Implementing OSPF on Cisco ASR 9000 Series Routers

Open Shortest Path First (OSPF) is an Interior Gateway Protocol (IGP) developed by the OSPF working group of the Internet Engineering Task Force (IETF). This module describes how to implement OSPF on Cisco ASR 9000 Series Aggregation Services Routers.

Designed expressly for IP networks, OSPF supports IP subnetting and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending and receiving packets.

OSPF Version 3 (OSPFv3) expands on OSPF Version 2, providing support for IPv6 routing prefixes. The term “OSPF” implies both versions of the routing protocol, unless otherwise noted.

For more information about OSPF on Cisco IOS XR software and complete descriptions of the OSPF commands listed in this module, see the “Related Documents” section of this module. To locate documentation for other commands that might appear during execution of a configuration task, search online in the Cisco ASR 9000 Series Routers master command index.

Feature History for Implementing OSPF on Cisco ASR 9000 Series Routers

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.7.2</td>
<td>This feature was introduced on Cisco ASR 9000 Series Routers.</td>
</tr>
</tbody>
</table>

Contents

- Prerequisites for Implementing OSPF, page RC-166
- Information About Implementing OSPF, page RC-166
- How to Implement OSPF, page RC-187
- Configuration Examples for Implementing OSPF, page RC-248
- Where to Go Next, page RC-255
- Additional References, page RC-255
Prerequisites for Implementing OSPF

The following are prerequisites for implementing OSPF on Cisco IOS XR software:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

- Configuration tasks for OSPFv3 assume that you are familiar with IPv6 addressing and basic configuration. See the Implementing Network Stack IPv4 and IPv6 on Cisco ASR 9000 Series Routers module of the Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Configuration Guide for information on IPv6 routing and addressing.

- Before you enable OSPFv3 on an interface, you must perform the following tasks:
  - Complete the OSPF network strategy and planning for your IPv6 network. For example, you must decide whether multiple areas are required.
  - Enable IPv6 on the interface.

- Configuring authentication (IP Security) is an optional task. If you choose to configure authentication, you must first decide whether to configure plain text or Message Digest 5 (MD5) authentication, and whether the authentication applies to an entire area or specific interfaces.

Information About Implementing OSPF

To implement OSPF you need to understand the following concepts:

- OSPF Functional Overview, page RC-167
- Key Features Supported in the Cisco IOS XR OSPF Implementation, page RC-168
- Comparison of Cisco IOS XR OSPFv3 and OSPFv2, page RC-169
- OSPF Hierarchical CLI and CLI Inheritance, page RC-169
- OSPF Routing Components, page RC-170
- OSPF Process and Router ID, page RC-172
- Supported OSPF Network Types, page RC-173
- Route Authentication Methods for OSPF, page RC-173
- Neighbors and Adjacency for OSPF, page RC-174
- Designated Router (DR) for OSPF, page RC-174
- Default Route for OSPF, page RC-174
- Link-State Advertisement Types for OSPF Version 2, page RC-175
- Link-State Advertisement Types for OSPFv3, page RC-175
- Virtual Link and Transit Area for OSPF, page RC-176
- OSPFv2 Sham Link Support for MPLS VPN, page RC-177
- Route Redistribution for OSPF, page RC-179
- OSPF Shortest Path First Throttling, page RC-179
- Nonstop Forwarding for OSPF Version 2, page RC-180
OSPF Functional Overview

OSPF is a routing protocol for IP. It is a link-state protocol, as opposed to a distance-vector protocol. A link-state protocol makes its routing decisions based on the states of the links that connect source and destination machines. The state of the link is a description of that interface and its relationship to its neighboring networking devices. The interface information includes the IP address of the interface, network mask, type of network to which it is connected, routers connected to that network, and so on. This information is propagated in various types of link-state advertisements (LSAs).

A router stores the collection of received LSA data in a link-state database. This database includes LSA data for the links of the router. The contents of the database, when subjected to the Dijkstra algorithm, extract data to create an OSPF routing table. The difference between the database and the routing table is that the database contains a complete collection of raw data; the routing table contains a list of shortest paths to known destinations through specific router interface ports.

OSPF is the IGP of choice because it scales to large networks. It uses areas to partition the network into more manageable sizes and to introduce hierarchy in the network. A router is attached to one or more areas in a network. All of the networking devices in an area maintain the same complete database information about the link states in their area only. They do not know about all link states in the network. The agreement of the database information among the routers in the area is called convergence.

At the intradomain level, OSPF can import routes learned using Intermediate System-to-Intermediate System (IS-IS). OSPF routes can also be exported into IS-IS. At the interdomain level, OSPF can import routes learned using Border Gateway Protocol (BGP). OSPF routes can be exported into BGP.

Unlike Routing Information Protocol (RIP), OSPF does not provide periodic routing updates. On becoming neighbors, OSPF routers establish an adjacency by exchanging and synchronizing their databases. After that, only changed routing information is propagated. Every router in an area advertises the costs and states of its links, sending this information in an LSA. This state information is sent to all OSPF neighbors one hop away. All the OSPF neighbors, in turn, send the state information unchanged. This flooding process continues until all devices in the area have the same link-state database.

To determine the best route to a destination, the software sums all of the costs of the links in a route to a destination. After each router has received routing information from the other networking devices, it runs the shortest path first (SPF) algorithm to calculate the best path to each destination network in the database.

The networking devices running OSPF detect topological changes in the network, flood link-state updates to neighbors, and quickly converge on a new view of the topology. Each OSPF router in the network soon has the same topological view again. OSPF allows multiple equal-cost paths to the same destination. Since all link-state information is flooded and used in the SPF calculation, multiple equal cost paths can be computed and used for routing.
On broadcast and nonbroadcast multiaccess (NBMA) networks, the designated router (DR) or backup DR performs the LSA flooding. On point-to-point networks, flooding simply exits an interface directly to a neighbor.

OSPF runs directly on top of IP; it does not use TCP or User Datagram Protocol (UDP). OSPF performs its own error correction by means of checksums in its packet header and LSAs.

In OSPFv3, the fundamental concepts are the same as OSPF Version 2, except that support is added for the increased address size of IPv6. New LSA types are created to carry IPv6 addresses and prefixes, and the protocol runs on an individual link basis rather than on an individual IP-subnet basis.

OSPF typically requires coordination among many internal routers: Area Border Routers (ABRs), which are routers attached to multiple areas, and Autonomous System Border Routers (ASBRs) that export reroutes from other sources (for example, IS-IS, BGP, or static routes) into the OSPF topology. At a minimum, OSPF-based routers or access servers can be configured with all default parameter values, no authentication, and interfaces assigned to areas. If you intend to customize your environment, you must ensure coordinated configurations of all routers.

**Key Features Supported in the Cisco IOS XR OSPF Implementation**

The Cisco IOS XR implementation of OSPF conforms to the OSPF Version 2 and OSPF Version 3 specifications detailed in the Internet RFC 2328 and RFC 2740, respectively.

The following key features are supported in the Cisco IOS XR implementation:

- Hierarchy—CLI hierarchy is supported.
- Inheritance—CLI inheritance is supported.
- Stub areas—Definition of stub areas is supported.
- NSF—Nonstop forwarding is supported.
- SPF throttling—Shortest path first throttling feature is supported.
- LSA throttling—LSA throttling feature is supported.
- Fast convergence—SPF and LSA throttle timers are set, configuring fast convergence. The OSPF LSA throttling feature provides a dynamic mechanism to slow down LSA updates in OSPF during network instability. LSA throttling also allows faster OSPF convergence by providing LSA rate limiting in milliseconds.
- Route redistribution—Routes learned using any IP routing protocol can be redistributed into any other IP routing protocol.
- Authentication—Plain text and MD5 authentication among neighboring routers within an area is supported.
- Routing interface parameters—Configurable parameters supported include interface output cost, retransmission interval, interface transmit delay, router priority, router “dead” and hello intervals, and authentication key.
- Virtual links—Virtual links are supported.
- Not-so-stubby area (NSSA)—RFC 1587 is supported.
- OSPF over demand circuit—RFC 1793 is supported.
Comparison of Cisco IOS XR OSPFv3 and OSPFv2

Much of the OSPFv3 protocol is the same as in OSPFv2. OSPFv3 is described in RFC 2740. The key differences between the Cisco IOS XR OSPFv3 and OSPFv2 protocols are as follows:

- OSPFv3 expands on OSPFv2 to provide support for IPv6 routing prefixes and the larger size of IPv6 addresses.
- When using an NBMA interface in OSPFv3, users must manually configure the router with the list of neighbors. Neighboring routers are identified by the link local address of the attached interface of the neighbor.
- Unlike in OSPFv2, multiple OSPFv3 processes can be run on a link.
- LSAs in OSPFv3 are expressed as “prefix and prefix length” instead of “address and mask.”
- The router ID is a 32-bit number with no relationship to an IPv6 address.

OSPF Hierarchical CLI and CLI Inheritance

Cisco IOS XR software introduces new OSPF configuration fundamentals consisting of hierarchical CLI and CLI inheritance.

Hierarchical CLI is the grouping of related network component information at defined hierarchical levels such as at the router, area, and interface levels. Hierarchical CLI allows for easier configuration, maintenance, and troubleshooting of OSPF configurations. When configuration commands are displayed together in their hierarchical context, visual inspections are simplified. Hierarchical CLI is intrinsic for CLI inheritance to be supported.

With CLI inheritance support, you need not explicitly configure a parameter for an area or interface. In Cisco IOS XR, the parameters of interfaces in the same area can be exclusively configured with a single command, or parameter values can be inherited from a higher hierarchical level—such as from the area configuration level or the router ospf configuration levels.

For example, the hello interval value for an interface is determined by this precedence “IF” statement:

- If the `hello interval` command is configured at the interface configuration level, then use the interface configured value, else
- If the `hello interval` command is configured at the area configuration level, then use the area configured value, else
- If the `hello interval` command is configured at the router ospf configuration level, then use the router ospf configured value, else
- Use the default value of the command.

Tip

Understanding hierarchical CLI and CLI inheritance saves you considerable configuration time. See the “Configuring Authentication at Different Hierarchical Levels for OSPF Version 2” section on page RC-198 to understand how to implement these fundamentals. In addition, Cisco IOS XR examples are provided in the “Configuration Examples for Implementing OSPF” section on page RC-248.
OSPF Routing Components

Before implementing OSPF, you must know what the routing components are and what purpose they serve. They consist of the autonomous system, area types, interior routers, ABRs, and ASBRs. Figure 1 illustrates the routing components in an OSPF network topology.

Figure 1  OSPF Routing Components

Autonomous Systems

The autonomous system is a collection of networks, under the same administrative control, that share routing information with each other. An autonomous system is also referred to as a routing domain. Figure 1 shows two autonomous systems: 109 and 65200. An autonomous system can consist of one or more OSPF areas.

Areas

Areas allow the subdivision of an autonomous system into smaller, more manageable networks or sets of adjacent networks. As shown in Figure 1, autonomous system 109 consists of three areas: Area 0, Area 1, and Area 2.

OSPF hides the topology of an area from the rest of the autonomous system. The network topology for an area is visible only to routers inside that area. When OSPF routing is within an area, it is called intra-area routing. This routing limits the amount of link-state information flood into the network, reducing routing traffic. It also reduces the size of the topology information in each router.
Also, the routers within an area cannot see the detailed network topology outside the area. Because of this restricted view of topological information, you can control traffic flow between areas and reduce routing traffic when the entire autonomous system is a single routing domain.

**Backbone Area**

A backbone area is responsible for distributing routing information between multiple areas of an autonomous system. OSPF routing occurring outside of an area is called *interarea routing*.

The backbone itself has all properties of an area. It consists of ABRs, routers, and networks only on the backbone. As shown in *Figure 1*, Area 0 is an OSPF backbone area. Any OSPF backbone area has a reserved area ID of 0.0.0.0.

**Stub Area**

A stub area is an area that does not accept or detailed network information external to the area. A stub area typically has only one router that interfaces the area to the rest of the autonomous system. The stub ABR advertises a single default route to external destinations into the stub area. Routers within a stub area use this route for destinations outside the area and the autonomous system. This relationship conserves LSA database space that would otherwise be used to store external LSAs flooded into the area. In *Figure 1*, Area 2 is a stub area that is reached only through ABR 2. Area 0 cannot be a stub area.

**Not-so-Stubby Area**

A Not-so-Stubby Area (NSSA) is similar to the stub area. NSSA does not flood Type 5 external LSAs from the core into the area, but can import autonomous system external routes in a limited fashion within the area.

NSSA allows importing of Type 7 autonomous system external routes within an NSSA area by redistribution. These Type 7 LSAs are translated into Type 5 LSAs by NSSA ABRs, which are flooded throughout the whole routing domain. Summarization and filtering are supported during the translation.

Use NSSA to simplify administration if you are a network administrator that must connect a central site using OSPF to a remote site that is using a different routing protocol. Before NSSA, the connection between the corporate site border router and remote router could not be run as an OSPF stub area because routes for the remote site could not be redistributed into a stub area, and two routing protocols needed to be maintained. A simple protocol like RIP was usually run and handled the redistribution. With NSSA, you can extend OSPF to cover the remote connection by defining the area between the corporate router and remote router as an NSSA. Area 0 cannot be an NSSA.

**Routers**

The OSPF network is composed of ABRs, ASBRs, and interior routers.

**Area Border Routers**

An area border routers (ABR) is a router with multiple interfaces that connect directly to networks in two or more areas. An ABR runs a separate copy of the OSPF algorithm and maintains separate routing data for each area that is attached to, including the backbone area. ABRs also send configuration summaries for their attached areas to the backbone area, which then distributes this information to other OSPF areas in the autonomous system. In *Figure 1*, there are two ABRs. ABR 1 interfaces Area 1 to the backbone area. ABR 2 interfaces the backbone Area 0 to Area 2, a stub area.
Autonomous System Boundary Routers (ASBR)

An autonomous system boundary router (ASBR) provides connectivity from one autonomous system to another system. ASBRs exchange their autonomous system routing information with boundary routers in other autonomous systems. Every router inside an autonomous system knows how to reach the boundary routers for its autonomous system.

ASBRs can import external routing information from other protocols like BGP and redistribute them as AS-external (ASE) Type 5 LSAs to the OSPF network. If the Cisco IOS XR router is an ASBR, you can configure it to advertise VIP addresses for content as autonomous system external routes. In this way, ASBRs flood information about external networks to routers within the OSPF network.

ASBR routes can be advertised as a Type 1 or Type 2 ASE. The difference between Type 1 and Type 2 is how the cost is calculated. For a Type 2 ASE, only the external cost (metric) is considered when multiple paths to the same destination are compared. For a Type 1 ASE, the combination of the external cost and cost to reach the ASBR is used. Type 2 external cost is the default and is always more costly than an OSPF route and used only if no OSPF route exists.

Interior Routers

An interior router (such as R1 in Figure 1) is attached to one area (for example, all the interfaces reside in the same area).

OSPF Process and Router ID

An OSPF process is a logical routing entity running OSPF in a physical router. This logical routing entity should not be confused with the logical routing feature that allows a system administrator (known as the Cisco IOS XR Owner) to partition the physical box into separate routers.

A physical router can run multiple OSPF processes, although the only reason to do so would be to connect two or more OSPF domains. Each process has its own link-state database. The routes in the routing table are calculated from the link-state database. One OSPF process does not share routes with another OSPF process unless the routes are redistributed.

Each OSPF process is identified by a router ID. The router ID must be unique across the entire routing domain. OSPFv2 obtains a router ID from the following sources, in order of decreasing preference:

- The 32-bit numeric value specified by the OSPF `router-id` command in OSPF router configuration mode. (This value can be any 32-bit value. It is not restricted to the IPv4 addresses assigned to interfaces on this router and need not be a routable IPv4 address.)

- If there is no explicit router-id assigned, OSPF uses a global router ID. The global router ID is determined by OSPF and is the lowest IPv4 address assigned to any loopback interface on this router. If no such loopbacks exist, OSPF assigns 0.0.0.0.

> Note The global router ID cannot be explicitly configured.

- The primary IPv4 address of an interface over which this OSPF process is running. The first interface address in the OSPF interface is selected.

We recommend that the router ID be set by the `router-id` command in router configuration mode. Separate OSPF processes could share the same router ID, in which case they cannot reside in the same OSPF routing domain.
Supported OSPF Network Types

OSPF classifies different media into the following types of networks:

- NBMA networks
- Point-to-point networks (POS)
- Broadcast networks (Gigabit Ethernet)
- Point-to-multipoint

You can configure your Cisco IOS XR network as either a broadcast or an NBMA network. Using this feature, you can configure broadcast networks as NBMA networks when, for example, you have routers in your network that do not support multicast addressing.

