled.

```
Device# show ospfv3
Routing Process "ospfv3 1" with ID 10.1.1.1
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of areas in this router is 1. 1 normal 0 stub 0 nssa
 Reference bandwidth unit is 100 mbps
 RFC 1583 compatibility disabled
    Area BACKBONE(0) (Inactive)
        Number of interfaces in this area is 1
        SPF algorithm executed 1 times
        Number of LSA 1. Checksum Sum 0x00D03D
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

CHAPTER 16

OSPFv3 Fast Convergence - LSA and SPF Throttling

Feature History for OSPFv3 Fast Convergence - LSA and SPF Throttling

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPFv3 Fast Convergence - LSA and SPF Throttling: OSPFv3 Fast Convergence: LSA and SPF Throttling is a network enhancement feature that improves OSPFv3 network stability and convergence speed by dynamically regulating how frequently LSAs are updated and how often the SPF algorithm is triggered, especially during periods of network instability.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

OSPFv3 Fast Convergence: LSA and SPF Throttling

OSPFv3 Fast Convergence: LSA and SPF Throttling is a network enhancement feature that improves OSPFv3 (Open Shortest Path First version 3) network stability and convergence speed by dynamically regulating how frequently link-state advertisements (LSAs) are updated and how often the shortest-path first (SPF) algorithm is triggered, especially during periods of network instability.

OSPFv3 typically uses static, user-configurable timers (measured in seconds) to limit the frequency of SPF calculations and LSA generation. This can slow down convergence, especially when rapid response is required.

The LSA and SPF throttling feature allows these rate-limiting timers to be configured in milliseconds rather than seconds, allowing for subsecond convergence. Instead of fixed intervals, the throttling mechanism increases (backs off) or decreases (speeds up) the timer based on network stability.

Configure OSPFv3 Fast Convergence: LSA and SPF Throttling

The following sections provide configuration information about OSPFv3 Fast Convergence: LSA and SPF throttling.

Tune LSA and SPF Timers for OSPFv3 Fast Convergence

To tune LSA and SPF timers for OSPFv3 fast convergence, perform this procedure:

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospfv3** *process-id*

Example:

```
Device(config)# router ospf 15
```

Enables OSPFv3 routing and enters router configuration mode.

process-id: The process ID is an internally used, identification parameter that is locally assigned. Each OSPF has a unique process ID.

Process ID can be a positive integer from 1 to 65535.

Step 4 **timers lsa arrival** *milliseconds*

Example:

```
Device(config-router)# timers lsa arrival 300
```

Sets the minimum interval (in milliseconds) at which the device must wait before accepting a new instance of the same LSA from the OSPFv3 neighbors.

Step 5 **timers pacing flood** *milliseconds*

Example:

```
Device(config-router)# timers pacing flood 30
```

Controls the time interval (in milliseconds) between batches of Link-State Advertisements (LSAs) that are flooded out OSPFv3 interfaces.

Step 6 **timers pacing lsa-group** *seconds*

Example:

```
Device(config-router)# timers pacing lsa-group 300
```

Controls the interval (in seconds) at which OSPFv3 devices group and send multiple LSA acknowledgments together, rather than sending each one immediately.

Step 7 **timers pacing retransmission** *milliseconds*

Example:

```
Device(config-router)# timers pacing retransmission 100
```

Specifies the interval (in milliseconds) between retransmissions of OSPFv3 LSAs that have not been acknowledged by neighbors.

Configure LSA and SPF Throttling for OSPFv3 Fast Convergence

To configure LSA and SPF throttling for OSPFv3 fast convergence, perform this procedure:

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ipv6 router ospf** *process-id*

Example:

```
Device(config)# ipv6 router ospf 1
```

Enables OSPFv3 router configuration mode.

process-id: The process ID is an internally used, identification parameter that is locally assigned. Each OSPF has a unique process ID.

Process ID can be a positive integer from 1 to 65535.

Step 4 **timers throttle spf** *spf-start spf-hold spf-max-wait*

Example:

```
Device(config-router)# timers throttle spf 200 200 200
```

Controls how OSPFv3 delays and schedules SPF calculations when there are frequent topology changes. Instead of recalculating routes immediately for every change, it uses a dynamic, throttled approach to optimize convergence speed and network stability.

- *spf-start*: Initial delay (in milliseconds) before running the first SPF calculation after a topology change.
- *spf-hold*: Minimum hold time (in milliseconds) between consecutive SPF calculations when changes persist.
- *spf-max-wait*: Maximum wait time (in milliseconds) between SPF calculations if instability continues.

Step 5 **timers throttle lsa** *start-interval hold-interval max-interval*

Example:

```
Device(config-router)# timers throttle lsa 200 200 200
```

Controls how frequently OSPFv3 generates and floods Link-State Advertisements (LSAs) during periods of network change. This throttling helps achieve fast convergence while protecting the router and network from excessive LSA generation during instability.

- *start-interval*: Initial delay (in milliseconds) before generating the first LSA after a change.
- *hold-interval*: Minimum time (in milliseconds) to wait before generating the next LSA if changes persist.
- *max-interval*: Maximum wait time (in milliseconds) between LSA generations during persistent instability.x

Step 6 **timers lsa arrival** *milliseconds*

Example:

```
Device(config-router)# timers lsa arrival 300
```

Sets the minimum interval (in milliseconds) at which the device must wait before accepting a new instance of the same LSA from the OSPFv3 neighbors.

Step 7 **timers pacing flood** *milliseconds*

Example:

```
Device(config-router)# timers pacing flood 30
```

Controls the time interval (in milliseconds) between batches of Link-State Advertisements (LSAs) that are flooded out OSPFv3 interfaces.

Configuration Example for LSA and SPF Throttling for OSPFv3 Fast Convergence

The following example show how to display the configuration values for SPF and LSA throttling timers:

```
Device# show ipv6 ospf

Routing Process "ospfv3 1" with ID 10.9.4.1
Event-log enabled, Maximum number of events: 1000, Mode: cyclic
 It is an autonomous system boundary router
 Redistributing External Routes from,
    ospf 2
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
```

CHAPTER 17

OSPFv3 Graceful Restart

Feature History for OSPFv3 Graceful Restart

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPFv3 Graceful Restart: The OSPFv3 Graceful Restart feature is an enhancement feature that allows devices to continue forwarding data traffic using previously known routes, even while the OSPFv3 process is restarting.	Cisco C9610 Series Smart Switches

OSPFv3 Graceful Restart

The OSPFv3 Graceful Restart feature is an enhancement feature that allows devices to continue forwarding data traffic using previously known routes, even while the OSPFv3 process is restarting.

How does OSPFv3 Graceful Restart work

For OSPFv3 graceful restart to work, the following conditions must be met:

- a device capable of graceful restart must be configured with high availability and stateful switchover (SSO).

- the neighboring devices be graceful-restart aware

A device will perform the graceful restart function when the following failures occur:

- A route processor (RP) failure that results in switchover to standby RP.
- A planned RP switchover to standby RP.

The restarting devices notifies its OSPFv3 neighbors that it is undergoing a graceful restart. Neighbors maintain their adjacencies and OSPFv3 state with the restarting device, avoiding unnecessary route recalculation.

How to configure OSPFv3 Graceful Restart

This section shows how to configure OSPFv3 graceful restart.

Enable OSPFv3 Graceful Restart on a graceful restart-capable device

Perform this task to enable OSPFv3 graceful restart on a graceful restart-capable device. You can perform this task on both IPv4 and IPv4 networks.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **graceful-restart** [**restart-interval** *interval*]

Example:

```
Device(config-router)# graceful-restart
```

Enables the OSPFv3 graceful restart feature on a graceful-restart-capable device.

restart-interval *interval*: (Optional) Sets the maximum time (in seconds) that the router will attempt to maintain OSPF state during the restart.

Enable OSPFv3 Graceful Restart on a graceful restart-aware device

Perform this task to enable OSPFv3 graceful restart on a graceful restart-aware device.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **graceful-restart helper** {**disable** | **strict-lsa-checking**}

Example:

```
Device(config-router)# graceful-restart helper strict-lsa-checking
```

Enables the OSPFv3 graceful restart feature on a graceful-restart-aware device.

- **disable**: Disables the helper mode. If the device detects a neighbor restarting, it will treat it as a failure and bring down the adjacency.
- **strict-lsa-checking**: Enables strict LSA checking during the helper mode. The helper device will terminate the helper process and bring down the adjacency if it detects any change in the LSAs that would impact network topology or routing (such as a new LSA or a changed LSA relevant to the restarting neighbor). This ensures that routing consistency is maintained during the restart.

Configuration example for OSPFv3 Graceful Restart

The following example shows how to enable OSPFv3 graceful restart.

```
Device# show ipv6 ospf graceful-restart
Routing Process "ospf 1"
    Graceful Restart enabled
        restart-interval limit: 120 sec, last restart 00:00:15 ago (took 36 secs)
    Graceful Restart helper support enabled
    Router status : Active
    Router is running in SSO mode
    OSPF restart state : NO_RESTART
    Router ID 10.1.1.1, checkpoint Router ID 10.0.0.0
```

The following example shows OSPFv3 information with graceful-restart helper support enabled on a graceful-restart-aware device.

```
Device# show ospfv3
    Routing Process "ospfv3 1" with ID 10.0.0.1
    Supports IPv6 Address Family
    Event-log enabled, Maximum number of events: 1000, Mode: cyclic
    Initial SPF schedule delay 5000 msecs
    Minimum hold time between two consecutive SPFs 10000 msecs
    Maximum wait time between two consecutive SPFs 10000 msecs
    Minimum LSA interval 5 secs
    Minimum LSA arrival 1000 msecs
    LSA group pacing timer 240 secs
    Interface flood pacing timer 33 msecs
    Retransmission pacing timer 66 msecs
    Number of external LSA 0. Checksum Sum 0x000000
    Number of areas in this router is 0. 0 normal 0 stub 0 nssa
    Graceful restart helper support enabled
    Reference bandwidth unit is 100 mbps
    Relay willingness value is 128
    Pushback timer value is 2000 msecs
    Relay acknowledgement timer value is 1000 msecs
    LSA cache Disabled : current count 0, maximum 1000
    ACK cache Disabled : current count 0, maximum 1000
    Selective Peering is not enabled
    Hello requests and responses will be sent multicast
```

CHAPTER 18

OSPFv3 Graceful Shutdown

Feature History for OSPFv3 Graceful Shutdown

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPFv3 Graceful Shutdown: The OSPFv3 Graceful Shutdown is an enhancement feature that allows network administrators to temporarily and smoothly disable OSPFv3 on a device. This process is designed to minimize disruption to network traffic.	Cisco C9610 Series Smart Switches

OSPFv3 Graceful Shutdown

The OSPFv3 Graceful Shutdown is an enhancement feature that allows network administrators to temporarily and smoothly disable OSPFv3 on a device. This process is designed to minimize disruption to network traffic.

How to configure Graceful Shutdown Support for OSPFv3

These sections provide configuration information on how to configure graceful shutdown support for OSPFv3.

Configure Graceful Shutdown of the OSPFv3 process

Perform this task to configure graceful shutdown of the OSPFv3 process.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 Choose one of the following:

- **ipv6 router ospf** *process-id*
- **router ospfv3** *process-id*

Example:

```
Device(config)# ipv6 router ospf 1
OR
Device(config)# router ospfv3 101
```

Enables OSPFv3 routing and enters router configuration mode.

process-id: The process ID is an internally used, identification parameter that is locally assigned. Each OSPF has a unique process ID.

Process ID can be a positive integer from 1 to 65535.

Step 4 **shutdown**

Example:

```
Device(config-router)# shutdown
```

Shuts down the selected interface

Step 5 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

Step 6 Choose one of the following:

- **show ipv6 ospf** [*process-id*]
- **show ospfv3** [**process-id**]

Example:

```
Device# show ipv6 ospf
OR
Device# show ospfv3
```

(Optional) Displays general information about OSPFv3 routing processes.

Configure Graceful Shutdown of the OSPFv3 process with Address Family

Perform this task to configure graceful shutdown of the OSPFv3 process with address family.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospf** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospf 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.
- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **address-family ipv6 unicast** [**vrf** *vrf-name*]

Example:

```
Device(config-router)# address-family ipv6
```

Enters IPv6 address family configuration mode for OSPFv3.

Step 5 **shutdown**

Example:

```
Device(config-router-af)# shutdown
```

Shuts down the selected interface.

Step 6 **end**

Example:

```
Device(config-router-af)# end
```

Returns to privileged EXEC mode.

Step 7 **show ospfv3** [*process-id*]

(Optional) Displays general information about OSPFv3 routing processes

Configure OSPFv3 Graceful Shutdown of an OSPFv3 interface

Perform this task to configure OSPFv3 graceful shutdown of an OSPFv3 interface.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *interface-id*

Example:

```
Device(config)# interface gigabitethernet 1/0/1
```

Enters interface configuration mode, and specifies the Layer 3 interface to configure.

Step 4 Choose one of the these commands

- **ipv6 ospf shutdown**
- **ospfv3 shutdown**

Example:

```
Device(config-if)#  ipv6 ospf shutdown
OR
Device(config-if)#  ospfv3 shutdown
```

Initiates an OSPFv3 protocol graceful shutdown at the interface level.

When the **ipv6 ospf shutdown** interface command is entered, the interface on which it is configured sends a ospfv3 shutdown link-state update advising its neighbors that is going down, which allows those neighbors to begin routing OSPFv3 traffic around this device.

Step 5 **end**

Example:

```
Device(config-if)# end
```

Returns to privileged EXEC mode.

Step 6 **show ospfv3** *process-id* [*area-id*] [*address-family*] [**vrf** {*vrf-name* | *}] **interface** [*type-number*] [**brief**]

(Optional) Displays OSPFv3-related interface information.

- *process-id*: The OSPFv3 routing process ID (a number identifying a specific OSPFv3 process).
- *area-id*: (Optional) Specifies information for a particular OSPF area.
- *address-family*: (Optional) Specifies address family (like ipv4 or ipv6).
- *type-number*: (Optional) Specifies the type and number of a specific interface (e.g., GigabitEthernet 0/0).

Configuration examples for Graceful Shutdown Support for OSPFv3

The following example shows how to configure graceful shutdown of the OSPFv3 process in IPv6 router OSPF configuration mode configuration mode:

```
Device> enable
Device# configure terminal
Device(config)# ipv6 router ospf 6
Device(config-router)# router-id 10.10.10.10
Device(config-router)# shutdown
```

The following example shows how to configure graceful shutdown of the OSPFv3 process in router OSPFv3 configuration mode

```
Device> enable
Device# configure terminal
Device(config)# router ospfv3 1
Device(config-router)# shutdown
!
Device(config-router)# address-family ipv6 unicast
Device(config-router)# exit-address-family
```

The following example shows how to configure graceful shutdown of the OSPFv3 process in address-family configuration mode:

```
Device> enable
Device# configure terminal
Device(config)# router ospfv3 1
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# shutdown
Device(config-router-af)# exit-address-family
```

The following example shows how to configure graceful shutdown of the OSPFv3 interface using the **ipv6 ospf shutdown** command:

```
Device> enable
Device# configure terminal
Device(config)# interface Serial2/1
Device(config-if)# no ip address
Device(config-if)# ipv6 enable
Device(config-if)# ipv6 ospf 6 area 0
Device(config-if)# ipv6 ospf shutdown
Device(config-if)# serial restart-delay 0
Device(config-if)# end
```

The following example shows how to configure graceful shutdown of the OSPFv3 interface using the **ospfv3 shutdown** command:

```
Device> enable
Device# configure terminal
Device(config)# interface Serial2/0
Device(config-if)# ip address 10.10.10.10 255.255.255.0
Device(config-if)# ip ospf 1 area 0
Device(config-if)# ipv6 enable
Device(config-if)# ospfv3 shutdown
Device(config-if)# ospfv3 1 ipv6 area 0
Device(config-if)# serial restart-delay 0
Device(config-if)# end
```

CHAPTER 19

OSPFv3 Max-Metric Router LSA

Feature History for OSPFv3 Max-Metric Router LSA

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPFv3 Max-Metric Router LSA: OSPFv3 Max-Metric Router LSA is a network optimization feature that enables OSPFv3 to advertise its locally generated LSAs with a maximum metric.	Cisco C9610 Series Smart Switches

OSPFv3 Max-Metric Router LSA

OSPFv3 Max-Metric Router LSA is a network optimization feature that enables OSPFv3 to advertise its locally generated router link-state advertisements (LSAs) with a maximum metric. The enables the OSPFv3 process on the device to converge with the network and discourages other devices from sending traffic through it if more optimal paths exist.

How OSPFv3 Max-Metric Router LSA works

By using the max-metric LSA control, an OSPFv3 device effectively places itself into a "stub router" role. A stub router's primary function is to forward packets only to its directly connected links.

In OSPFv3 networks, a device can achieve the stub router status by advertising a very large metric for its connected links. This makes the cost of routing traffic through this device significantly higher than through alternative paths. Specifically, the OSPFv3 stub router advertisement allows a device to advertise an "infinity metric" (0xFFFF) for its connected links in router LSAs. The device advertises the normal interface cost if the link is a stub network.

Configure OSPFv3 Max-Metric Router LSA

Perform this task on a device to configure OSPFv3 max-metric router LSA.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospfv3** *process-id*

Example:

```
Device(config)# router ospf 15
```

Enables OSPFv3 routing and enters router configuration mode.

process-id: The process ID is an internally used, identification parameter that is locally assigned. Each OSPF has a unique process ID.

Process ID can be a positive integer from 1 to 65535.

Step 4 **address-family ipv6 unicast** [**vrf** *vrf-name*]

Example:

```
Device(config-router)# address-family ipv6
```

Enters IPv6 address family configuration mode for OSPFv3.

Step 5 **max-metric router-lsa** [**external-lsa** [*max-metric-value*]] [**include-stub**] [**inter-area-lsas** *max-metric-value*]] [**on-startup** {*seconds* | **wait-for-bgp**}] [**prefix-lsa**] [**stub-prefix-lsa** [*max-metric-value*]] [**summary-lsa** [*max-metric-value*]]

Example:

```
Device(config-router-af)# max-metric router-lsa on-startup wait-for-bgp
```

Configures a device that is running the OSPFv3 protocol to advertise a maximum metric so that other devices do not prefer the device as an intermediate hop in their SPF calculations.

- **external-lsa***max-metric-value*: Specifies that the ASBR (Autonomous System Boundary Router) should advertise external routes (Type 5 LSAs) with a maximized metric.

  This configuration is useful if you want routes learned from other routing protocols (and redistributed into OSPF to appear less favorable.

- **include-stub**: Specifies that even stub links (links to networks directly connected to the device that are considered stub networks) are advertised with the max-metric. This command is used along with **max-metric router-lsa** command.

  By default, the `max-metric router-lsa` command might only apply to transit links. The **include-stub** ensures all connected links are affected.

- **inter-area-lsas** *max-metric-value*: Specifies that the the ABR to advertise inter-area routes (Type 3 LSAs) with a maximized metric.

  This configuration is useful to control traffic flow between OSPF areas, making the ABR less desirable for inter-area transit.

- **on-startup** {*seconds* | **wait-for-bgp**: Controls the behavior of the max-metric feature when the router starts up.

  - *seconds*: The device will advertise max-metric LSAs for a specified number of seconds after startup. After this timer expires, it will revert to advertising normal metrics. This configuration is useful to allow the device to fully converge and establish all adjacencies and routes before it starts attracting transit traffic.

  - **wait-for-bgp**: The device will advertise max-metric LSAs until BGP (Border Gateway Protocol) has fully converged and all BGP prefixes have been received. This configuration is particularly useful in environments where OSPF and BGP are interdependent, ensuring that the device doesn't attract traffic until it has a complete view of external routes.

- **prefix-lsa**: Specifies the device to advertise its connected prefixes with a maximized metric.

  In OSPFv3, prefix information is carried in separate LSAs (Type 8 and Type 9) from the router LSA (Type 1). This option ensures that the prefix advertisements also reflect the max-metric state.

- **stub-prefix-lsa** *max-metric-value*: Specifies the device to advertise its stub prefixes with a maximized metric.

- **summary-lsa** *max-metric-value*: Specifies the ABR to advertise summarized routes to other areas with a maximized metric.

Step 6 **end**

Example:

```
Device(config-router-af)# end
```

Exits address family configuration mode and returns to privileged EXEC mode.

Step 7 **show ospfv3** [*process-id*] **max-metric**

Example:

```
Device(config)# show ospfv3 1 max-metric
```

Displays OSPFv3 maximum metric origination information.

process-id: The process ID is an internally used, identification parameter that is locally assigned. Each OSPF has a unique process ID.

Process ID can be a positive integer from 1 to 65535.

Configuration example to verify the OSPFv3 Max-Metric Router LSA

The following example is a sample output of the **show ipv6 ospf max-metric** command.