Route Authentication Methods for OSPF

OSPF Version 2 supports two types of authentication: plain text authentication and MD5 authentication. By default, no authentication is enabled (referred to as null authentication in RFC 2178).

OSPF Version 3 supports all types of authentication except key rollover.

Plain Text Authentication

Plain text authentication (also known as Type 1 authentication) uses a password that travels on the physical medium and is easily visible to someone that does not have access permission and could use the password to infiltrate a network. Therefore, plain text authentication does not provide security. It might protect against a faulty implementation of OSPF or a misconfigured OSPF interface trying to send erroneous OSPF packets.

MD5 Authentication

MD5 authentication provides a means of security. No password travels on the physical medium. Instead, the router uses MD5 to produce a message digest of the OSPF packet plus the key, which is sent on the physical medium. Using MD5 authentication prevents a router from accepting unauthorized or deliberately malicious routing updates, which could compromise your network security by diverting your traffic.

Note

MD5 authentication supports multiple keys, requiring that a key number be associated with a key.

See the “OSPF Authentication Message Digest Management” section on page RC-186.

Authentication Strategies

Authentication can be specified for an entire process or area, or on an interface or a virtual link. An interface or virtual link can be configured for only one type of authentication, not both. Authentication configured for an interface or virtual link overrides authentication configured for the area or process.

If you intend for all interfaces in an area to use the same type of authentication, you can configure fewer commands if you use the authentication command in the area configuration submode (and specify the message-digest keyword if you want the entire area to use MD5 authentication). This strategy requires fewer commands than specifying authentication for each interface.
Key Rollover

To support the changing of an MD5 key in an operational network without disrupting OSPF adjacencies (and hence the topology), a key rollover mechanism is supported. As a network administrator configures the new key into the multiple networking devices that communicate, some time exists when different devices are using both a new key and an old key. If an interface is configured with a new key, the software sends two copies of the same packet, each authenticated by the old key and new key. The software tracks which devices start using the new key, and the software stops sending duplicate packets after it detects that all of its neighbors are using the new key. The software then discards the old key. The network administrator must then remove the old key from each the configuration file of each router.

Neighbors and Adjacency for OSPF

Routers that share a segment (Layer 2 link between two interfaces) become neighbors on that segment. OSPF uses the hello protocol as a neighbor discovery and keep alive mechanism. The hello protocol involves receiving and periodically sending hello packets out each interface. The hello packets list all known OSPF neighbors on the interface. Routers become neighbors when they see themselves listed in the hello packet of the neighbor. After two routers are neighbors, they may proceed to exchange and synchronize their databases, which creates an adjacency. On broadcast and NBMA networks all neighboring routers have an adjacency.

Designated Router (DR) for OSPF

On point-to-point and point-to-multipoint networks, the Cisco IOS XR software floods routing updates to immediate neighbors. No DR or backup DR (BDR) exists; all routing information is flooded to each router.

On broadcast or NBMA segments only, OSPF minimizes the amount of information being exchanged on a segment by choosing one router to be a DR and one router to be a BDR. Thus, the routers on the segment have a central point of contact for information exchange. Instead of each router exchanging routing updates with every other router on the segment, each router exchanges information with the DR and BDR. The DR and BDR relay the information to the other routers. On broadcast network segments the number of OSPF packets is further reduced by the DR and BDR sending such OSPF updates to a multicast IP address that all OSPF routers on the network segment are listening on.

The software looks at the priority of the routers on the segment to determine which routers are the DR and BDR. The router with the highest priority is elected the DR. If there is a tie, then the router with the higher router ID takes precedence. After the DR is elected, the BDR is elected the same way. A router with a router priority set to zero is ineligible to become the DR or BDR.

Default Route for OSPF

Type 5 (ASE) LSAs are generated and flooded to all areas except stub areas. For the routers in a stub area to be able to route packets to destinations outside the stub area, a default route is injected by the ABR attached to the stub area.

The cost of the default route is 1 (default) or is determined by the value specified in the `default-cost` command.
Link-State Advertisement Types for OSPF Version 2

Each of the following LSA types has a different purpose:

- **Router LSA (Type 1)**—Describes the links that the router has within a single area, and the cost of each link. These LSAs are flooded within an area only. The LSA indicates if the router can compute paths based on quality of service (QoS), whether it is an ABR or ASBR, and if it is one end of a virtual link. Type 1 LSAs are also used to advertise stub networks.

- **Network LSA (Type 2)**—Describes the link state and cost information for all routers attached to a multiaccess network segment. This LSA lists all the routers that have interfaces attached to the network segment. It is the job of the designated router of a network segment to generate and track the contents of this LSA.

- **Summary LSA for ABRs (Type 3)**—Advertises internal networks to routers in other areas (interarea routes). Type 3 LSAs may represent a single network or a set of networks aggregated into one prefix. Only ABRs generate summary LSAs.

- **Summary LSA for ASBRs (Type 4)**—Advertises an ASBR and the cost to reach it. Routers that are trying to reach an external network use these advertisements to determine the best path to the next hop. ABRs generate Type 4 LSAs.

- **Autonomous system external LSA (Type 5)**—Redistributes routes from another autonomous system, usually from a different routing protocol into OSPF.

- **Autonomous system external LSA (Type 7)**—Provides for carrying external route information within an NSSA. Type 7 LSAs may be originated by and advertised throughout an NSSA. NSSAs do not receive or originate Type 5 LSAs. Type 7 LSAs are advertised only within a single NSSA. They are not flooded into the backbone area or into any other area by border routers.

- **Intra-area-prefix LSAs (Type 9)**—A router can originate multiple intra-area-prefix LSAs for every router or transit network, each with a unique link-state ID. The link-state ID for each intra-area-prefix LSA describes its association to either the router LSA or network LSA and contains prefixes for stub and transit networks.

- **Area local scope (Type 10)**—Opaque LSAs are not flooded past the borders of their associated area.

- **Link-state (Type 11)**—The LSA is flooded throughout the AS. The flooding scope of Type 11 LSAs are equivalent to the flooding scope of AS-external (Type 5) LSAs. Similar to Type 5 LSAs, the LSA is rejected if a Type 11 opaque LSA is received in a stub area from a neighboring router within the stub area. Type 11 opaque LSAs have these attributes:
  - LSAs are flooded throughout all transit areas.
  - LSAs are not flooded into stub areas from the backbone.
  - LSAs are not originated by routers into their connected stub areas.

Link-State Advertisement Types for OSPFv3

Each of the following LSA types has a different purpose:

- **Router LSA (Type 1)**—Describes the link state and costs of a the router link to the area. These LSAs are flooded within an area only. The LSA indicates whether the router is an ABR or ASBR and if it is one end of a virtual link. Type 1 LSAs are also used to advertise stub networks. In OSPFv3, these LSAs have no address information and are network protocol independent. In OSPFv3, router interface information may be spread across multiple router LSAs. Receivers must concatenate all router LSAs originated by a given router before running the SPF calculation.
Network LSA (Type 2)—Describes the link state and cost information for all routers attached to a multiaccess network segment. This LSA lists all OSPF routers that have interfaces attached to the network segment. Only the elected designated router for the network segment can generate and track the network LSA for the segment. In OSPFv3, network LSAs have no address information and are network-protocol-independent.

Interarea-prefix LSA for ABRs (Type 3)—Advertises internal networks to routers in other areas (interarea routes). Type 3 LSAs may represent a single network or set of networks aggregated into one prefix. Only ABRs generate Type 3 LSAs. In OSPFv3, addresses for these LSAs are expressed as “prefix and prefix length” instead of “address and mask.” The default route is expressed as a prefix with length 0.

Interarea-router LSA for ASBRs (Type 4)—Advertises an ASBR and the cost to reach it. Routers that are trying to reach an external network use these advertisements to determine the best path to the next hop. ABRs generate Type 4 LSAs.

Autonomous system external LSA (Type 5)—Redistributes routes from another autonomous system, usually from a different routing protocol into OSPF. In OSPFv3, addresses for these LSAs are expressed as “prefix and prefix length” instead of “address and mask.” The default route is expressed as a prefix with length 0.

Autonomous system external LSA (Type 7)—Provides for carrying external route information within an NSSA. Type 7 LSAs may be originated by and advertised throughout an NSSA. NSSAs do not receive or originate Type 5 LSAs. Type 7 LSAs are advertised only within a single NSSA. They are not flooded into the backbone area or into any other area by border routers.

Link LSA (Type 8)—Has link-local flooding scope and is never flooded beyond the link with which it is associated. Link LSAs provide the link-local address of the router to all other routers attached to the link or network segment, inform other routers attached to the link of a list of IPv6 prefixes to associate with the link, and allow the router to assert a collection of Options bits to associate with the network LSA that is originated for the link.

Intra-area-prefix LSAs (Type 9)—A router can originate multiple intra-area-prefix LSAs for every router or transit network, each with a unique link-state ID. The link-state ID for each intra-area-prefix LSA describes its association to either the router LSA or network LSA and contains prefixes for stub and transit networks.

An address prefix occurs in almost all newly defined LSAs. The prefix is represented by three fields: Prefix Length, Prefix Options, and Address Prefix. In OSPFv3, addresses for these LSAs are expressed as “prefix and prefix length” instead of “address and mask.” The default route is expressed as a prefix with length 0.

Inter-area-prefix and intra-area-prefix LSAs carry all IPv6 prefix information that, in IPv4, is included in router LSAs and network LSAs. The Options field in certain LSAs (router LSAs, network LSAs, interarea-router LSAs, and link LSAs) has been expanded to 24 bits to provide support for OSPF in IPv6. In OSPFv3, the sole function of link-state ID in interarea-prefix LSAs, interarea-router LSAs, and autonomous system external LSAs is to identify individual pieces of the link-state database. All addresses or router IDs that are expressed by the link-state ID in OSPF Version 2 are carried in the body of the LSA in OSPFv3.

Virtual Link and Transit Area for OSPF

In OSPF, routing information from all areas is first summarized to the backbone area by ABRs. The same ABRs, in turn, propagate such received information to their attached areas. Such hierarchical distribution of routing information requires that all areas be connected to the backbone area (Area 0).
Occasions might exist for which an area must be defined, but it cannot be physically connected to Area 0. Examples of such an occasion might be if your company makes a new acquisition that includes an OSPF area, or if Area 0 itself is partitioned.

In the case in which an area cannot be connected to Area 0, you must configure a virtual link between that area and Area 0. The two endpoints of a virtual link are ABRs, and the virtual link must be configured in both routers. The common nonbackbone area to which the two routers belong is called a transit area. A virtual link specifies the transit area and the router ID of the other virtual endpoint (the other ABR).

A virtual link cannot be configured through a stub area or NSSA.

Figure 2 illustrates a virtual link from Area 3 to Area 0.

**Figure 2 Virtual Link to Area 0**

---

**OSPFv2 Sham Link Support for MPLS VPN**

In an MPLS VPN environment, several VPN client sites can be connected in the same OSPF area. If these sites are connected over a backdoor link (intra-area link) and connected over the VPN backbone, all traffic passes over the backdoor link instead of over the VPN backbone, because provider edge routers advertise OSPF routes learned over the VPN backbone as inter-area or external routes that are less preferred than intra-area routes advertised over backdoor links.

To correct this default OSPF behavior in an MPLS VPN, configure a sham link between two provider edge (PE) routers to connect the sites through the MPLS VPN backbone. A sham link represents an intra-area (unnumbered point-to-point) connection between PE routers. All other routers in the area see the sham link and use it to calculate intra-area shortest path first (SPF) routes to the remote site. A cost must be configured with each sham link to determine whether traffic is sent over the backdoor link or sham link.
Configured source and destination addresses serve as the endpoints of the sham link. The source and destination IP addresses must belong to the VRF and must be advertised by Border Gateway Protocol (BGP) as host routes to remote PE routers. The sham-link endpoint addresses should not be advertised by OSPF.

For example, Figure 3 shows three client sites, each with backdoor links. Because each site runs OSPF within Area 1 configuration, all routing between the sites follows the intra-area path across the backdoor links instead of over the MPLS VPN backbone.

If the backdoor links between the sites are used only for backup purposes, default route selection over the backbone link is not acceptable as it creates undesirable traffic flow. To establish the desired path selection over the MPLS backbone, an additional OSPF intra-area (sham link) link between the ingress and egress PEs routers must be created.

A sham link is required between any two VPN sites that belong to the same OSPF area and share an OSPF backdoor link. If no backdoor link exists between sites, no sham link is required.

Figure 3  Backdoor Paths Between OSPF Client Sites
Figure 4 shows an MPLS VPN topology where a sham link configuration is necessary. A VPN client has three sites, each with a backdoor link. Two sham links are configured, one between PE1 and PE2 and another between PE2 and PE3. A sham link is not required between PE1 and PE3, because there is no backdoor link between these sites.

When a sham link is configured between the PE routers, the PE routers can populate the virtual routing and forwarding (VRF) table with the OSPF routes learned over the sham link. These OSPF routes have a larger administrative distance than BGP routes. If BGP routes are available, they are preferred over these OSPF routes with the high administrative distance.

Route Redistribution for OSPF

Redistribution allows different routing protocols to exchange routing information. This technique can be used to allow connectivity to span multiple routing protocols. It is important to remember that the redistribute command controls redistribution into an OSPF process and not from OSPF. See the “Configuration Examples for Implementing OSPF” section on page RC-248 for an example of route redistribution for OSPF.

OSPF Shortest Path First Throttling

OSPF SPF throttling makes it possible to configure SPF scheduling in millisecond intervals and to potentially delay SPF calculations during network instability. SPF is scheduled to calculate the Shortest Path Tree (SPT) when there is a change in topology. One SPF run may include multiple topology change events.
The interval at which the SPF calculations occur is chosen dynamically and based on the frequency of topology changes in the network. The chosen interval is within the boundary of the user-specified value ranges. If network topology is unstable, SPF throttling calculates SPF scheduling intervals to be longer until topology becomes stable.

SPF calculations occur at the interval set by the `timers throttle spf` command. The wait interval indicates the amount of time to wait until the next SPF calculation occurs. Each wait interval after that calculation is twice as long as the previous interval until the interval reaches the maximum wait time specified.

The SPF timing can be better explained using an example. In this example, the start interval is set at 5 milliseconds (ms), initial wait interval at 1000 ms, and maximum wait time at 90,000 ms.

```
timers spf 5 1000 90000
```

Figure 5 shows the intervals at which the SPF calculations occur as long as at least one topology change event is received in a given wait interval.

![Figure 5](image)

**SPF Calculation Intervals Set by the timers spf Command**

Notice that the wait interval between SPF calculations doubles when at least one topology change event is received during the previous wait interval. After the maximum wait time is reached, the wait interval remains the same until the topology stabilizes and no event is received in that interval.

If the first topology change event is received after the current wait interval, the SPF calculation is delayed by the amount of time specified as the start interval. The subsequent wait intervals continue to follow the dynamic pattern.

If the first topology change event occurs after the maximum wait interval begins, the SPF calculation is again scheduled at the start interval and subsequent wait intervals are reset according to the parameters specified in the `timers throttle spf` command. Notice in Figure 6 that a topology change event was received after the start of the maximum wait time interval and that the SPF intervals have been reset.

![Figure 6](image)

**Timer Intervals Reset After Topology Change Event**

Nonstop Forwarding for OSPF Version 2

Cisco IOS XR NSF for OSPF Version 2 allows for the forwarding of data packets to continue along known routes while the routing protocol information is being restored following a failover. With NSF, peer networking devices do not experience routing flaps. During failover, data traffic is forwarded...
through intelligent line cards while the standby Route Switch Processor (RSP) assumes control from the failed RSP. The ability of line cards to remain up through a failover and to be kept current with the Forwarding Information Base (FIB) on the active RSP is key to Cisco IOS XR NSF operation.

Routing protocols, such as OSPF, run only on the active RSP or DRSP and receive routing updates from their neighbor routers. When an OSPF NSF-capable router performs an RSP failover, it must perform two tasks to resynchronize its link-state database with its OSPF neighbors. First, it must relearn the available OSPF neighbors on the network without causing a reset of the neighbor relationship. Second, it must reacquire the contents of the link-state database for the network.

As quickly as possible after an RSP failover, the NSF-capable router sends an OSPF NSF signal to neighboring NSF-aware devices. This signal is in the form of a link-local LSA generated by the failed-over router. Neighbor networking devices recognize this signal as a cue that the neighbor relationship with this router should not be reset. As the NSF-capable router receives signals from other routers on the network, it can begin to rebuild its neighbor list.

After neighbor relationships are reestablished, the NSF-capable router begins to resynchronize its database with all of its NSF-aware neighbors. At this point, the routing information is exchanged between the OSPF neighbors. After this exchange is completed, the NSF-capable device uses the routing information to remove stale routes, update the RIB, and update the FIB with the new forwarding information. OSPF on the router and the OSPF neighbors are now fully converged.

**Graceful Restart for OSPFv3**

In the current release, various restart scenarios in the control plane of an IPv6-enabled router can disrupt data forwarding. The OSPFv3 Graceful Restart feature can preserve the data plane capability in the following circumstances:

- RP failure, resulting in a switchover to the backup processor
- Planned OSPFv3 process restart, such as software upgrade or downgrade
- Unplanned OSPFv3 process restart, such as a process crash

This feature supports nonstop data forwarding on established routes while the OSPFv3 routing protocol is restarting. (Therefore, this feature enhances high availability of IPv6 forwarding.)

**Modes of Graceful Restart Operation**

The two operational modes that a router can be in for this feature are restart mode and helper mode. Restart mode occurs when the OSPFv3 process is doing a graceful restart. Helper mode refers to the neighbor routers that continue to forward traffic on established OSPFv3 routes while OSPFv3 is restarting on a neighboring router.

**Restart Mode**

When the OSPFv3 process starts up, it determines whether it must attempt a graceful restart. The determination is based on whether graceful restart was previously enabled. (OSPFv3 does not attempt a graceful restart upon the first-time startup of the router.) When OSPFv3 graceful restart is enabled, it changes the purge timer in the RIB to a nonzero value. See the “Configuring OSPFv3 Graceful Restart” section on page RC-222 for descriptions of how to enable and configure graceful restart.

During a graceful restart, the router does not populate OSPFv3 routes in the RIB. It tries to bring up full adjacencies with the fully adjacent neighbors that OSPFv3 had before the restart. Eventually, the OSPFv3 process indicates to the RIB that it has converged, either for the purpose of terminating the graceful restart (for any reason) or because it has completed the graceful restart.
The following are general details about restart mode. More detailed information on behavior and certain restrictions and requirements appears in the “Graceful Restart Requirements and Restrictions” section on page RC-182.

- If OSPFv3 attempts a restart too soon after the most recent restart, the OSPFv3 process is most likely crashing repeatedly, so the new graceful restart stops running. To control the period between allowable graceful restarts, use the `graceful-restart interval` command.

- When OSPFv3 starts a graceful restart with the first interface that comes up, a timer starts running to limit the duration (or lifetime) of the graceful restart. You can configure this period with the `graceful-restart lifetime` command. On each interface that comes up, a grace LSA (Type 11) is flooded to indicate to the neighboring routers that this router is attempting graceful restart. The neighbors enter into helper mode.

- The designated router and backup designated router check of the hello packet received from the restarting neighbor is bypassed, because it might not be valid.

### Helper Mode

Helper mode is enabled by default. When a (helper) router receives a grace LSA (Type 11) from a router that is attempting a graceful restart, the following events occur:

- If helper mode has been disabled through the `graceful-restart helper disable` command, the router drops the LSA packet.

- If helper mode is enabled, the router enters helper mode if all of the following conditions are met:
  - The local router itself is not attempting a graceful restart.
  - The local (helping) router has full adjacency with the sending neighbor.
  - The value of `lsage` (link state age) in the received LSA is less than the requested grace period.
  - The sender of the grace LSA is the same as the originator of the grace LSA.

- Upon entering helper mode, a router performs its helper function for a specific period of time. This time period is the lifetime value from the router that is in restart mode—minus the value of `lsage` in the received grace LSA. If the graceful restart succeeds in time, the helper’s timer is stopped before it expires. If the helper’s timer does expire, the adjacency to the restarting router is brought down, and normal OSPFv3 functionality resumes.

- The dead timer is not honored by the router that is in helper mode.

- A router in helper mode ceases to perform the helper function in any of the following cases:
  - The helper router is able to bring up a FULL adjacency with the restarting router.
  - The local timer for the helper function expires.

### Graceful Restart Requirements and Restrictions

The requirements for supporting the Graceful Restart feature include:

- Cooperation of a router’s neighbors during a graceful restart. In relation to the router on which OSPFv3 is restarting, each router is called a helper.

- All neighbors of the router that does a graceful restart must be capable of doing a graceful restart.

- A graceful restart does not occur upon the first-time startup of a router.

- OSPFv3 neighbor information and database information are not check-pointed.

- An OSPFv3 process rebuilds adjacencies after it restarts.
To ensure consistent databases after a restart, the OSPFv3 configuration must be identical to the configuration before the restart. (This requirement applies to self-originated information in the local database.) A graceful restart can fail if configurations change during the operation. In this case, data forwarding would be affected. OSPFv3 resumes operation by regenerating all its LSAs and resynchronizing its database with all its neighbors.

Although IPv6 FIB tables remain unchanged during a graceful restart, these tables eventually mark the routes as stale through the use of a holddown timer. Enough time is allowed for the protocols to rebuild state information and converge.

The router on which OSPFv3 is restarting must send OSPFv3 hellos within the dead interval of the process restart. Protocols must be able to retain adjacencies with neighbors before the adjacency dead timer expires. The default for the dead timer is 40 seconds. If hellos do not arrive on the adjacency before the dead timer expires, the router takes down the adjacency. The OSPFv3 Graceful Restart feature does not function properly if the dead timer is configured to be less than the time required to send hellos after the OSPFv3 process restarts.

Simultaneous graceful restart sessions on multiple routers are not supported on a single network segment. If a router determines that multiple routers are in restart mode, it terminates any local graceful restart operation.

This feature utilizes the available support for changing the purge time of existing OSPFv3 routes in the Routing Information Base (RIB). When graceful restart is enabled, the purge timer is set to 90 seconds by default. If graceful restart is disabled, the purge timer setting is 0.

This feature has an associated grace LSA. This link-scope LSA is Type 11.

According to the RFC, the OSPFv3 process should flush all old, self-originated LSAs during a restart. With the Graceful Restart feature, however, the router delays this flushing of unknown self-originated LSAs during a graceful restart. OSPFv3 can learn new information and build new LSAs to replace the old LSAs. When the delay is over, all old LSAs are flushed.

If graceful restart is enabled, the adjacency creation time of all the neighbors is saved in the system database (SysDB). The purpose for saving the creation time is so that OSPFv3 can use the original adjacency creation time to display the uptime for that neighbor after the restart.

**Warm Standby and Nonstop Routing for OSPF Version 2**

OSPFv2 warm standby provides high availability across RSP switchovers. With warm standby extensions, each process running on the active RSP has a corresponding standby process started on the standby RSP. A standby OSPF process can send and receive OSPF packets with no performance impact to the active OSPF process.

Nonstop routing (NSR) allows an RSP failover, process restart, or in-service upgrade to be invisible to peer routers and ensures that there is minimal performance or processing impact. Routing protocol interactions between routers are not impacted by NSR. NSR is built on the warm standby extensions. NSR alleviates the requirement for Cisco NSF and IETF graceful restart protocol extensions.

**Multicast-Intact Support for OSPF**

The multicast-intact feature provides the ability to run multicast routing (PIM) when IGP shortcuts are configured and active on the router. Both OSPFv2 and IS-IS support the multicast-intact feature.
You can enable multicast-intact in the IGP when multicast routing protocols (PIM) are configured and IGP shortcuts are configured on the router. IGP shortcuts are MPLS tunnels that are exposed to IGP. The IGP routes IP traffic over these tunnels to destinations that are downstream from the egress router of the tunnel (from an SPF perspective). PIM cannot use IGP shortcuts for propagating PIM joins, because reverse path forwarding (RPF) cannot work across a unidirectional tunnel.

When you enable multicast-intact on an IGP, the IGP publishes a parallel or alternate set of equal-cost next hops for use by PIM. These next hops are called mcast-intact next hops. The mcast-intact next hops have the following attributes:

- They are guaranteed not to contain any IGP shortcuts.
- They are not used for unicast routing but are used only by PIM to look up an IPv4 next-hop to a PIM source.
- They are not published to the FIB.
- When multicast-intact is enabled on an IGP, all IPv4 destinations that were learned through link-state advertisements are published with a set equal-cost mcast-intact next hops to the RIB. This attribute applies even when the native next hops have no IGP shortcuts.

In OSPF, the max-paths (number of equal-cost next hops) limit is applied separately to the native and mcast-intact next hops. The number of equal cost mcast-intact next hops is the same as that configured for the native next hops.

## Load Balancing in OSPF Version 2 and OSPFv3

When a router learns multiple routes to a specific network by using multiple routing processes (or routing protocols), it installs the route with the lowest administrative distance in the routing table. Sometimes the router must select a route from among many learned by using the same routing process with the same administrative distance. In this case, the router chooses the path with the lowest cost (or metric) to the destination. Each routing process calculates its cost differently; the costs may need to be manipulated to achieve load balancing.

OSPF performs load balancing automatically. If OSPF finds that it can reach a destination through more than one interface and each path has the same cost, it installs each path in the routing table. The only restriction on the number of paths to the same destination is controlled by the `maximum-paths` (OSPF) command. On Cisco ASR 9000 Series Routers, the range for maximum paths is from 1 to 4, and the default number of maximum paths is 4.

## Multi-Area Adjacency for OSPF Version 2

The multi-area adjacency feature for OSPFv2 allows a link to be configured on the primary interface in more than one area so that the link could be considered as an intra-area link in those areas and configured as a preference over more expensive paths.

This feature establishes a point-to-point unnumbered link in an OSPF area. A point-to-point link provides a topological path for that area, and the primary adjacency uses the link to advertise the link consistent with draft-ietf-ospf-multi-area-adj-06.

The following are multi-area interface attributes and limitations:

- Exists as a logical construct over an existing primary interface for OSPF; however, the neighbor state on the primary interface is independent of the multi-area interface.
- Establishes a neighbor relationship with the corresponding multi-area interface on the neighboring router. A mixture of multi-area and primary interfaces is not supported.
- Advertises an unnumbered point-to-point link in the router link state advertisement (LSA) for the corresponding area when the neighbor state is full.
- Created as a point-to-point network type and is not configurable.
- Created only on native point-to-point interfaces, such as Gigabit Ethernet, including VLANs, bundles, and VLAN over Link Bundles (VoLB), or serial.
- Inherits the Bidirectional Forwarding Detection (BFD) characteristics from its primary interface. BFD is not configurable under a multi-area interface; however, it is configurable under the primary interface.

The multi-area interface inherits the interface characteristics from its primary interface, but some interface characteristics can be configured under the multi-area interface configuration mode as shown below:

```
rp/0/rsp0/cpu0:router(config-ospf-ar)# multi-area-interface gigabitethernet 0/1/0/3
rp/0/rsp0/cpu0:router(config-ospf-ar-mif)#?
```

```
authentication    Enable authentication
authentication-key Authentication password (key)
commit            Commit the configuration changes to running
cost              Interface cost
database-filter   Filter OSPF LSA during synchronization and flooding
dead-interval     Interval after which a neighbor is declared dead
describe          Describe a command without taking real actions
distribute-list   Filter networks in routing updates
do                Run an exec command
exit              Exit from this submode
hello-interval    Time between HELLO packets
message-digest-key Message digest authentication password (key)
mtu-ignore        Enable/Disable ignoring of MTU in DBD packets
no                Negate a command or set its defaults
packetsize        Customize size of OSPF packets up to MTU
pwd                Commands used to reach current submode
retransmit-interval Time between retransmitting lost link state advertisements
root              Exit to the global configuration mode
show              Show contents of configuration
transmit-delay    Estimated time needed to send link-state update packet
```

**Label Distribution Protocol IGP Auto-configuration for OSPF**

Label Distribution Protocol (LDP) Interior Gateway Protocol (IGP) auto-configuration simplifies the procedure to enable LDP on a set of interfaces used by an IGP instance, such as OSPF. LDP IGP auto-configuration can be used on a large number of interfaces (for example, when LDP is used for transport in the core) and on multiple OSPF instances simultaneously.

This feature supports the IPv4 unicast address family for the default VPN routing and forwarding (VRF) instance.

LDP IGP auto-configuration can also be explicitly disabled on an individual interface basis under LDP using the `igp auto-config disable` command. This allows LDP to receive all OSPF interfaces minus the ones explicitly disabled.

See the *Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide* for information on configuring LDP IGP auto-configuration.
OSPF Authentication Message Digest Management

All OSPF routing protocol exchanges are authenticated and the method used can vary depending on how authentication is configured. When using cryptographic authentication, the OSPF routing protocol uses the Message Digest 5 (MD5) authentication algorithm to authenticate packets transmitted between neighbors in the network. For each OSPF protocol packet, a key is used to generate and verify a message digest that is appended to the end of the OSPF packet. The message digest is a one-way function of the OSPF protocol packet and the secret key. Each key is identified by the combination of interface used and the key identification. An interface may have multiple keys active at any time.

To manage the rollover of keys and enhance MD5 authentication for OSPF, you can configure a container of keys called a *keychain* with each key comprising the following attributes: generate/accept time, key identification, and authentication algorithm.

GTSM TTL Security Mechanism for OSPF

OSPF is a link state protocol that requires networking devices to detect topological changes in the network, flood Link State Advertisement (LSA) updates to neighbors, and quickly converge on a new view of the topology. However, during the act of receiving LSAs from neighbors, network attacks can occur, because there are no checks that unicast or multicast packets are originating from a neighbor that is one hop away or multiple hops away over virtual links.

For virtual links, OSPF packets travel multiple hops across the network; hence, the TTL value can be decremented several times. For these type of links, a minimum TTL value must be allowed and accepted for multiple-hop packets.

To filter network attacks originating from invalid sources traveling over multiple hops, the Generalized TTL Security Mechanism (GTSM), RFC 3682, is used to prevent the attacks. GTSM filters link-local addresses and allows for only one-hop neighbor adjacencies through the configuration of TTL value 255. The TTL value in the IP header is set to 255 when OSPF packets are originated, and checked on the received OSPF packets against the default GTSM TTL value 255 or the user configured GTSM TTL value, blocking unauthorized OSPF packets originated from TTL hops away.

Path Computation Element for OSPFv2

A PCE is an entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints.

PCE is accomplished when a PCE address and client is configured for MPLS-TE. PCE communicates its PCE address and capabilities to OSPF then OSPF packages this information in the PCE Discovery type-length-value (TLV) (Type 2) and reoriginates the RI LSA. OSPF also includes the Router Capabilities TLV (Type 1) in all its RI LSAs. The PCE Discovery TLV contains the PCE address sub-TLV (Type 1) and the Path Scope Sub-TLV (Type 2).

The PCE Address Sub-TLV specifies the IP address that must be used to reach the PCE. It should be a loop-back address that is always reachable, this TLV is mandatory, and must be present within the PCE Discovery TLV. The Path Scope Sub-TLV indicates the PCE path computation scopes, which refers to the PCE ability to compute or participate in the computation of intra-area, inter-area, inter-AS or inter-layer TE LSPs.
PCE extensions to OSPFv2 include support for the Router Information Link State Advertisement (RI LSA). OSPFv2 is extended to receive all area scopes (LSA Types 9, 10, and 11). However, OSPFv2 originates only area scope Type 10.

For detailed information for the Path Computation Element feature see the *Implementing MPLS Traffic Engineering on Cisco ASR 9000 Series Routers* module of the *Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide* and the following IETF drafts:

- draft-ietf-ospf-cap-09
- draft-ietf-pce-disco-proto-ospf-00

### How to Implement OSPF

This section contains the following procedures:

- Enabling OSPF, page RC-188 (required)
- Configuring Stub and Not-So-Stubby Area Types, page RC-190 (optional)
- Configuring Neighbors for Nonbroadcast Networks, page RC-193 (optional)
- Configuring Authentication at Different Hierarchical Levels for OSPF Version 2, page RC-198 (optional)
- Controlling the Frequency That the Same LSA Is Originated or Accepted for OSPF, page RC-201 (optional)
- Creating a Virtual Link with MD5 Authentication to Area 0 for OSPF, page RC-203 (optional)
- Summarizing Subnetwork LSAs on an OSPF ABR, page RC-207 (optional)
- Redistributing Routes from One IGP into OSPF, page RC-209 (optional)
- Configuring OSPF Shortest Path First Throttling, page RC-213 (optional)
- Configuring Nonstop Forwarding Specific to Cisco for OSPF Version 2, page RC-215 (optional)
- Configuring OSPF Version 2 for MPLS Traffic Engineering, page RC-217 (optional)
- Configuring OSPFv3 Graceful Restart, page RC-222 (optional)
- Configuring an OSPFv2 Sham Link, page RC-225 (optional)
- Enabling Nonstop Routing for OSPFv2, page RC-228 (optional)
- Enabling Multicast-intact for OSPFv2, page RC-230 (optional)
- Associating Interfaces to a VRF, page RC-231 (optional)
- Configuring OSPF as a Provider Edge to Customer Edge (PE-CE) Protocol, page RC-233 (optional)
- Creating Multiple OSPF Instances (OSPF Process and a VRF), page RC-236 (optional)
- Configuring Multi-area Adjacency, page RC-237 (optional)
- Configuring Label Distribution Protocol IGP Auto-configuration for OSPF, page RC-239 (optional)
- Configuring Authentication Message Digest Management for OSPF, page RC-240 (optional)
- Configuring Generalized TTL Security Mechanism (GTSM) for OSPF, page RC-244 (optional)
- Verifying OSPF Configuration and Operation, page RC-247 (optional)
Enabling OSPF

This task explains how to perform the minimum OSPF configuration on your router that is to enable an OSPF process with a router ID, configure a backbone or nonbackbone area, and then assign one or more interfaces on which OSPF runs.