```
Device# show ipv6 ospf max-metric

OSPFv3 Router with ID (192.1.1.1) (Process ID 1)

Start time: 00:00:05.886, Time elapsed: 3d02h
Originating router-LSAs with maximum metric
Condition: always, State: active
```

CHAPTER 20

OSPFv3 NSSA

Feature History for OSPFv3 NSSA Option

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPFv3 NSSA Option: The OSPFv3 NSSA Option is a network optimization feature that allows a central site that runs on OSPF protocol to connect to a remote site that runs on any other routing protocol.	Cisco C9610 Series Smart Switches

OSPFv3 NSSA Option

The OSPFv3 NSSA Option is a network optimization feature that allows a central site that runs on OSPF protocol to connect to a remote site that runs on any other routing protocol.

Without OSPFv3 NSSA option

Without OSPFv3 NSSA option, the following happens:

- OSPF stub areas do not allow external route redistribution (from protocols like RIP).
- Two routing protocols must be maintained (OSPF in the core, RIP at the edge).

- The stub area cannot import external routes from the remote network.

Key components in OSPFv3 NSSA option

- Autonomous System Border Router (ASBR): ASBRs connect an OSPFv3 network and a non-OSPFv3 network (or another OSPFv3 domain if routes are being explicitly redistributed). The ASBR redistributes routes from RIP (or another protocol) into OSPFv3 NSSA as type 7 LSAs.
- Area Border Router (ABR): ABRs connect regular areas to the backbone (Area 0). An ABR converts type 7 LSAs to type 5 LSAs, allowing the rest of the OSPFv3 domain to learn about the external routes.
- Route summarization: Route summarization is the process of combining a group of contiguous network addresses into a single summary address. This reduces the size of routing tables and improves routing efficiency. In OSPFv3, summarization is performed at Area Border Routers (ABRs) and Autonomous System Boundary Routers (ASBRs). If an OSPFv3 area contains multiple contiguous subnets, the ABR can be configured to advertise a single summary route to other areas instead of multiple individual routes.

How to configure OSPFv3 NSSA

The following sections provide configuration information on how to configure OSPFv3 NSSA.

Configure an OSPFv3 NSSA area and its parameters

Perform this procedure to configure an OSPFv3 NSSA area and it parameters.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospfv3** *process-id*

Example:

```
Device(config)# router ospf 15
```

Enables OSPFv3 routing and enters router configuration mode.

process-id: The process ID is an internally used, identification parameter that is locally assigned. Each OSPF has a unique process ID.

Process ID can be a positive integer from 1 to 65535.

Step 4 **area** *area-id* **nssa default-information-originate nssa-only**

Example:

```
Device(config-router)# area 1 nssa default-information-originate nssa-only
```

Configures an NSSA area and sets the default advertisement to this NSSA area.

- **default-information-originate**: Select on an ABR to allow importing type 7 LSAs into the NSSA.
- **nssa-only**: Advertises the summary only as a Type 7 LSA into NSSA areas

Step 5 **address-family** {**ipv4** |**ipv6**} [**unicast**]

Example:

```
Device(config-router)# address-family ipv4 unicast
OR
Device(config-router)# address-family ipv6 unicast
```

(Optional) Enables address family configuration mode for Open Shortest Path First version 3 (OSPFv3).

- The **address-family ipv4 unicast** command configures an IPv4 address family.
- The **address-family ipv6 unicast** command configures an IPv6 address family.

Step 6

- For IPv4

  summary-prefix {*ip-prefix* | *ip-address-mask*} [**not-advertise** | [**tag** *tag-value*] [**nssa-only**]]
- For IPv6

  summary-prefix *ipv6-prefix* [**not-advertise** | [**tag** *tag-value*] [**nssa-only**]]

Example:

```
Device(config-router-af)# summary-prefix 10.1.0.0/16 nssa-only
OR
Device(config-router-af)# summary-prefix 2001:DB8::/32 nssa-only
```

Controls the route summarization and filtering during the translation and limits the summary to NSSA areas

- *ip-prefix*: The summarized IPv4 network address and the network mask for the summary route.
- *ipv6-prefix*: The summarized IPv6 network address and the network mask for the summary route.
- **not-advertise**: (Optional) Suppresses the summary route advertisement.
- **tag** *tag*: (Optional) Associates a tag with the summary route for policy or routing decisions.
- **nssa-only**: (Optional) Advertises the summary only as a Type 7 LSA for NSSA areas.

Step 7 **exit**

Example:

```
Device(config-router-af)# exit
```

Exits address-family router configuration mode and returns to the router configuration mode.

Step 8 **redistribute** *protocol* [*process-id*] [*as-number*] [**include-connected** {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value* | **transparent**] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**subnets**] [**nssa-only**] [**tag** *tag-value*] [**route-map** *map-tag*]

Example:

```
Device(config-router)# redistribute rip subnets
```

Redistributes routes from one routing domain into another routing domain.

- *protocol*: Specifies the source of the routes to be redistributed. Common options include:
  - **connected**: Redistributes directly connected networks that are not already part of the OSPF process.
  - **static**: Redistributes static routes configured on the router.
  - **bgp**: Redistributes routes learned via BGP.
  - **eigrp**: Redistributes routes learned via EIGRP.
  - **rip**: Redistributes routes learned via RIP.
  - **ospf**: Redistributes routes from another OSPF process (e.g., if running multiple OSPF instances or different process IDs).
  - **isis**: Redistributes routes learned via IS-IS.
- *process-id*: (Optional) If the source protocol (e.g., EIGRP, OSPF, RIP) uses a process ID, you specify it here to identify the specific instance from which routes should be taken.
- *as-number*: (Optional) If the source `protocol` (e.g., BGP or EIGRP) uses an Autonomous System (AS) number, you specify it here. This is often used interchangeably with `process-id` for protocols that use AS numbers.
- **include-connected** {**level-1** | **level-1-2** | **level-2**}: (Optional) This parameter is typically specific to IS-IS redistribution into OSPF, or sometimes OSPF into IS-IS, or even OSPF into OSPF (though less common). It refers to the IS-IS routing levels:
  - level-1: Redistributes Level-1 IS-IS routes.
  - level-2: Redistributes Level-2 IS-IS routes.
  - level-1-2: Redistributes both Level-1 and Level-2 IS-IS routes.
  - This part of the syntax suggests a context where IS-IS is the source protocol.
- **metric** {*metric-value* | **transparent**}: This is a critical parameter for redistribution. When routes are redistributed from one protocol to another, they need to be assigned a metric that is meaningful to the destination protocol.
  - *metric-value*: A specific numerical value to assign as the metric. The meaning of this value depends on the destination protocol (e.g., hop count for RIP, cost for OSPF).
  - **transparent**: This attempts to carry the original metric information across, but its effectiveness can vary depending on the protocols involved.
- **metric-type** *type-value*: (Optional) For OSPF, this is crucial for how external routes are handled:

- type-1 (E1): The cost to the external destination is the sum of the external metric and the internal OSPF cost to reach the ASBR. This is often preferred for more accurate path selection.
- type-2 (E2): The cost to the external destination is *only* the external metric, regardless of the internal OSPF cost to reach the ASBR. This is the default if not specified. Type 2 routes are always preferred over Type 1 routes if their external metric is the same.

- **match** {**internal** | **external 1** | **external 2**}: This parameter is specific to OSPF when redistributing OSPF routes into another protocol. It allows you to specify which *types* of OSPF routes to redistribute:
  - **internal**: Redistributes OSPF intra-area (Type 1) and inter-area (Type 2) routes.
  - **external 1**: Redistributes OSPF external Type 1 routes.
  - **external 2**: Redistributes OSPF external Type 2 routes.
- **nssa-only**: (Optional) This keyword is specifically used when redistributing routes into a Not-So-Stubby Area (NSSA). When nssa-only is specified, the redistributed routes will be advertised as Type 7 LSAs within the NSSA, and then translated to Type 5 LSAs by the NSSA ABR for propagation to other areas. Without this, redistribution into an NSSA might not be allowed or might behave differently
- **tag** *tag-value*: (Optional) Assigns a numerical tag to the redistributed routes. This tag can be used later in route maps or distribute lists for filtering or policy-based routing.
- **route-map** *map-tag*: (Optional) A very powerful and commonly used parameter. It references a `route-map` that provides granular control over which routes are redistributed and how their attributes (like metric, metric-type, tag) are modified during the redistribution process.
- **subnets**: This parameter is specific to OSPF. By default, OSPF only redistributes classful networks. To redistribute subnetted networks (which is almost always desired in modern networks), you must include the **subnets** keyword. Without it, only major network numbers (e.g., 10.0.0.0/8, 192.168.1.0/24) would be redistributed.

Step 9 **end**

Example:

```
Device(config-router)# end
```

Exits router configuration mode and returns to privileged EXEC mode.

Configure an NSSA ABR as a forced NSSA LSA translator for OSPFv3

Perform this procedure to configure an NSSA ABR as a forced NSSA LSA translator for OSPFv3.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospfv3** *process-id*

Example:

```
Device(config)# router ospfv3 15
```

Enables OSPFv3 routing and enters router configuration mode.

process-id: The process ID is an internally used, identification parameter that is locally assigned. Each OSPF has a unique process ID.

Process ID can be a positive integer from 1 to 65535.

Step 4 **area** *area-id* **nssa translate type7 always**

Example:

```
Device(config-router)# area 10 nssa translate type7 always
```

Configures a NSSA ABR device as a forced NSSA LSA translator.

Note
This command can be used if RFC 3101 is disabled and RFC 1587 is used.

- *area-id*: The OSPF area number or address in dotted-decimal format.
- **always**: Manually force this ABR to always be the translator, ensuring consistent LSA translation and avoiding translator role changes due to device restarts or network topology changes.

Step 5 **area** *area-id* **nssa translate type7 suppress-fa**

Example:

```
Device(config-router)# area 10 nssa translate type7 suppress-fa
```

Allows ABR to suppress the forwarding address in translated Type-5 LSA.

Step 6 **end**

Example:

```
Device(config-router)# end
```

Exits router configuration mode and returns to privileged EXEC mode.

Disable RFC 3101 compatibility and enable RFC 1587 compatibility (OSPFv3)

Perform this task to disable RFC 3101 compatibility and enable RFC 1587 compatibility.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospfv3** *process-id*

Example:

```
Device(config)# router ospfv3 1
```

Enables OSPFv3 routing and enters router configuration mode.

process-id: The process ID is an internally used, identification parameter that is locally assigned. Each OSPF has a unique process ID.

Process ID can be a positive integer from 1 to 65535.

Step 4 **compatible rfc1587**

Example:

```
Device(config-router)# compatible rfc1587
```

Enables the device to be RFC 1587 compliant.

Step 5 **end**

Example:

```
Device(config-router)# end
```

Enables the device to be RFC 1587 compliant.

Configuration example for NSSA for OSPFv3

Use the **show ospfv3** command to confirm that the device is acting as an ASBR and that the OSPFv3 Area 1 has been configured as a NSSA area.