Prerequisites

Although you can configure OSPF before you configure an IP address, no OSPF routing occurs until at least one IP address is configured.

SUMMARY STEPS

1. configure
2. router ospf process-name
   or
   router ospfv3 process-name
3. router-id {router-id}
4. area area-id
5. interface type instance
6. Repeat Step 5 for each interface that uses OSPF.
7. log adjacency changes {detail} {enable | disable}
8. end
   or
   commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

Example:
RP/0/RSP0/CPU0:router# configure

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>router ospf process-name</td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</td>
</tr>
<tr>
<td></td>
<td>or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>router ospfv3 process-name</td>
<td>Enables OSPFv3 routing for the specified routing process and places the router in router ospfv3 configuration mode.</td>
</tr>
</tbody>
</table>

Example:
RP/0/RSP0/CPU0:router(config)# router ospf 1
or
RP/0/RSP0/CPU0:router(config)# router ospfv3 1

Note The process-name argument is any alphanumeric string no longer than 40 characters.
### Command or Action

<table>
<thead>
<tr>
<th>Step 3</th>
<th><strong>router-id</strong> <em>(router-id)</em></th>
</tr>
</thead>
</table>

**Example:**

RP/0/RSP0/CPU0:router(config-ospf)# router-id 192.168.4.3

**Purpose:** Configures a router ID for the OSPF process.

**Note** We recommend using a stable IP address as the router ID.

<table>
<thead>
<tr>
<th>Step 4</th>
<th><strong>area</strong> <em>area-id</em></th>
</tr>
</thead>
</table>

**Example:**

RP/0/RSP0/CPU0:router(config-ospf)# area 0

**Purpose:** Enters area configuration mode and configures an area for the OSPF process.

- Backbone areas have an area ID of 0.
- Nonbackbone areas have a nonzero area ID.
- The *area-id* argument can be entered in dotted-decimal or IPv4 address notation, such as area 1000 or area 0.0.3.232. However, you must choose one form or the other for an area. We recommend using the IPv4 address notation.

<table>
<thead>
<tr>
<th>Step 5</th>
<th><strong>interface</strong> <em>type instance</em></th>
</tr>
</thead>
</table>

**Example:**

RP/0/RSP0/CPU0:router(config-ospf-ar)# interface GigabitEthernet0/1/0/3

**Purpose:** Enters interface configuration mode and associates one or more interfaces for the area configured in Step 4.

<table>
<thead>
<tr>
<th>Step 6</th>
<th>Repeat Step 5 for each interface that uses OSPF.</th>
</tr>
</thead>
</table>
Configuring Stub and Not-So-Stubby Area Types

This task explains how to configure the stub area and the NSSA for OSPF.

SUMMARY STEPS

1. configure
2. router ospf process-name
   or
   router ospfv3 process-name
3. router-id (router-id)
4. area area-id
5. stub [no-summary]
   or
   nssa [no-redistribution] [default-information-originate] [no-summary]
6. stub
   or
   nssa
7. default-cost cost
8. end
   or
   commit
9. Repeat this task on all other routers in the stub area or NSSA.

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
</tbody>
</table>
| **Step 2** router ospf process-name  
or router ospfv3 process-name | Enables OSPF routing for the specified routing process and places the router in router configuration mode.  
or Enables OSPFv3 routing for the specified routing process and places the router in router ospfv3 configuration mode. |
| **Example:** RP/0/RSP0/CPU0:router(config)# router ospf 1  
or RP/0/RSP0/CPU0:router(config)# router ospfv3 1 | |
| **Note** The process-name argument is any alphanumeric string no longer than 40 characters. |
| **Step 3** router-id (router-id) | Configures a router ID for the OSPF process.  
| **Example:** RP/0/RSP0/CPU0:router(config-ospf)# router-id  
192.168.4.3 | |
| **Note** We recommend using a stable IP address as the router ID. |
| **Step 4** area area-id | Enters area configuration mode and configures a nonbackbone area for the OSPF process.  
| **Example:** RP/0/RSP0/CPU0:router(config-ospf)# area 1 |  
- The area-id argument can be entered in dotted-decimal or IPv4 address notation, such as area 1000 or area 0.0.3.232. However, you must choose one form or the other for an area. We recommend using the IPv4 address notation. |
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>stub [no-summary] or nssa [no-redistribution] [default-information-originate] [no-summary]</td>
<td>Defines the nonbackbone area as a stub area.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar)# stub no summary or RP/0/RSP0/CPU0:router(config-ospf-ar)# nssa no-redistribution</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>stub or nssa</td>
<td>(Optional) Turns off the options configured for stub and NSSA areas.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar)# stub or RP/0/RSP0/CPU0:router(config-ospf-ar)# nssa</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>default-cost cost</td>
<td>(Optional) Specifies a cost for the default summary route sent into a stub area or an NSSA.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar)# default-cost 15</td>
<td></td>
</tr>
</tbody>
</table>

### Purpose

- See the “Configuring Stub and Not-So-Stubby Area Types” section on page RC-190.
- Specify the no-summary keyword to further reduce the number of LSAs sent into a stub area. This keyword prevents the ABR from sending summary link-state advertisements (Type 3) in the stub area.
- Defines an area as an NSSA.
- See the “Configuring Stub and Not-So-Stubby Area Types” section on page RC-190.

- If you configured the stub and NSSA areas using the optional keywords (no-summary, no-redistribution, default-information-originate, and no-summary) in Step 5, you must now reissue the stub and nssa commands without the keywords—rather than using the no form of the command.
- For example, the no nssa default-information-originate form of the command changes the NSSA area into a normal area that inadvertently brings down the existing adjacencies in that area.
- Use this command only on ABRs attached to the NSSA. Do not use it on any other routers in the area.
- The default cost is 1.
Configuring Neighbors for Nonbroadcast Networks

This task explains how to configure neighbors for a nonbroadcast network. This task is optional.

Prerequisites

Configuring NBMA networks as either broadcast or nonbroadcast assumes that there are virtual circuits from every router to every router or fully meshed network.

SUMMARY STEPS

- configure
- router ospf `process-name`
  or
  router ospfv3 `process-name`
- router-id `{router-id}`
- area `area-id`
- network `{broadcast | non-broadcast | point-to-multipoint [non-broadcast] | point-to-point}`
- `dead-interval seconds`
- `hello-interval seconds`
- `interface type instance`
9. `neighbor ip-address [priority number] [poll-interval seconds] [cost number]` or
   `neighbor ipv6-link-local-address [priority number] [poll-interval seconds] [cost number] [database-filter [all]]`

10. Repeat Step 9 for all neighbors on the interface.

11. `exit`

12. `interface type instance`

13. `neighbor ip-address [priority number] [poll-interval seconds] [cost number] [database-filter [all]]`
   or
   `neighbor ipv6-link-local-address [priority number] [poll-interval seconds] [cost number] [database-filter [all]]`

14. Repeat Step 13 for all neighbors on the interface.

15. `end`
   or
   `commit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure</code></td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RSP0/CPU0:router# configure</code></td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>router ospf process-name</code> or</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RSP0/CPU0:router(config)# router ospf 1</code> or</td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config)# router ospfv3 1</code></td>
</tr>
<tr>
<td></td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</td>
</tr>
<tr>
<td>or</td>
<td>Enables OSPFv3 routing for the specified routing process and places the router in router ospfv3 configuration mode.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>The <code>process-name</code> argument is any alphanumeric string no longer than 40 characters.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>router-id (router-id)</code></td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RSP0/CPU0:router(config-ospf)# router-id 192.168.4.3</code></td>
</tr>
<tr>
<td></td>
<td>Configures a router ID for the OSPF process.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>We recommend using a stable IP address as the router ID.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>area area-id</code></td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RSP0/CPU0:router(config-ospf)# area 0</code></td>
</tr>
<tr>
<td></td>
<td>Enters area configuration mode and configures an area for the OSPF process.</td>
</tr>
<tr>
<td><em>The example configures a backbone area.</em></td>
<td></td>
</tr>
<tr>
<td><em>The <code>area-id</code> argument can be entered in dotted-decimal or IPv4 address notation, such as area 1000 or area 0.0.3.232. However, you must choose one form or the other for an area. We recommend using the IPv4 address notation.</em></td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 5</th>
<th>network (broadcast</th>
<th>non-broadcast</th>
<th>(point-to-multipoint</th>
<th>non-broadcast)</th>
<th>point-to-point))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar)# network non-broadcast</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purpose</td>
<td>Configures the OSPF network type to a type other than the default for a given medium.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• The example sets the network type to NBMA.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Step 6 | dead-interval seconds |
| Example: | RP/0/RSP0/CPU0:router(config-ospf-ar)# dead-interval 40 |
| Purpose | (Optional) Sets the time to wait for a hello packet from a neighbor before declaring the neighbor down. |

| Step 7 | hello-interval seconds |
| Example: | RP/0/RSP0/CPU0:router(config-ospf-ar)# hello-interval 10 |
| Purpose | (Optional) Specifies the interval between hello packets that OSPF sends on the interface. |

| Step 8 | interface type instance |
| Example: | RP/0/RSP0/CPU0:router(config-ospf-ar)# interface GigabitEthernet 0/2/0/0 |
| Purpose | Enters interface configuration mode and associates one or more interfaces for the area configured in Step 4. |
| • In this example, the interface inherits the nonbroadcast network type and the hello and dead intervals from the areas because the values are not set at the interface level. |
How to Implement OSPF

Step 9

**neighbor ip-address [priority number] [poll-interval seconds] [cost number]**
or

**neighbor ipv6-link-local-address [priority number] [poll-interval seconds] [cost number] [database-filter [all]]**

**Example:**
RP/0/RSP0/CPU0:router(config-ospf-ar-if)# neighbor 10.20.20.1 priority 3 poll-interval 15 or
RP/0/RSP0/CPU0:router(config-ospf-ar-if)# neighbor fe80::3203:a0ff:fe9d:f3fe

- Configures the IPv4 address of OSPF neighbors interconnecting to nonbroadcast networks.
- Configures the link-local IPv6 address of OSPFv3 neighbors.
  - The *ipv6-link-local-address* argument must be in the form documented in RFC 2373 in which the address is specified in hexadecimal using 16-bit values between colons.
  - The *priority* keyword notifies the router that this neighbor is eligible to become a DR or BDR. The priority value should match the actual priority setting on the neighbor router. The neighbor priority default value is zero. This keyword does not apply to point-to-multipoint interfaces.
  - The *poll-interval* keyword does not apply to point-to-multipoint interfaces. RFC 1247 recommends that this value be much larger than the hello interval. The default is 120 seconds (2 minutes).
  - Neighbors with no specific cost configured assumes the cost of the interface, based on the *cost* command. On point-to-multipoint interfaces, *cost number* is the only keyword and argument combination that works. The *cost* keyword does not apply to NBMA networks.
  - The *database-filter* keyword filters outgoing LSAs to an OSPF neighbor. If you specify the *all* keyword, incoming and outgoing LSAs are filtered. Use with extreme caution since filtering may cause the routing topology to be seen as entirely different between two neighbors, resulting in “black-holing” of data traffic or routing loops.

Step 10
Repeat Step 9 for all neighbors on the interface.

Step 11
**exit**

**Example:**
RP/0/RSP0/CPU0:router(config-ospf-ar-if)# exit

Enters area configuration mode.

Step 12
**interface type instance**

**Example:**
RP/0/RSP0/CPU0:router(config-ospf-ar)# interface GigabitEthernet 0/3/0/1

Enters interface configuration mode and associates one or more interfaces for the area configured in Step 4.
- In this example, the interface inherits the nonbroadcast network type and the hello and dead intervals from the areas because the values are not set at the interface level.
## Command or Action

<table>
<thead>
<tr>
<th>Command Syntax</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 13 neighbor ip-address [priority number]</td>
<td>Configures the IPv4 address of OSPF neighbors interconnecting to nonbroadcast networks.</td>
</tr>
<tr>
<td>Step 13 neighbor ipv6-link-local-address [priority number]</td>
<td>Configures the link-local IPv6 address of OSPFv3 neighbors.</td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar)# neighbor 10.34.16.6</td>
<td></td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar)# neighbor fe80::3203:a0ff:fe9d:f3f</td>
<td></td>
</tr>
</tbody>
</table>

### Example:

- **neighbor ip-address [priority number]**
  - Configures the IPv4 address of OSPF neighbors interconnecting to nonbroadcast networks.
  - The **ipv6-link-local-address** argument must be in the form documented in RFC 2373 in which the address is specified in hexadecimal using 16-bit values between colons.
  - The **priority** keyword notifies the router that this neighbor is eligible to become a DR or BDR. The priority value should match the actual priority setting on the neighbor router. The neighbor priority default value is zero. This keyword does not apply to point-to-multipoint interfaces.
  - The **poll-interval** keyword does not apply to point-to-multipoint interfaces. RFC 1247 recommends that this value be much larger than the hello interval. The default is 120 seconds (2 minutes).
  - Neighbors with no specific cost configured assumes the cost of the interface, based on the **cost** command. On point-to-multipoint interfaces, **cost number** is the only keyword and argument combination that works. The **cost** keyword does not apply to NBMA networks.
  - The **database-filter** keyword filters outgoing LSAs to an OSPF neighbor. If you specify the **all** keyword, incoming and outgoing LSAs are filtered. Use with extreme caution since filtering may cause the routing topology to be seen as entirely different between two neighbors, resulting in “black-holing” or routing loops.

### Purpose:

- Configures the IPv4 address of OSPF neighbors interconnecting to nonbroadcast networks.
- Configures the link-local IPv6 address of OSPFv3 neighbors.
Configuring Authentication at Different Hierarchical Levels for OSPF Version 2

This task explains how to configure MD5 (secure) authentication on the OSPF router process, configure one area with plain text authentication, and then apply one interface with clear text (null) authentication.

**Note**

Authentication configured at the interface level overrides authentication configured at the area level and the router process level. If an interface does not have authentication specifically configured, the interface inherits the authentication parameter value from a higher hierarchical level. See the “OSPF Hierarchical CLI and CLI Inheritance” section on page RC-169 for more information about hierarchy and inheritance.

**Prerequisites**

If you choose to configure authentication, you must first decide whether to configure plain text or MD5 authentication, and whether the authentication applies to all interfaces in a process, an entire area, or specific interfaces. See the “Route Authentication Methods for OSPF” section on page RC-173 for information about each type of authentication and when you should use a specific method for your network.

**SUMMARY STEPS**

1. configure
2. router ospf process-name
3. router-id (router-id)
4. authentication [message-digest [keychain keychain] | null]
5. `message-digest-key key-id md5 {key | clear key | encrypted key}
6. `area area-id
7. `interface type instance
8. Repeat Step 7 for each interface that must communicate, using the same authentication.
9. `exit
10. `area area-id
11. `authentication [message-digest [keychain keychain] | null]
12. `interface type instance
13. Repeat Step 12 for each interface that must communicate, using the same authentication.
14. `interface type instance
15. `authentication [message-digest [keychain keychain] | null]
16. `end
   or
   `commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 router ospf process-name</td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# router ospf 1</td>
<td></td>
</tr>
<tr>
<td>Step 3 router-id (router-id)</td>
<td>Configures a router ID for the OSPF process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# router-id 192.168.4.3</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# authentication message-digest</td>
<td></td>
</tr>
<tr>
<td>Step 5 message-digest-key key-id md5 {key</td>
<td>clear key</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# message-digest-key 4 md5 yourkey</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 6</strong> area area-id</td>
<td>Enters area configuration mode and configures a backbone area for the OSPF process.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf)# area 0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> interface type instance</td>
<td>Enters interface configuration mode and associates one or more interfaces to the backbone area.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf-ar)# interface GigabitEthernet 0/1/0/3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> Repeat Step 7 for each interface that must communicate, using the same authentication.</td>
<td>—</td>
</tr>
<tr>
<td><strong>Step 9</strong> exit</td>
<td>Enters area OSPF configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf-ar)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> area area-id</td>
<td>Enters area configuration mode and configures a nonbackbone area 1 for the OSPF process.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf)# area 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> authentication [message-digest [keychain keychain]</td>
<td>null]</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf-ar)# authentication</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> interface type instance</td>
<td>Enters interface configuration mode and associates one or more interfaces to the nonbackbone area 1 specified in Step 10.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf-ar)# interface GigabitEthernet 0/1/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong> Repeat Step 12 for each interface that must communicate using the same authentication.</td>
<td>—</td>
</tr>
<tr>
<td><strong>Step 14</strong> interface type instance</td>
<td>Enters interface configuration mode and associates one or more interfaces to a different authentication type.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf-ar)# interface GigabitEthernet 0/3/0/0</td>
<td></td>
</tr>
</tbody>
</table>
Controlling the Frequency That the Same LSA Is Originated or Accepted for OSPF

This task explains how to tune the convergence time of OSPF routes in the routing table when many LSAs need to be flooded in a very short time interval.

**SUMMARY STEPS**

1. configure
2. router ospf *process-name*  
   or  
   router ospfv3 *process-name*
3. router-id {router-id}
4. Perform Step 5 or Step 6 or both to control the frequency that the same LSA is originated or accepted.
5. timers lsa gen-interval *seconds*
6. timers lsa min-arrival *seconds*
7. timers lsa group-pacing *seconds*
## How to Implement OSPF

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Enters global configuration mode.</strong></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>router ospf</strong> process-name <strong>or</strong> router ospfv3 process-name**</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</strong> <strong>or</strong> <strong>Enables OSPFv3 routing for the specified routing process and places the router in router ospfv3 configuration mode.</strong> <strong>Note</strong> The <strong>process-name</strong> argument is any alphanumeric string no longer than 40 characters.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# router ospf 1 or RP/0/RSP0/CPU0:router(config)# router ospfv3 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>router-id</strong> <em>(router-id)</em></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Configures a router ID for the OSPF process.</strong> <strong>Note</strong> We recommend using a stable IP address as the router ID.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# router-id 192.168.4.3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>Perform Step 5 or Step 6 or both to control the frequency that the same LSA is originated or accepted.</strong> <strong>—</strong></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>timers lsa gen-interval</strong> <em>(seconds)</em></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Changes the minimum interval between the same OSPF LSAs that the router originates.</strong> <strong>• The default is 5 seconds for both OSPF and OSPFv3.</strong></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# timers lsa gen-interval 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>timers lsa min-arrival</strong> <em>(seconds)</em></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Limits the frequency that new processes of any particular OSPF Version 2 LSA can be accepted during flooding.</strong> <strong>• The default is 1 second.</strong></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# timers lsa min-arrival 2</td>
<td></td>
</tr>
</tbody>
</table>
### Creating a Virtual Link with MD5 Authentication to Area 0 for OSPF

This task explains how to create a virtual link to your backbone (area 0) and apply MD5 authentication. You must perform the steps described on both ABRs, one at each end of the virtual link. To understand virtual links, see the “Virtual Link and Transit Area for OSPF” section on page RC-176.

**Note**

After you explicitly configure area parameter values, they are inherited by all interfaces bound to that area—unless you override the values and configure them explicitly for the interface. An example is provided in the “Virtual Link Configured with MD5 Authentication for OSPF Version 2: Example” section on page RC-253.

#### Prerequisites

The following prerequisites must be met before creating a virtual link with MD5 authentication to area 0:

- You must have the router ID of the neighbor router at the opposite end of the link to configure the local router. You can execute the `show ospf` or `show ospfv3` command on the remote router to get its router ID.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 7**

`timers lsa group-pacing seconds`

**Example:**

RP/0/RSP0/CPU0:router(config-ospf)# timers lsa group-pacing 1000

Changes the interval at which OSPF link-state LSAs are collected into a group for flooding.

- The default is 240 seconds.

| **Step 8**

`end` or `commit`

**Example:**

RP/0/RSP0/CPU0:router(config-ospf)# end

Saves configuration changes.

- When you issue the `end` command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.

  - Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
For a virtual link to be successful, you need a stable router ID at each end of the virtual link. You do not want them to be subject to change, which could happen if they are assigned by default. (See the “OSPF Process and Router ID” section on page RC-172 for an explanation of how the router ID is determined.) Therefore, we recommend that you perform one of the following tasks before configuring a virtual link:

- Use the `router-id` command to set the router ID. This strategy is preferable.
- Configure a loopback interface so that the router has a stable router ID.

Before configuring your virtual link for OSPF Version 2, you must decide whether to configure plain text authentication, MD5 authentication, or no authentication (which is the default). Your decision determines whether you need to perform additional tasks related to authentication.

If you decide to configure plain text authentication or no authentication, see the `authentication` command provided in the OSPF Commands on Cisco ASR 9000 Series Routers module in the Cisco ASR 9000 Series Aggregation Services Router Routing Command Reference.

**SUMMARY STEPS**

1. `show ospf [process-name]`
   or
   `show ospfv3 [process-name]`
2. `configure`
3. `router ospf process-name`
   or
   `router ospfv3 process-name`
4. `router-id {router-id}`
5. `area area-id`
6. `virtual-link router-id`
7. `authentication message-digest`
8. `message-digest-key key-id md5 {key | clear key | encrypted key}`
9. Repeat all the steps in this task on the ABR that is at the other end of the virtual link. Specify the same key ID and key that you specified for the virtual link on this router.
10. `end`
    or
    `commit`
11. `show ospf [process-name] [area-id] virtual-links`
    or
    `show ospfv3 [process-name] virtual-links`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>(Optional) Displays general information about OSPF routing processes.</td>
</tr>
<tr>
<td><code>show ospf [process-name]</code> or <code>show ospfv3 [process-name]</code></td>
<td>- The output displays the router ID of the local router. You need this router ID to configure the other end of the link.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# show ospf or RP/0/RSP0/CPU0:router# show ospfv3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</td>
</tr>
<tr>
<td><code>router ospf process-name</code> or <code>router ospfv3 process-name</code></td>
<td>or Enables OSPFv3 routing for the specified routing process and places the router in router ospfv3 configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# router ospf 1 or RP/0/RSP0/CPU0:router(config)# router ospfv3 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Configures a router ID for the OSPF process.</td>
</tr>
<tr>
<td><code>router-id (router-id)</code></td>
<td><strong>Note</strong> We recommend using a stable IPv4 address as the router ID.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# router-id 192.168.4.3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Enters area configuration mode and configures a nonbackbone area for the OSPF process.</td>
</tr>
<tr>
<td><code>area area-id</code></td>
<td>- The <code>area-id</code> argument can be entered in dotted-decimal or IPv4 address notation, such as area 1000 or area 0.0.3.232. However, you must choose one form or the other for an area. We recommend using the IPv4 address notation.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# area 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Defines an OSPF virtual link.</td>
</tr>
<tr>
<td><code>virtual-link router-id</code></td>
<td>- See the “Virtual Link and Transit Area for OSPF” section on page RC-176.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar)# virtual-link 10.3.4.5</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Selects MD5 authentication for this virtual link.</td>
</tr>
<tr>
<td><code>authentication message-digest</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar-vl)# authentication message-digest</td>
<td></td>
</tr>
</tbody>
</table>
## How to Implement OSPF

### Command or Action
```
message-digest-key key-id md5 (key | clear key | encrypted key)
```

### Purpose
Defines an OSPF virtual link.
- See the “Virtual Link and Transit Area for OSPF” section on page RC-176 to understand a virtual link.
- The `key-id` argument is a number in the range from 1 to 255. The `key` argument is an alphanumeric string of up to 16 characters. The routers at both ends of the virtual link must have the same key identifier and key to be able to route OSPF traffic.
- The `authentication-key key` command is not supported for OSPFv3.
- Once the key is encrypted it must remain encrypted.

### Example:
```
RP/0/RSP0/CPU0:router(config-ospf-ar-vl)#
message-digest-key 4 md5 yourkey
```

### Step 8
Repeat all of the steps in this task on the ABR that is at the other end of the virtual link. Specify the same key ID and key that you specified for the virtual link on this router.

### Step 9
End or commit

### Example:
```
RP/0/RSP0/CPU0:router(config-ospf-ar-vl)#
end
```

### Purpose
Saves configuration changes.
- When you issue the `end` command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:

  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.

### Step 10
```
show ospf [process-name] [area-id] virtual-links
```

### Purpose
(Optional) Displays the parameters and the current state of OSPF virtual links.

### Example:
```
RP/0/RSP0/CPU0:router# show ospf 1 2 virtual-links
```

```
show ospfv3 [process-name] virtual-links
```

### Example:
```
RP/0/RSP0/CPU0:router# show ospfv3 1 virtual-links
```
Examples

In the following example, the `show ospfv3 virtual links` EXEC command verifies that the OSPF_VL0 virtual link to the OSPFV3 neighbor is up, the ID of the virtual link interface is 2, and the IPv6 address of the virtual link endpoint is 2003:3000::1.

```
RP/0/RSP0/CPU0:router# show ospfv3 virtual-links

Virtual Links for OSPFv3 1

Virtual Link OSPF_VL0 to router 10.0.0.3 is up
  Interface ID 2, IPv6 address 2003:3000::1
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 0.1.20.255, via interface GigabitEthernet 0/1/0/1, Cost of using 2
  Transmit Delay is 5 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  Hello due in 00:00:02
  Adjacency State FULL (Hello suppressed)
  Index 0/2/3, retransmission queue length 0, number of retransmission 1
  First 0(0)/0(0)/0(0) Next 0(0)/0(0)/0(0)
  Last retransmission scan length is 1, maximum is 1
  Last retransmission scan time is 0 msec, maximum is 0 msec

Check for lines:
Virtual Link OSPF_VL0 to router 10.0.0.3 is up
  Adjacency State FULL (Hello suppressed)

State is up and Adjacency State is FULL
```

Summarizing Subnetwork LSAs on an OSPF ABR

If you configured two or more subnetworks when you assigned your IP addresses to your interfaces, you might want the software to summarize (aggregate) into a single LSA all of the subnetworks that the local area advertises to another area. Such summarization would reduce the number of LSAs and thereby conserve network resources. This summarization is known as interarea route summarization. It applies to routes from within the autonomous system. It does not apply to external routes injected into OSPF by way of redistribution.

This task configures OSPF to summarize subnetworks into one LSA, by specifying that all subnetworks that fall into a range are advertised together. This task is performed on an ABR only.

**SUMMARY STEPS**

1. configure
2. `router ospf process-name`
   or
   `router ospfv3 process-name`
3. `router-id [router-id]`
4. `area area-id`
5. `range ip-address mask [advertise | not-advertise]`
   or
   `range ipv6-prefix/prefix-length [advertise | not-advertise]`
6. `interface type instance`
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</td>
</tr>
<tr>
<td><code>router ospf</code></td>
<td></td>
</tr>
<tr>
<td><code>process-name</code></td>
<td></td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td><code>router ospfv3</code></td>
<td></td>
</tr>
<tr>
<td><code>process-name</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# router ospf 1</td>
<td></td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# router ospfv3 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Configures a router ID for the OSPF process.</td>
</tr>
<tr>
<td><code>router-id</code></td>
<td></td>
</tr>
<tr>
<td><code>{router-id}</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# router-id 192.168.4.3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Enters area configuration mode and configures a nonbackbone area for the OSPF process.</td>
</tr>
<tr>
<td><code>area</code></td>
<td></td>
</tr>
<tr>
<td><code>area-id</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# area 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Consolidates and summarizes OSPF routes at an area boundary.</td>
</tr>
<tr>
<td><code>range ip-address</code></td>
<td></td>
</tr>
<tr>
<td><code>mask</code></td>
<td></td>
</tr>
<tr>
<td>`[advertise</td>
<td>not-advertise]`</td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td><code>range ipv6-prefix</code></td>
<td></td>
</tr>
<tr>
<td><code>prefix-length</code></td>
<td></td>
</tr>
<tr>
<td>`[advertise</td>
<td>not-advertise]`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar)# range 192.168.0.0 255.255.0.0 advertise</td>
<td></td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar)# range 4004:f000::/32 advertise</td>
<td></td>
</tr>
</tbody>
</table>
Implementing OSPF on Cisco ASR 9000 Series Routers

How to Implement OSPF

Redistributing Routes from One IGP into OSPF

This task redistributes routes from an IGP (could be a different OSPF process) into OSPF.

Prerequisites

For information about configuring routing policy, see the Implementing Routing Policy on Cisco ASR 9000 Series Routers module in the Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide (this publication).

SUMMARY STEPS

1. configure
2. router ospf process-name
   or
   router ospfv3 process-name
3. router-id {router-id}
4. redistribute protocol [process-id] {level-1 | level-1-2 | level-2} [metric metric-value] [metric-type type-value] [match {internal | external [1 | 2] | nssa-external [1 | 2]}] [tag tag-value] [route-map map-tag] [route-policy policy-tag]

Enters interface configuration mode and associates one or more interfaces to the area.

Saves configuration changes.

- When you issue the end command, the system prompts you to commit changes:
  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
  - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.
5. `summary-prefix address mask [not-advertise] [tag tag]`
   or
   `summary-prefix ipv6-prefix/prefix-length [not-advertise] [tag tag]`

6. `end`
   or
   `commit`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> router ospf process-name or router ospfv3 process-name</td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode. or Enables OSPFv3 routing for the specified routing process and places the router in router ospfv3 configuration mode.</td>
</tr>
<tr>
<td><strong>Note</strong> The <code>process-name</code> argument is any alphanumeric string no longer than 40 characters.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# router ospf 1 or RP/0/RSP0/CPU0:router(config)# router ospfv3 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router-id (router-id)</td>
<td>Configures a router ID for the OSPF process.</td>
</tr>
<tr>
<td><strong>Note</strong> We recommend using a stable IPv4 address as the router ID.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf)# router-id 192.168.4.3</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

| Step 4 | redistribute protocol [process-id] (level-1 | level-1-2 | level-2) [metric metric-value] [metric-type type-value] [match {internal | external [1 | 2] | nssa-external [1 | 2] } ] [tag tag-value] [route-map map-tag | policy policy-tag] |

### Purpose

Redistributes OSPF routes from one routing domain to another routing domain.

or

Redistributes OSPFv3 routes from one routing domain to another routing domain.

- This command causes the router to become an ASBR by definition.
- OSPF tags all routes learned through redistribution as external.
- The protocol and its process ID, if it has one, indicate the protocol being redistributed into OSPF.
- The metric is the cost you assign to the external route. The default is 20 for all protocols except BGP, whose default metric is 1.
- The OSPF example redistributes BGP autonomous system 1, Level 1 routes into OSPF as Type 2 external routes.
- The OSPFv3 example redistributes BGP autonomous system 1, Level 1 and 2 routes into OSPF. The external link type associated with the default route advertised into the OSPFv3 routing domain is the Type 1 external route.

### Example:

```
RP/0/RSP0/CPU0:router(config-ospf)# redistribute bgp 1 level-1
or
RP/0/RSP0/CPU0:router(config-router)# redistribute bgp 1 level-1-2 metric-type 1
```

### Note

RPL is not supported for OSPFv3.
How to Implement OSPF

**Step 5**

```
summary-prefix address mask [not-advertise] [tag tag]
```

or

```
summary-prefix ipv6-prefix/prefix-length [not-advertise] [tag tag]
```

**Example:**

RP/0/RSP0/CPU0:router(config-ospf)# summary-prefix 10.1.0.0 255.255.0.0

or

RP/0/RSP0/CPU0:router(config-router)# summary-prefix 2010:11:22::/32

(Optional) Creates aggregate addresses for OSPF.

or

(Optional) Creates aggregate addresses for OSPFv3.

- This command provides external route summarization of the non-OSPF routes.
- External ranges that are being summarized should be contiguous. Summarization of overlapping ranges from two different routers could cause packets to be sent to the wrong destination.
- This command is optional. If you do not specify it, each route is included in the link-state database and advertised in LSAs.
- In the OSPFv2 example, the summary address 10.1.0.0 includes address 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. Only the address 10.1.0.0 is advertised in an external LSA.
- In the OSPFv3 example, the summary address 2010:11:22::/32 has addresses such as 2010:11:22:0:1000::1, 2010:11:22:0:2000:679:1, and so on. Only the address 2010:11:22::/32 is advertised in the external LSA.

**Step 6**

```
end
```

or

```
commit
```

**Example:**

RP/0/RSP0/CPU0:router(config-ospf)# end

or

RP/0/RSP0/CPU0:router(config-ospf)# commit

Saves configuration changes.

- When you issue the `end` command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
Configuring OSPF Shortest Path First Throttling

This task explains how to configure SPF scheduling in millisecond intervals and potentially delay SPF calculations during times of network instability. This task is optional.

Prerequisites

See the “OSPF Shortest Path First Throttling” section on page RC-179 for information about OSPF SPF throttling.