```
Device#show ospfv3

 OSPFv3 1 address-family ipv4
 Router ID 3.3.3.3
```

```
 Supports NSSA (compatible with RFC 1587)
 It is an autonomous system boundary router
 Redistributing External Routes from,
    static
 Router is not originating router-LSAs with maximum metric
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of areas in this router is 1. 0 normal 0 stub 1 nssa
 Graceful restart helper support enabled
 Reference bandwidth unit is 100 mbps
 RFC1583 compatibility enabled
    Area 1
        Number of interfaces in this area is 1
        It is a NSSA area
        Configured to translate Type-7 LSAs, inactive (RFC3101 support disabled)
        Perform type-7/type-5 LSA translation, suppress forwarding address
        Area has no authentication
        SPF algorithm last executed 00:00:07.160 ago
        SPF algorithm executed 3 times
        Area ranges are
        Number of LSA 3. Checksum Sum 0x0245F0
        Number of opaque link LSA 0. Checksum Sum 0x000000
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

The table below describes the significant **show ip ospf** display fields and their descriptions.

Table 5: show ospfv3 Field Descriptions

Field	Description
Supports NSSA (compatible with RFC 1587)	Specifies that RFC 1587 is active or that the OSPFv3 NSSA area is RFC 1587 compatible.
Configured to translate Type-7 LSAs, inactive (RFC3101 support disabled)	Specifies that the OSPFv3 NSSA area has an ABR device configured to act as a forced translator of Type 7 LSAs. However, it is inactive because RFC 3101 is disabled.

The output of the router LSA in LSDB displays Nt-Bit if it is set in the header of LSA.

```
Router Link States (Area 1)

  LS age: 94
  Options: (N-Bit, R-bit, DC-Bit, AF-Bit, Nt-Bit)
  LS Type: Router Links
  Link State ID: 0
  Advertising Router: 2.2.2.2
  LS Seq Number: 80000002
  Checksum: 0x8AD5
  Length: 56
  Area Border Router
```

```
AS Boundary Router
Unconditional NSSA translator
Number of Links: 2
```

The “Unconditional NSSA translator” line indicates that the NSSA ASBR device is as a forced NSSA LSA translator.

Additional references

Table 6: Standards and RFCs

Standard/RFC	Title
RFC 3101	The OSPF Not-So-Stubby Area (NSSA) Option
RFC 1587	The OSPF NSSA Option

CHAPTER 21

Prefix Suppression Support for OSPFv3

Feature History for Prefix Suppression Support for OSPFv3

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	Prefix Suppression Support for OSPFv3: The OSPFv3 Prefix Suppression Support is an enhancement feature that allows devices to suppress the advertisement of IPv4 and IPv6 prefixes for directly connected networks within their LSAs.	Cisco C9610 Series Smart Switches

Prefix Suppression Support for OSPFv3

The OSPFv3 Prefix Suppression Support is an enhancement feature that allows devices to suppress the advertisement of IPv4 and IPv6 prefixes for directly connected networks within their Link-State Advertisements (LSAs). In large OSPFv3 deployments, reducing the number of prefixes included in LSAs helps to improve OSPFv3 convergence times.

Why does prefix suppression work on OSPFv3 and not OSPFv2

In OSPFv3, the handling of addressing information has been decoupled from the core OSPF protocol. Unlike OSPFv2, where network addresses are embedded within primary LSA types such as Router-LSAs and

Network-LSAs, OSPFv3's Router-LSAs and Network-LSAs do not include network addresses. Instead, these LSAs focus solely on describing the network topology. This separation of topology and addressing information makes OSPFv3 flexible and network-protocol-independent, allowing it to support multiple address families more efficiently.

As a result, hiding prefixes in OSPFv3 is straightforward. Suppressed prefixes are simply excluded from the intra-area-prefix-LSA, the LSA type responsible for advertising both IPv4 and IPv6 prefixes within an area. Additionally, OSPFv3 can also communicate certain prefixes through link-LSAs, which are used to advertise a device's IPv6 link-local address and other link-specific information. This approach streamlines prefix management and enhances flexibility in large OSPFv3 deployments.

Note A device only suppresses prefixes that are locally configured on its own interfaces; it does not suppress or filter prefixes that are learned from other routers through link-LSAs. This ensures that only the device's local prefix advertisements are affected by suppression, while information received from neighboring routers remains unchanged.

Prefix suppression methods

There are two prefix suppression methods available.

- Global Prefix Suppression:

  You can decrease OSPFv3 convergence time by configuring the OSPFv3 process to suppress the advertisement of all IPv4 and IPv6 prefixes. This is done using the **prefix-suppression** command in either router configuration mode or address-family configuration mode.

- Interface-Level Prefix Suppression:

  You can prevent a specific OSPFv3 interface from advertising its IP network to neighbors by using the **ipv6 ospf prefix-suppression** or **ospfv3 prefix-suppression** command in interface configuration mode.

How to configure Prefix Suppression Support for OSPFv3

This sections shows how to configure prefix suppression support for OSPFv3.

Configure Prefix Suppression Support of the OSPFv3 Process

Perform this task to suppress prefixes of the OSPFv3 process on a global level. This task explains how to suppress prefixes on a global level and not based on IPv6 address-family.

Before you begin

Prefixes that are associated with loopbacks, secondary IP addresses, and passive interfaces are not suppressed because typical network designs require prefixes to remain reachable.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospfv3** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospfv3 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **prefix-suppression**

Example:

```
Device(config-router)# prefix-suppression
```

Prevents OSPFv3 from advertising all IPv4 and IPv6 prefixes, except for prefixes associated with loopbacks, secondary IP addresses, and passive interfaces.