SUMMARY STEPS

1. configure
2. router ospf process-name
   or
   router ospfv3 process-name
3. router-id {router-id}
4. timers throttle spf spf-start spf-hold spf-max-wait
5. area area-id
6. interface type instance
7. end
   or
   commit
8. show ospf [process-name]
   or
   show ospfv3 [process-name]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong>&lt;br&gt;Enables OSPF routing for the specified routing process and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>router ospf process-name&lt;br&gt;or&lt;br&gt;router ospfv3 process-name&lt;br&gt;Enables OSPFv3 routing for the specified routing process and places the router in router ospfv3 configuration mode.</td>
</tr>
</tbody>
</table>
| Example:          | RP/0/RSP0/CPU0:router(config)# router ospf 1<br>or<br>RP/0/RSP0/CPU0:router(config)# router ospfv3 1
| **Note**          | The process-name argument is any alphanumeric string no longer than 40 characters. |
## Command or Action | Purpose
---|---
### Step 3 | **router-id** *(router-id)*
**Example:**
RP/0/RSP0/CPU0:router(config-ospf)# router-id 192.168.4.3
Configures a router ID for the OSPF process.
**Note** We recommend using a stable IPv4 address as the router ID.

### Step 4 | **timers throttle spf** *spf-start* *spf-hold* *spf-max-wait*
**Example:**
RP/0/RSP0/CPU0:router(config-ospf)# timers throttle spf 10 4800 90000
Sets SPF throttling timers.

### Step 5 | **area** *area-id*
**Example:**
RP/0/RSP0/CPU0:router(config-ospf)# area 0
Enters area configuration mode and configures a backbone area.
- The *area-id* argument can be entered in dotted-decimal or IPv4 address notation, such as area 1000 or area 0.0.3.232. However, you must choose one form or the other for an area. We recommend using the IPv4 address notation.

### Step 6 | **interface** *type* *instance*
**Example:**
RP/0/RSP0/CPU0:router(config-ospf-ar)# interface GigabitEthernet 0/1/0/3
Enters interface configuration mode and associates one or more interfaces to the area.

### Step 7 | **end** or **commit**
**Example:**
RP/0/RSP0/CPU0:router(config-ospf-ar-if)# end
or
RP/0/RSP0/CPU0:router(config-ospf-ar-if)# commit
Saves configuration changes.
- When you issue the **end** command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Implementing OSPF on Cisco ASR 9000 Series Routers

How to Implement OSPF

Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide

Examples

In the following example, the `show ospf` EXEC command is used to verify that the initial SPF schedule delay time, minimum hold time, and maximum wait time are configured correctly. Additional details are displayed about the OSPF process, such as the router type and redistribution of routes.

```
RP/0/RSP0/CPU0:router# show ospf 1
```

```
Routing Process "ospf 1" with ID 192.168.4.3
Supports only single TOS(TOS0) routes
Supports opaque LSA
It is an autonomous system boundary router
Redistributing External Routes from, ospf 2
Initial SPF schedule delay 5 msecs
Minimum hold time between two consecutive SPF s 100 msecs
Maximum wait time between two consecutive SPFs 1000 msecs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 00000000
Number of opaque AS LSA 0. Checksum Sum 00000000
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. 1 normal 0 stub 0 nssa
External flood list length 0
Non-Stop Forwarding enabled
```

Note

For a description of each output display field, see the `show ospf` command in the OSPF Commands on Cisco ASR 9000 Series Routers module in the Cisco ASR 9000 Series Aggregation Services Router Routing Command Reference.

Configuring Nonstop Forwarding Specific to Cisco for OSPF Version 2

This task explains how to configure OSPF NSF specific to Cisco on your NSF-capable router. This task is optional.

Prerequisites

OSPF NSF requires that all neighbor networking devices be NSF aware, which happens automatically after you install the Cisco IOS XR image on the router. If an NSF-capable router discovers that it has non-NSF-aware neighbors on a particular network segment, it disables NSF capabilities for that segment. Other network segments composed entirely of NSF-capable or NSF-aware routers continue to provide NSF capabilities.
Implementing OSPF on Cisco ASR 9000 Series Routers

How to Implement OSPF

See the “Nonstop Forwarding for OSPF Version 2” section on page RC-180 for conceptual information.

Restrictions

The following are restrictions when configuring nonstop forwarding:

- OSPF Cisco NSF for virtual links is not supported.
- Neighbors must be NSF aware.

SUMMARY STEPS

1. configure
2. router ospf process-name
3. router-id {router-id}
4. nsf cisco
   or
   nsf cisco enforce global
5. nsf interval seconds
6. end
   or
   commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong>&lt;br&gt;configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:&lt;br&gt;RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong>&lt;br&gt;router ospf process-name</td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode. &lt;br&gt;Note The process-name argument is any alphanumeric string no longer than 40 characters.</td>
</tr>
<tr>
<td>Example:&lt;br&gt;RP/0/RSP0/CPU0:router(config)# router ospf 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong>&lt;br&gt;router-id {router-id}</td>
<td>Configures a router ID for the OSPF process. &lt;br&gt;Note We recommend using a stable IPv4 address as the router ID.</td>
</tr>
<tr>
<td>Example:&lt;br&gt;RP/0/RSP0/CPU0:router(config-ospf)# router-id 192.168.4.3</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring OSPF Version 2 for MPLS Traffic Engineering

This task explains how to configure OSPF for MPLS TE. This task is optional.

For a description of the MPLS TE tasks and commands that allow you to configure the router to support tunnels, configure an MPLS tunnel that OSPF can use, and troubleshoot MPLS TE, see the *Implementing MPLS Traffic Engineering on Cisco ASR 9000 Series Routers* module of the *Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide*.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 4**
  
  nsf cisco  
  or  
  nsf cisco enforce global  
  
  Example:  
  RP/0/RSP0/CPU0:router(config-ospf)# nsf cisco enforce global  
  | Enables Cisco NSF operations for the OSPF process.  
  
  • Use the *nsf cisco* command without the optional *enforce* and *global* keywords to abort the NSF restart mechanism on the interfaces of detected non-NSF neighbors and allow NSF neighbors to function properly.  
  
  • Use the *nsf cisco* command with the optional *enforce* and *global* keywords if the router is expected to perform NSF during restart. However, if non-NSF neighbors are detected, NSF restart is canceled for the entire OSPF process. |
| **Step 5**
  
  nsf interval seconds  
  
  Example:  
  RP/0/RSP0/CPU0:router(config-ospf)# nsf interval 120  
  | Sets the minimum time between NSF restart attempts.  
  
  **Note**  
  When you use this command, the OSPF process must be up for at least 90 seconds before OSPF attempts to perform an NSF restart. |
| **Step 6**
  
  end  
  or  
  commit  
  
  Example:  
  RP/0/RSP0/CPU0:router(config-ospf)# end  
  or  
  RP/0/RSP0/CPU0:router(config-ospf)# commit  
  | Saves configuration changes.  
  
  • When you issue the *end* command, the system prompts you to commit changes:  
  
  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:  
  
  – Entering *yes* saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.  
  
  – Entering *no* exits the configuration session and returns the router to EXEC mode without committing the configuration changes.  
  
  – Entering *cancel* leaves the router in the current configuration session without exiting or committing the configuration changes.  
  
  • Use the *commit* command to save the configuration changes to the running configuration file and remain within the configuration session. |
Prerequisites

Your network must support the following Cisco IOS XR features before you enable MPLS TE for OSPF on your router:

- MPLS
- IP Cisco Express Forwarding (CEF)

Note

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

SUMMARY STEPS

1. configure
2. router ospf process-name
3. router-id {router-id}
4. mpls traffic-eng router-id {ip-address \ interface-type interface-instance}
5. area area-id
6. mpls traffic-eng
7. interface type instance
8. end or commit
9. show ospf [process-name] [area-id] mpls traffic-eng {link | fragment}

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td>Step 2 router ospf process-name</td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# router ospf 1</td>
</tr>
<tr>
<td>Step 3 router-id {router-id}</td>
<td>Configures a router ID for the OSPF process.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-ospf)# router-id 192.168.4.3</td>
</tr>
</tbody>
</table>
### Step 4

**Command or Action**

```
mpls traffic-eng router-id (ip-address | interface-type interface-instance)
```

**Example:**

```
RP/0/RSP0/CPU0:router(config-ospf)# mpls traffic-eng router-id loopback 0
```

**Purpose**

(Optional) Specifies that the traffic engineering router identifier for the node is the IP address associated with a given interface.

- This IP address is flooded to all nodes in TE LSAs.
- For all traffic engineering tunnels originating at other nodes and ending at this node, you must set the tunnel destination to the traffic engineering router identifier of the destination node because that is the address that the traffic engineering topology database at the tunnel head uses for its path calculation.
- We recommend that loopback interfaces be used for MPLS TE router ID because they are more stable than physical interfaces.

### Step 5

**Command or Action**

```
area area-id
```

**Example:**

```
RP/0/RSP0/CPU0:router(config-ospf)# area 0
```

**Purpose**

Enters area configuration mode and configures an area for the OSPF process.

- The `area-id` argument can be entered in dotted-decimal or IPv4 address notation, such as area 1000 or area 0.0.3.232. However, you must choose one form or the other for an area.

### Step 6

**Command or Action**

```
mpls traffic-eng
```

**Example:**

```
RP/0/RSP0/CPU0:router(config-ospf)# mpls traffic-eng
```

**Purpose**

Configures the MPLS TE under the OSPF area.

### Step 7

**Command or Action**

```
interface type instance
```

**Example:**

```
RP/0/RSP0/CPU0:router(config-ospf-ar)# interface interface loopback0
```

**Purpose**

Enters interface configuration mode and associates one or more interfaces to the area.
### Command or Action | Purpose
--- | ---
**Step 8**  
end  
commit  

**Example:**  
RP/0/RSP0/CPU0:router(config-ospf-ar-if)# end  
RP/0/RSP0/CPU0:router(config-ospf-ar-if)# commit  

**Step 9**  
`show ospf [process-name] [area-id] mpls traffic-eng {link | fragment}`  

**Example:**  
RP/0/RSP0/CPU0:router# show ospf 1 0 mpls traffic-eng link

Saves configuration changes.  
- When you issue the `end` command, the system prompts you to commit changes:  
  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:  
  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.  
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.  
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.  
- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.  

(Optional) Displays information about the links and fragments available on the local router for MPLS TE.
Examples

This section provides the following output examples:

- Sample Output for the show ospf Command Before Configuring MPLS TE, page RC-221
- Sample Output for the show ospf mpls traffic-eng Command, page RC-221
- Sample Output for the show ospf Command After Configuring MPLS TE, page RC-222

Sample Output for the show ospf Command Before Configuring MPLS TE

In the following example, the `show route ospf` EXEC command verifies that GigabitEthernet interface 0/3/0/0 exists and MPLS TE is not configured:

```
RP/0/RSP0/CPU0:router# show route ospf 1
O  11.0.0.0/24 [110/15] via 0.0.0.0, 3d19h, tunnel-te1
O  192.168.0.12/32 [110/11] via 11.1.0.2, 3d19h, GigabitEthernet0/3/0/0
O  192.168.0.13/32 [110/6] via 0.0.0.0, 3d19h, tunnel-te1
RP/0/RSP0/CPU0:router#
```

Sample Output for the show ospf mpls traffic-eng Command

In the following example, the `show ospf mpls traffic-eng` EXEC command verifies that the MPLS TE fragments are configured correctly:

```
RP/0/RSP0/CPU0:router# show ospf 1 mpls traffic-eng fragment
OSPF Router with ID (192.168.4.3) (Process ID 1)
Area 0 has 1 MPLS TE fragment. Area instance is 3.
MPLS router address is 192.168.4.2
Next fragment ID is 1
Fragment 0 has 1 link. Fragment instance is 3.
Fragment has 0 link the same as last update.
Fragment advertise MPLS router address
Link is associated with fragment 0. Link instance is 3
  Link connected to Point-to-Point network
  Link ID :55.55.55.55
  Interface Address :192.168.50.21
  Neighbor Address :192.168.4.1
  Admin Metric :0
  Maximum bandwidth :19440000
  Maximum global pool reservable bandwidth :25000000
  Maximum sub pool reservable bandwidth :31250000
  Number of Priority :8
  Global pool unreserved BW
    Priority 0 : 25000000 Priority 1 : 25000000
    Priority 2 : 25000000 Priority 3 : 25000000
    Priority 4 : 25000000 Priority 5 : 25000000
    Priority 6 : 25000000 Priority 7 : 25000000
  Sub pool unreserved BW
    Priority 0 : 31250000 Priority 1 : 31250000
    Priority 2 : 31250000 Priority 3 : 31250000
    Priority 4 : 31250000 Priority 5 : 31250000
    Priority 6 : 31250000 Priority 7 : 31250000
  Affinity Bit :0
```
In the following example, the `show ospf mpls traffic-eng` EXEC command verifies that the MPLS TE links on area instance 3 are configured correctly:

```
RP/0/RSP0/CPU0:router# show ospf mpls traffic-eng link

OSPF Router with ID (192.168.4.1) (Process ID 1)

Area 0 has 1 MPLS TE links. Area instance is 3.

Links in hash bucket 53.
  Link is associated with fragment 0. Link instance is 3
  Link connected to Point-to-Point network
  Link ID :192.168.20.50
  Interface Address :192.168.4.1
  Admin Metric :0
  Maximum bandwidth :19440000
  Maximum global pool reservable bandwidth :25000000
  Maximum sub pool reservable bandwidth :3125000
  Number of Priority :8
  Global pool unreserved BW
    Priority 0 :  25000000 Priority 1 :  25000000
    Priority 2 :  25000000 Priority 3 :  25000000
    Priority 4 :  25000000 Priority 5 :  25000000
    Priority 6 :  25000000 Priority 7 :  25000000
  Sub pool unreserved BW
    Priority 0 :   3125000 Priority 1 :   3125000
    Priority 2 :   3125000 Priority 3 :   3125000
    Priority 4 :   3125000 Priority 5 :   3125000
    Priority 6 :   3125000 Priority 7 :   3125000
  Affinity Bit :0
```

Sample Output for the `show ospf` Command After Configuring MPLS TE

In the following example, the `show route ospf` EXEC command verifies that the MPLS TE tunnels replaced GigabitEthernet interface 0/3/0/0 and that configuration was performed correctly:

```
RP/0/RSP0/CPU0:router# show route ospf 1

O E2 192.168.10.0/24 [110/20] via 0.0.0.0, 00:00:15, tunnel2
O E2 192.168.11.0/24 [110/20] via 0.0.0.0, 00:00:15, tunnel2
O E2 192.168.12.0/24 [110/20] via 0.0.0.0, 00:00:15, tunnel2
O  192.168.12.0/24 [110/2] via 0.0.0.0, 00:00:15, tunnel2
```

Configuring OSPFv3 Graceful Restart

This task explains how to configure a graceful restart for an OSPFv3 process. This task is optional.

**SUMMARY STEPS**

1. `configure`
2. `router ospfv3 process-name`
3. `graceful-restart`
4. `graceful-restart lifetime`
5. `graceful-restart interval seconds`
6. `graceful-restart helper disable`
### How to Implement OSPF

#### Step 1
```
configure
```
**Purpose:** Enters global configuration mode.

**Example:**
```
RP/0/RSP0/CPU0:router# configure
```

#### Step 2
```
router ospfv3 [process-name [area-id]]
```
**Purpose:** Enters router configuration mode for OSPFv3. The process name is a word that uniquely identifies an OSPF routing process. The process name is any alphanumeric string no longer than 40 characters without spaces.

**Example:**
```
RP/0/RSP0/CPU0:router(config)# router ospfv3 test
```

#### Step 3
```
graceful-restart
```
**Purpose:** Enables graceful restart on the current router.

**Example:**
```
RP/0/RSP0/CPU0:router(config-ospfv3)# graceful-restart
```

#### Step 4
```
graceful-restart lifetime
```
**Purpose:** Specifies a maximum duration for a graceful restart.

- The default lifetime is 95 seconds.
- The range is 90 to 3600 seconds.

**Example:**
```
RP/0/RSP0/CPU0:router(config-ospfv3)# graceful-restart lifetime 120
```

#### Step 5
```
graceful-restart interval [seconds]
```
**Purpose:** Specifies the interval (minimal time) between graceful restarts on the current router.

- The default value for the interval is 90 seconds.
- The range is 90 to 3600 seconds.

**Example:**
```
RP/0/RSP0/CPU0:router(config-ospfv3)# graceful-restart interval 120
```

#### Step 6
```
graceful-restart helper disable
```
**Purpose:** Disables the helper capability.

**Example:**
```
RP/0/RSP0/CPU0:router(config-ospfv3)# graceful-restart helper disable
```

8. **show ospfv3 [process-name [area-id]] database grace**

### Detailed Step

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> router ospfv3 [process-name [area-id]]</td>
<td>Enters router configuration mode for OSPFv3. The process name is a word that uniquely identifies an OSPF routing process. The process name is any alphanumeric string no longer than 40 characters without spaces.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>router ospfv3 [process-name]</td>
<td>Enters router configuration mode for OSPFv3.</td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config)# router ospfv3 test</code></td>
<td>Enters router configuration mode for OSPFv3. The process name is a word that uniquely identifies an OSPF routing process. The process name is any alphanumeric string no longer than 40 characters without spaces.</td>
</tr>
<tr>
<td><strong>Step 3</strong> graceful-restart</td>
<td>Enables graceful restart on the current router.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>graceful-restart</td>
<td>Enables graceful restart on the current router.</td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-ospfv3)# graceful-restart</code></td>
<td>Enables graceful restart on the current router.</td>
</tr>
<tr>
<td><strong>Step 4</strong> graceful-restart lifetime [seconds]</td>
<td>Specifies a maximum duration for a graceful restart.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>graceful-restart lifetime [seconds]</td>
<td>Specifies a maximum duration for a graceful restart.</td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-ospfv3)# graceful-restart lifetime 120</code></td>
<td>Specifies a maximum duration for a graceful restart.</td>
</tr>
<tr>
<td><strong>Step 5</strong> graceful-restart interval [seconds]</td>
<td>Specifies the interval (minimal time) between graceful restarts on the current router.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>graceful-restart interval [seconds]</td>
<td>Specifies the interval (minimal time) between graceful restarts on the current router.</td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-ospfv3)# graceful-restart interval 120</code></td>
<td>Specifies the interval (minimal time) between graceful restarts on the current router.</td>
</tr>
<tr>
<td><strong>Step 6</strong> graceful-restart helper disable</td>
<td>Disables the helper capability.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>graceful-restart helper disable</td>
<td>Disables the helper capability.</td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-ospfv3)# graceful-restart helper disable</code></td>
<td>Disables the helper capability.</td>
</tr>
</tbody>
</table>
How to Implement OSPF

Displaying Information About Graceful Restart

This section describes the tasks you can use to display information about a graceful restart.

To see if the feature is enabled and when the last graceful restart ran, use the `show ospf` command. To see details for an OSPFv3 instance, use the `show ospfv3 process-name [area-id] database grace` command.

Displaying the State of the Graceful Restart Feature

The following screen output shows the state of the graceful restart capability on the local router:

```
RP/0/RSP0/CPU0:router# show ospfv3 1 database grace

Routing Process "ospfv3 1" with ID 2.2.2.2
Initial SPF schedule delay 5000 msecs
Minimum hold time between two consecutive SPFfs 10000 msecs
Maximum wait time between two consecutive SPFfs 10000 msecs
Initial LSA throttle delay 0 msecs
Minimum hold time for LSA throttle 5000 msecs
Maximum wait time for LSA throttle 5000 msecs
Minimum LSA arrival 1000 msecs
LSA group pacing timer 240 secs
Interface flood pacing timer 33 msecs
Retransmission pacing timer 66 msecs
Maximum number of configured interfaces 255
Number of external LSA 0. Checksum Sum 00000000
```
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
Graceful Restart enabled, last GR 11:12:26 ago (took 6 secs)
Area BACKBONE(0)
  Number of interfaces in this area is 1
  SPF algorithm executed 1 times
  Number of LSA 6. Checksum Sum 0x0268a7
  Number of DCbitless LSA 0
  Number of indication LSA 0
  Number of DoNotAge LSA 0
  Flood list length 0

Displaying Graceful Restart Information for an OSPFv3 Instance
The following screen output shows the link state for an OSPFv3 instance:

```
RP/0/RSP0/CPU0:router# show ospfv3 1 database grace

OSPFv3 Router with ID (2.2.2.2) (Process ID 1)

Router Link States (Area 0)
ADV Router | Age | Seq# | Fragment ID | Link count | Bits
-----------|-----|------|-------------|------------|------
1.1.1.1    | 1949| 0x8000000e| 0           | 1          | None
2.2.2.2    | 2007| 0x80000011| 0           | 1          | None

Link (Type-8) Link States (Area 0)
ADV Router | Age | Seq# | Link ID | Interface
-----------|-----|------|---------|----------
1.1.1.1    | 180 | 0x80000006| 1       | P00/2/0/0
2.2.2.2    | 2007| 0x80000006| 1       | P00/2/0/0

Intra Area Prefix Link States (Area 0)
ADV Router | Age | Seq# | Link ID | Ref-lstype | Ref-LSID
-----------|-----|------|---------|------------|------
1.1.1.1    | 180 | 0x80000006| 0       | 0x2001     | 0
2.2.2.2    | 2007| 0x80000006| 0       | 0x2001     | 0

Grace (Type-11) Link States (Area 0)
ADV Router | Age | Seq# | Link ID | Interface
-----------|-----|------|---------|----------
2.2.2.2    | 2007| 0x80000005| 1       | P00/2/0/0

RP/0/RSP0/CPU0:router#
```

Configuring an OSPFv2 Sham Link
This task explains how to configure a provider edge (PE) router to establish an OSPFv2 sham link connection across a VPN backbone. This task is optional.

Prerequisites
Before configuring a sham link in a Multiprotocol Label Switching (MPLS) VPN between provider edge (PE) routers, OSPF must be enabled as follows:

- Create an OSPF routing process.
- Configure a loopback interface that belongs to VRF and assign an IPv4 address with the host mask to it.
- Configure the sham link under the area submode.

See the “Enabling OSPF” section on page RC-188 for information on these OSPF configuration prerequisites.
### Summary Steps

1. `configure`
2. `interface type instance`
3. `vrf vrf-name`
4. `ipv4 address ipv4-address mask`
5. `end`
6. `router ospf instance-id`
7. `vrf vrf-name`
8. `router-id {router-id}`
9. `redistribute bgp process-id`
10. `area area-id`
11. `sham-link source-address destination-address`
12. `cost cost`
13. `end` or `commit`

### Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>interface type instance</code></td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# interface loopback 3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>vrf vrf-name</code></td>
<td>Assigns an interface to the VPN routing and forwarding (VRF) instance.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-if)# vrf vrf1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>ipv4 address ip-address mask</code></td>
<td>Assigns an IP address and subnet mask to the interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.225</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>end</td>
</tr>
<tr>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if)# end</td>
</tr>
<tr>
<td>Purpose</td>
<td>Saves configuration changes. When you issue the <code>end</code> command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td></td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td>• Entering <code>yes</code> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Entering <code>no</code> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Entering <code>cancel</code> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th><code>router ospf</code> instance-id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# router ospf isp</td>
</tr>
<tr>
<td>Purpose</td>
<td>Enables OSPF routing for the specified routing process, and places the router in router configuration mode. In this example, the OSPF instance is called isp.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th><code>vrf</code> vrf-name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-ospf)# vrf vrf1</td>
</tr>
<tr>
<td>Purpose</td>
<td>Creates a VRF instance and enters VRF configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th><code>router-id</code> (router-id)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-ospf-vrf)# router-id 192.168.4.3</td>
</tr>
<tr>
<td>Purpose</td>
<td>Configures a router ID for the OSPF process.</td>
</tr>
<tr>
<td>Note</td>
<td>We recommend using a stable IPv4 address as the router ID.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th><code>redistribute bgp</code> process-id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-ospf-vrf)# redistribute bgp 1</td>
</tr>
<tr>
<td>Purpose</td>
<td>Redistributes OSPF routes from the one routing domain to another routing domain.</td>
</tr>
<tr>
<td></td>
<td>• This command causes the router to become an ASBR by definition.</td>
</tr>
<tr>
<td></td>
<td>• OSPF tags all routes learned through redistribution as external.</td>
</tr>
<tr>
<td></td>
<td>• The protocol and its process ID, if it has one, indicate the protocol being redistributed into OSPF.</td>
</tr>
<tr>
<td></td>
<td>• The BGP MED value is copied to the LSA metric field when BGP VPN routes are redistributed to OSPF.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 10</th>
<th><code>area</code> area-id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-ospf-vrf)# area 0</td>
</tr>
<tr>
<td>Purpose</td>
<td>Enters area configuration mode and configures an area for the OSPF process.</td>
</tr>
<tr>
<td></td>
<td>• The <code>area-id</code> argument can be entered in dotted-decimal or IPv4 address notation, such as area 1000 or area 0.0.3.232. However, you must choose one form or the other for an area.</td>
</tr>
</tbody>
</table>
Enabling Nonstop Routing for OSPFv2

This optional task describes how to enable nonstop routing (NSR) for OSPFv2 processes. NSR is disabled by default. When NSR is enabled, OSPF processes on the active RSP, synchronizes all necessary data, and states with the OSPF process on the standby RSP. When the switchover happens, OSPF processes on the newly active RSP, and has all the necessary data and states to continue running, and does not require any help from its neighbors.

Summary Steps

1. configure
2. router ospf instance-id
3. nsr
4. exit
5. `event manager policy ospf_sysmgr_abort.tcl username user`
   or
   `event manager policy ospf_sysmgr_user.tcl username user`
6. `aaa authorization eventmanager default local`
7. `end`
   or
   `commit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>router ospf instance-id</code></td>
<td>Enables OSPF routing for the specified routing process, and places the router in router configuration mode. In this example, the OSPF instance is called isp.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# router ospf isp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>nsr</code></td>
<td>Enables NSR for the OSPFv2 process.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf)# nsr</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>exit</code></td>
<td>Exits interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>event manager policy ospf_sysmgr_abort.tcl username user</code> or <code>event manager policy ospf_sysmgr_user.tcl username user</code></td>
<td>Registers an Embedded Event Manager (EEM) policy with the EEM.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# event manager policy ospf_sysmgr_abort.tcl username user</td>
<td></td>
</tr>
</tbody>
</table>
How to Implement OSPF

---

## Enabling Multicast-intact for OSPFv2

This optional task describes how to enable multicast-intact for OSPFv2 routes that use IPv4 addresses.

### Summary Steps

1. configure
2. router ospf instance-id
3. mpls traffic-eng multicast-intact
4. end
   or
   commit

### Command or Action | Purpose
--- | ---
Step 6 | aaa authorization eventmanager default local
Example: RP/0/RSP0/CPU0:router(config)# aaa authorization eventmanager default local
| Creates a method list for authorization.

Step 7 | end
or
commit
Example: RP/0/RSP0/CPU0:router(config-ospf)# end
or
RP/0/RSP0/CPU0:router(config-ospf)# commit
| Saves configuration changes.
- When you issue the **end** command, the system prompts you to commit changes:
  Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:
  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> router ospf instance-id</td>
<td>Enables OSPF routing for the specified routing process, and places the router in router configuration mode. In this example, the OSPF instance is called isp.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# router ospf isp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> mpls traffic-eng multicast-intact</td>
<td>Enables multicast-intact.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# mpls traffic-eng multicast-intact</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# end or commit</td>
<td></td>
</tr>
</tbody>
</table>

Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

- When you issue the end command, the system prompts you to commit changes:
  
  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  
  - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  
  - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

### Associating Interfaces to a VRF

This task explains how to associate an interface with a VPN Routing and Forwarding (VRF) instance.

### SUMMARY STEPS

1. configure
2. router ospf process-name
3. vrf vrf-name
4. interface type instance
5. **ipv4 address** *ip-address mask*
6. **ipv6 address** *ipv6-prefix/prefix-length [eui-64]*
7. **ipv4 mtu** *mtu*
8. **end**
   or
   **commit**

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong></td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Enters global configuration mode.</strong></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>router ospf</strong> <em>process-name</em>*</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</strong></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# router ospf 1</td>
<td><strong>Note</strong>  The <em>process-name</em> argument is any alphanumeric string no longer than 40 characters.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>vrf</strong> <em>vrf-name</em>*</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Creates a VRF instance and enters VRF configuration mode.</strong></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# vrf vrf1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>interface</strong> <em>type instance</em>*</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Enters interface configuration mode and associates one or more interfaces to the VRF.</strong></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-vrf)# interface GigabitEthernet 0/0/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>ipv4 address</strong> <em>ip-address mask</em>*</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Assigns an IP address and subnet mask to the interface.</strong></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>ipv6 address</strong> <em>ipv6-prefix/prefix-length [eui-64]</em></td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface.</strong></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-if)# ipv6 address 2001:0DB8:C18:1::64</td>
<td><strong>A slash-mark (/) must precede the <em>prefix-length</em> argument, and there is no space between the <em>ipv6-prefix</em> argument and the slash.</strong></td>
</tr>
</tbody>
</table>
**Configuring OSPF as a Provider Edge to Customer Edge (PE-CE) Protocol**

**SUMMARY STEPS**

1. configure
2. router ospf process-name
3. vrf vrf-name
4. router-id {router-id}
5. redistribute protocol [process-id] {level-1 | level-1-2 | level-2} [metric metric-value] [metric-type type-value] [match {internal | external [1 | 2] | nssa-external [1 | 2]}] [tag tag-value] [route-map map-tag] [route-policy policy-tag]
6. area area-id
7. interface type instance
8. exit
9. domain-id [secondary] type {0005 | 0105 | 0205 | 8005} value value
10. domain-tag tag
11. disable-dn-bit-check

---

**Command or Action** | **Purpose**
--- | ---
Step 7 `ipv4 mtu mtu` | Sets the maximum transmission unit (MTU) size of IPv4 packets sent on the interface.

**Example:**

RP/0/RSP0/CPU0:router(config-if)# ipv4 mtu 300

Step 8 `end` or `commit` | Saves configuration changes.

**Example:**

RP/0/RSP0/CPU0:router(config-ospf-ar-if)# end
or
RP/0/RSP0/CPU0:router(config-ospf-ar-if)# commit

- When you issue the `end` command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
  
  - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  
  - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  
  - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> router ospf process-name</td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Note: The <code>process-name</code> argument is any alphanumeric string no longer than 40 characters.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# router ospf 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> vrf vrf-name</td>
<td>Creates a VRF instance and enters VRF configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# vrf vrf1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> router-id (router-id)</td>
<td>Configures a router ID for the OSPF process.</td>
</tr>
<tr>
<td>Example:</td>
<td>Note: We recommend using a stable IPv4 address as the router ID.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-vrf)# router-id 192.168.4.3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> redistribute protocol [process-id] (level-1</td>
<td>level-1-2</td>
</tr>
<tr>
<td>Example:</td>
<td>• This command causes the router to become an ASBR by definition.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-vrf)# redistribute bgp 1 level-1</td>
<td>• OSPF tags all routes learned through redistribution as external.</td>
</tr>
<tr>
<td><strong>Step 6</strong> area area-id</td>
<td>Enters area configuration mode and configures an area for the OSPF process.</td>
</tr>
<tr>
<td>Example:</td>
<td>• The protocol and its process ID, if it has one, indicate the protocol being redistributed into OSPF.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-vrf)# area 0</td>
<td>• The metric is the cost you assign to the external route. The default is 20 for all protocols except BGP, whose default metric is 1.</td>
</tr>
<tr>
<td></td>
<td>• The example shows the redistribution of BGP autonomous system 1, Level 1 routes into OSPF as Type 2 external routes.</td>
</tr>
</tbody>
</table>
**Implementing OSPF on Cisco ASR 9000 Series Routers**

**How to Implement OSPF**

**RC-235**

**Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td><code>interface type instance</code></td>
<td>Enters interface configuration mode and associates one or more interfaces to the VRF.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-ospf-vrf)# interface GigabitEthernet 0/0/0</code></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><code>exit</code></td>
<td>Exits interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-if)# exit</code></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>`domain-id [secondary] type [0005</td>
<td>0105</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-ospf-vrf)# domain-id type 0105 value 1AF234</code></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><code>domain-tag tag</code></td>
<td>Specifies the OSPF VRF domain tag.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-ospf-vrf)# domain-tag 234</code></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><code>disable-dn-bit-check</code></td>
<td>Specifies that down bits should be ignored.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-ospf-vrf)# disable-dn-bit-check</code></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td><code>end</code> or <code>commit</code></td>
<td>Saves configuration changes.</td>
</tr>
</tbody>
</table>

- **Example:**
  - `RP/0/RSP0/CPU0:router(config-ospf-vrf)# end`
  - `RP/0/RSP0/CPU0:router(config-ospf-vrf)# commit`

**Command or Action Purpose**

- **Step 7**
  - `interface type instance`
    - Enters interface configuration mode and associates one or more interfaces to the VRF.

- **Step 8**
  - `exit`
    - Exits interface configuration mode.

- **Step 9**
  - `domain-id [secondary] type [0005 | 0105 | 0205 | 8005] value value`
    - Specifies the OSPF VRF domain ID.
    - _The value argument is a six-octet hex number._

- **Step 10**
  - `domain-tag tag`
    - Specifies the OSPF VRF domain tag.
    - _The valid range for tag is 0 to 4294967295._

- **Step 11**
  - `disable-dn-bit-check`
    - Specifies that down bits should be ignored.