Step 5 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

Step 6 **show ospfv3**

Example:

```
Device# show ospfv3
```

Displays general information about OSPFv3 routing processes.

Configure Prefix Suppression Support of the OSPFv3 Process with IPv6 address family

Perform this task to suppress prefixes of the OSPFv3 process on a global level. This task explains how to suppress prefixes based on IPv6 address-family.

Before you begin

Prefixes that are associated with loopbacks, secondary IP addresses, and passive interfaces are not suppressed because typical network designs require prefixes to remain reachable.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router ospfv3** *process-id* [**vrf** *vrf-name*]

Example:

```
Device(config)# router ospfv3 15
```

Enables OSPF routing and enters router configuration mode.

- *process-id*: The process ID is an internally used identification parameter that is locally assigned. Each OSPF process has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **vrf**: Indicates that the OSPF process is being configured for a specific VRF.
- *vrf-name*: Specifies the name of the VRF for which this OSPF process is being created.

Step 4 **address-family ipv6 unicast**

Example:

```
Device(config-router)# address-family ipv6 unicast
```

Enters IPv6 address family configuration mode for OSPFv3.

Step 5 **prefix-suppression**

Example:

```
Device(config-router)# prefix-suppression
```

Prevents OSPFv3 from advertising all IPv4 and IPv6 prefixes, except for prefixes associated with loopbacks, secondary IP addresses, and passive interfaces.

Step 6 **end**

Example:

```
Device(config-router)# end
```

Returns to privileged EXEC mode.

Step 7 **show ospfv3**

Example:

```
Device# show ospfv3
```

Displays general information about OSPFv3 routing processes.

Configure Prefix Suppression Support on a per-interface basis

Perform this task to suppress prefixes of the OSPFv3 process on a global level.

Before you begin

If you have globally suppressed IPv4 and IPv6 prefixes from connected IP networks by configuring the **prefix-suppression** router configuration command, the interface configuration command takes precedence over the router configuration command.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *interface-id*

Example:

```
Device(config)# interface gigabitethernet 1/0/1
```

Enters interface configuration mode, and specifies the Layer 3 interface to configure.

Step 4 Choose one of the following:

- **ipv6 ospf prefix-suppression** [**disable**]
- **ospfv3 prefix-suppression disable**

Example:

```
Device(config-if)# ipv6 ospf prefix-suppression
OR
Device(config-if)# ospfv3 1 prefix-suppression disable
```

Prevents OSPFv3 from advertising IPv4 and IPv6 prefixes that belong to a specific interface, except those that are associated with secondary IP addresses.

Step 5 **end**

Example:

```
Device(config-if)# end
```

Returns to privileged EXEC mode.

Step 6 **show ospfv3**

Example:

```
Device# show ospfv3
```

Displays general information about OSPFv3 routing processes.

Troubleshoot IPv4 and IPv6 prefix suppression

Perform this task to check the configuration for IPv4 and IPv6 prefix suppression.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **debug ospfv3 lsa-generation**

Example:

```
Device# debug ospfv3 lsa-generation
```

Displays information about each OSPFv3 LSA that is generated.

Step 3 **debug condition interface** *interface-type interface-number*[**dlci** *dlci*] [**vc** {*vci* | *vpi* | *vci*}]

Example:

```
Device# debug condition interface ethernet1/0/1
```

Limits output for some debug commands on the basis of the interface or virtual circuit.

- *interface-type*: The type of interface (e.g., GigabitEthernet, Serial, FastEthernet).
- *interface-number:* The specific interface number (e.g., 0/0, 1/0/1).
- **dlci** *dlci*: (Optional) Specifies a Data Link Connection Identifier, used with Frame Relay interfaces.
- **vc** {*vci* | *vpi* | *vci*}: Specifies virtual circuit identifiers, used for ATM or other VC-based interfaces.

Step 4 **show debugging**

Example:

```
Device# show debugging
```

Displays information about the types of debugging that are enabled for your device.

Step 5 **show logging** [**slot** *slot-number* | **summary**]

Example:

```
Device# show logging
```

Displays the state of syslog and the contents of the standard system logging buffer.

Configuration example for Prefix Suppression Support for OSPFv3

The following example shows how to configure prefix suppression support for OSPFv3 in router configuration mode:

```
Device> enable
Device# configure terminal
Device(config)# router ospfv3 15
Device(config-router)# prefix-suppression
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# router-id 0.0.0.6
Device(config-router-af)# exit-address-family
```

The following example shows how to configure prefix suppression support for OSPFv3 in router configuration mode:

```
Device> enable
Device# configure terminal
Device(config)# router ospfv3 15
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# router-id 0.0.0.6
Device(config-router)# prefix-suppression
Device(config-router-af)# exit-address-family
```

The following example shows how to configure prefix suppression support for OSPFv3 in interface configuration mode:

```
Device> enable
Device# configure terminal
Device(config)# interface Ethernet1/0/1
Device(config-if)# ip address 10.0.0.1 255.255.255.0
Device(config-if)# ipv6 address 2001:201::201/64
```

```
Device(config-if)# ipv6 enable
Device(config-if)# ospfv3 prefix-suppression
Device(config-if)# ospfv3 1 ipv4 area 0
Device(config-if)# ospfv3 1 ipv6 area 0
Device(config-if)# end
```