- **Step 12**
  - `end` or `commit`
    - Saves configuration changes.
    - _When you issue the end command, the system prompts you to commit changes:_
      
      ```plaintext
      Uncommitted changes found, commit them before exiting(yes/no/cancel)?
      [cancel]:
      
      - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
      - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
      - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.
    
    - Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.
Creating Multiple OSPF Instances (OSPF Process and a VRF)

This task explains how to create multiple OSPF instances. In this case, the instances are a normal OSPF instance and a VRF instance.

SUMMARY STEPS

1. configure
2. router ospf process-name
3. area area-id
4. interface type instance
5. exit
6. vrf vrf-name
7. area area-id
8. interface type instance
9. end
   or
   commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> router ospf process-name</td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# router ospf 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> area area-id</td>
<td>Enters area configuration mode and configures a backbone area.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf)# area 0</td>
<td><em>The area-id argument can be entered in dotted-decimal or IPv4 address notation, such as area 1000 or area 0.0.3.232. However, you must choose one form or the other for an area. We recommend using the IPv4 address notation.</em></td>
</tr>
<tr>
<td><strong>Step 4</strong> interface type instance</td>
<td>Enters interface configuration mode and associates one or more interfaces to the area.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf-ar)# interface GigabitEthernet 0/1/0/3</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring Multi-area Adjacency

This task explains how to create multiple areas on an OSPF primary interface.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong> exit</td>
<td>Enters OSPF configuration mode.</td>
</tr>
<tr>
<td><strong>Step 6</strong> vrf vrf-name</td>
<td>Creates a VRF instance and enters VRF configuration mode.</td>
</tr>
<tr>
<td><strong>Step 7</strong> area area-id</td>
<td>Enters area configuration mode and configures an area for a VRF instance under the OSPF process.</td>
</tr>
<tr>
<td><strong>Step 8</strong> interface type instance</td>
<td>Enters interface configuration mode and associates one or more interfaces to the VRF.</td>
</tr>
<tr>
<td><strong>Step 9</strong> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
</tbody>
</table>

- When you issue the `end` command, the system prompts you to commit changes:
  
  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
Restrictions

Multiple areas are created only on native point-to-point interfaces, such as Packet-over-SONET (PoS) or serial.

SUMMARY STEPS

1. configure
2. router ospf process-name
3. area area-id
4. interface type instance
5. area area-id
6. multi-area-interface type instance
7. end
   or
   commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</td>
</tr>
<tr>
<td>router ospf process-name</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# router ospf 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enters area configuration mode and configures a backbone area.</td>
</tr>
<tr>
<td>area area-id</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# area 0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Enters interface configuration mode and associates one or more interfaces to the area.</td>
</tr>
<tr>
<td>interface type instance</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar)# interface Serial 0/1/0/3</td>
<td></td>
</tr>
</tbody>
</table>

Note: The process-name argument is any alphanumeric string no longer than 40 characters.

- The area-id argument can be entered in dotted-decimal or IPv4 address notation, such as area 1000 or area 0.0.3.232. However, you must choose one form or the other for an area. We recommend using the IPv4 address notation.
Configuring Label Distribution Protocol IGP Auto-configuration for OSPF

This task explains how to configure LDP auto-configuration for an OSPF instance.

Optionally, you can configure this feature for an area of an OSPF instance.

**SUMMARY STEPS**

1. configure
2. router ospf process-name
3. mpls ldp auto-config
4. end
   or
   commit
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 router ospf process-name</td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# router ospf 1</td>
<td></td>
</tr>
<tr>
<td>Step 3 mpls ldp auto-config</td>
<td>Enables LDP IGP interface auto-configuration for an OSPF instance.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# mpls ldp auto-config</td>
<td></td>
</tr>
<tr>
<td>Step 4 end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# end or commit</td>
<td></td>
</tr>
</tbody>
</table>

Configuring Authentication Message Digest Management for OSPF

This task explains how to manage authentication of a keychain on the OSPF interface.

Prerequisites

A valid keychain must be configured before this task can be attempted.

To learn how to configure a keychain and its associated attributes, see the Implementing Key Chain Management on Cisco ASR 9000 Series Routers module of the Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide.
SUMMARY STEPS

1. configure
2. router ospf process-name
3. router-id router-id
4. area area-id
5. interface type instance
6. authentication message-digest keychain keychain
7. end
   or
   commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 router ospf process-name</td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Note: The process-name argument is any alphanumeric string no longer than 40 characters.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# router ospf 1</td>
<td></td>
</tr>
<tr>
<td>Step 3 router-id router-id</td>
<td>Configures a router ID for the OSPF process.</td>
</tr>
<tr>
<td>Example:</td>
<td>Note: We recommend using a stable IPv4 address as the router ID.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# router id</td>
<td></td>
</tr>
<tr>
<td>192.168.4.3</td>
<td></td>
</tr>
<tr>
<td>Step 4 area area-id</td>
<td>Enters area configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>The area-id argument can be entered in dotted-decimal or IPv4 address notation, such as area 1000 or area 0.0.3.232. However, you must choose one form or the other for an area. We recommend using the IPv4 address notation.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf)# area</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Step 5 interface type instance</td>
<td>Enters interface configuration mode and associates one or more interfaces to the area.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar)# interface</td>
<td></td>
</tr>
<tr>
<td>GigabitEthernet0/4/0/1</td>
<td></td>
</tr>
</tbody>
</table>
Examples

The following example shows how to configure the keychain ospf_intf_1 that contains five key IDs. Each key ID is configured with different send-lifetime values; however, all key IDs specify the same text string for the key.

```
key chain ospf_intf_1
key 1
send-lifetime 11:30:30 May 1 2007 duration 600
cryptographic-algorithm MD5T
key-string clear ospf_intf_1
key 2
send-lifetime 11:40:30 May 1 2007 duration 600
cryptographic-algorithm MD5
key-string clear ospf_intf_1
key 3
send-lifetime 11:50:30 May 1 2007 duration 600
cryptographic-algorithm MD5
key-string clear ospf_intf_1
key 4
send-lifetime 12:00:30 May 1 2007 duration 600
cryptographic-algorithm MD5
key-string clear ospf_intf_1
key 5
send-lifetime 12:10:30 May 1 2007 duration 600
cryptographic-algorithm MD5
key-string clear ospf_intf_1
```
The following example shows that keychain authentication is enabled on the Gigabit Ethernet 0/4/0/1 interface:

```
RP/0/RSP0/CPU0:router# show ospf 1 interface GigabitEthernet0/4/0/1
```

GigabitEthernet0/4/0/1 is up, line protocol is up
Internet Address 100.10.10.2/24, Area 0
Process ID 1, Router ID 2.2.2.1, Network Type BROADCAST, Cost: 1
Transmit Delay is 1 sec, State DR, Priority 1
Designated Router (ID) 2.2.2.1, Interface address 100.10.10.2
Backup Designated router (ID) 1.1.1.1, Interface address 100.10.10.1
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:02
Index 3/3, flood queue length 0
Next 0(0)/0(0)
Last flood scan length is 2, maximum is 16
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 1.1.1.1 (Backup Designated Router)
Suppress hello for 0 neighbor(s)
Keychain-based authentication enabled
Key id used is 3
Multi-area interface Count is 0

The following example shows output for configured keys that are active:

```
RP/0/RSP0/CPU0:router# show key chain ospf_intf_1
Key-chain: ospf intf_1/ -
Key 1 -- text "0700325C4836100B0314345D"
cryptographic-algorithm -- MD5
Send lifetime: 11:30:30, 01 May 2007 - (Duration) 600
Accept lifetime: Not configured
Key 2 -- text "10411A0903281B051802157A"
cryptographic-algorithm -- MD5
Send lifetime: 11:40:30, 01 May 2007 - (Duration) 600
Accept lifetime: Not configured
Key 3 -- text "06091C314A71001711112D5A"
cryptographic-algorithm -- MD5
Send lifetime: 11:50:30, 01 May 2007 - (Duration) 600 [Valid now]
Accept lifetime: Not configured
Key 4 -- text "151D181C0215222A3C350A73"
cryptographic-algorithm -- MD5
Send lifetime: 12:00:30, 01 May 2007 - (Duration) 600
Accept lifetime: Not configured
Key 5 -- text "151D181C0215222A3C350A73"
cryptographic-algorithm -- MD5
Send lifetime: 12:10:30, 01 May 2007 - (Duration) 600
Accept lifetime: Not configured
```

RP/0/RSP0/CPU0:router#
Configuring Generalized TTL Security Mechanism (GTSM) for OSPF

This task explains how to set the security time-to-live mechanism on an interface for GTSM.

SUMMARY STEPS

1. **configure**
2. **router ospf process-name**
3. **router-id {router-id}**
4. **log adjacency changes [detail | disable]**
5. **nsf {cisco [enforce global] | ietf [helper disable]}**
6. **timers throttle spf spf-start spf-hold spf-max-wait**
7. **area area-id**
8. **interface type instance**
9. **security ttl [disable | hops hop-count]**
10. **end**
    or
    **commit**
11. **show ospf [process-name] [vrf vrf-name] [area-id] interface [type instance]**

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> router ospf process-name</td>
<td>Enables OSPF routing for the specified routing process and places the router in router configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# router ospf 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router-id (router-id)</td>
<td>Configures a router ID for the OSPF process.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf)# router id 10.10.10.100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> log adjacency changes [detail</td>
<td>disable]</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf-ar-if)# log adjacency changes detail</td>
<td></td>
</tr>
</tbody>
</table>

- By default, this feature is enabled.
- The messages generated by neighbor changes are considered notifications, which are categorized as severity Level 5 in the **logging console** command. The **logging console** command controls which severity level of messages are sent to the console. By default, all severity level messages are sent.
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>`nsf {cisco [enforce global]</td>
<td>ietf [helper disable]}</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-ospf)# nsf ietf</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><code>timers throttle spf spf-start spf-hold spf-max-wait</code></td>
<td>(Optional) Sets SPF throttling timers.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-ospf)# timers throttle spf 500 500 10000</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><code>area area-id</code></td>
<td>Enters area configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-ospf)# area 1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><code>interface type instance</code></td>
<td>Enters interface configuration mode and associates one or more interfaces to the area.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar)# interface GigabitEthernet0/5/0/0</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>`security ttl [disable</td>
<td>hops hop-count]`</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-ospf-ar-if)# security ttl hopes 2</td>
<td></td>
</tr>
</tbody>
</table>
How to Implement OSPF

### Command or Action

**Step 10**

- `end` or `commit`  

**Example:**

```
RP/0/RSP0/CPU0:router(config-ospf-ar-if)# end
```

```
RP/0/RSP0/CPU0:router(config-ospf-ar-if)# commit
```

### Purpose

Saves configuration changes.

- When you issue the **end** command, the system prompts you to commit changes:
  
  Uncommitted changes found, commit them before exiting(yes/no/cancel)?

  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

### Examples

The following is sample output that displays the GTSM security TTL value configured on an OSPF interface:

```
RP/0/RSP0/CPU0:router# show ospf 1 interface GigabitEthernet0/5/0/0
```

```
GigabitEthernet0/5/0/0 is up, line protocol is up
Internet Address 120.10.10.1/24, Area 0
Process ID 1, Router ID 100.100.100.100, Network Type BROADCAST, Cost: 1
Transmit Delay is 1 sec, State BDR, Priority 1
TTL security enabled, hop count 2
Designated Router (ID) 102.102.102.102, Interface address 120.10.10.3
Backup Designated router (ID) 100.100.100.100, Interface address 120.10.10.1
Flush timer for old DR LSA due in 00:02:36
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
   Hello due in 00:00:05
Index 1/1, flood queue length 0
Next 0/0/0/0
Last flood scan length is 1, maximum is 4
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
   Adjacent with neighbor 102.102.102.102 (Designated Router)
Suppress hello for 0 neighbor(s)
   Multi-area interface Count is 0
RP/0/RSP0/CPU0:router#
```
Verifying OSPF Configuration and Operation

This task explains how to verify the configuration and operation of OSPF.

SUMMARY STEPS

1. `show {ospf | ospfv3} [process-name]`
2. `show {ospf | ospfv3} [process-name] border-routers [router-id]`
3. `show {ospf | ospfv3} [process-name] database`
4. `show {ospf | ospfv3} [process-name] [area-id] flood-list interface type instance`
5. `show {ospf | ospfv3} [process-name] [vrf vrf-name] [area-id] interface [type instance]`
6. `show {ospf | ospfv3} [process-name] [area-id] neighbor [interface-type interface-instance] [neighbor-id] [detail]`
7. `clear {ospf | ospfv3} [process-name] process`
8. `clear {ospf | ospfv3} [process-name] statistics [neighbor [interface-type interface-instance] [ip-address]]`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>Step 2</strong></td>
</tr>
<tr>
<td>`show {ospf</td>
<td>ospfv3} [process-name]`</td>
</tr>
<tr>
<td>Example:</td>
<td>Example:</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# show ospf group1</td>
<td>RP/0/RSP0/CPU0:router# show ospf group1 border-routers</td>
</tr>
</tbody>
</table>

(Optional) Displays general information about OSPF routing processes.

(Optional) Displays the internal OSPF routing table entries to an ABR and ASBR.

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

- The various forms of this command deliver information about different OSPF LSAs.

(Optional) Displays a list of OSPF LSAs waiting to be flooded over an interface.
Configuration Examples for Implementing OSPF

This section provides the following configuration examples:

- **Cisco IOS XR Software for OSPF Version 2 Configuration: Example**, page RC-248
- **CLI Inheritance and Precedence for OSPF Version 2: Example**, page RC-250
- **MPLS TE for OSPF Version 2: Example**, page RC-251
- **ABR with Summarization for OSPFV3: Example**, page RC-251
- **ABR Stub Area for OSPFV3: Example**, page RC-251
- **ABR Totally Stub Area for OSPFV3: Example**, page RC-251
- **Route Redistribution for OSPFV3: Example**, page RC-252
- **Virtual Link Configured Through Area 1 for OSPFV3: Example**, page RC-252
- **Virtual Link Configured with MD5 Authentication for OSPF Version 2: Example**, page RC-253
- **VPN Backbone and Sham Link Configured for OSPF Version 2: Example**, page RC-253

**Cisco IOS XR Software for OSPF Version 2 Configuration: Example**

The following example shows how an OSPF interface is configured for an area in Cisco IOS XR software.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong> show {ospf</td>
<td>ospfv3} [process-name] [vrf vrf-name] [area-id] interface [type instance]</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# show ospf 100 interface GigabitEthernet 0/3/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> show {ospf</td>
<td>ospfv3} [process-name] [area-id] neighbor [interface-type interface-instance] [neighbor-id] [detail]</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# show ospf 100 neighbor</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> clear {ospf</td>
<td>ospfv3} [process-name] process</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# clear ospf 100 process</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> clear {ospf</td>
<td>ospfv3} [process-name] statistics [neighbor [interface-type interface-instance] [ip-address]]</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# clear ospf 100 statistics</td>
<td></td>
</tr>
</tbody>
</table>
In Cisco IOS XR software, area 0 must be explicitly configured with the `area` command and all interfaces that are in the range from 10.1.2.0 to 10.1.2.255 are bound to area 0. Interfaces are configured with the `interface` command (while the router is in area configuration mode) and the `area` keyword is not included in the interface statement.

**Cisco IOS XR Software Configuration**

```
interface GigabitEthernet 0/3/0/0
  ip address 10.1.2.1 255.255.255.255
  negotiation auto
!
router ospf 1
  router-id 10.2.3.4
  area 0
    interface GigabitEthernet 0/3/0/0
      !

The following example shows how OSPF interface parameters are configured for an area in Cisco IOS XR software.

In Cisco IOS XR software, OSPF interface-specific parameters are configured in interface configuration mode and explicitly defined for area 0. In addition, the `ip ospf` keywords are no longer required.

**Cisco IOS XR Software Configuration**

```
interface GigabitEthernet 0/3/0/0
  ip address 10.1.2.1 255.255.255.0
  negotiation auto
!
router ospf 1
  router-id 10.2.3.4
  area 0
    interface GigabitEthernet 0/3/0/0
      cost 77
      mtu-ignore
      authentication message-digest
      message-digest-key 1 md5 0 test
      !

The following example shows the hierarchical CLI structure of Cisco IOS XR software.

In Cisco IOS XR software, OSPF areas must be explicitly configured, and interfaces configured under the area configuration mode are explicitly bound to that area. In this example, interface 10.1.2.0/24 is bound to area 0 and interface 10.1.3.0/24 is bound to area 1.

**Cisco IOS XR Software Configuration**

```
interface GigabitEthernet 0/3/0/0
  ip address 10.1.2.1 255.255.255.0
  negotiation auto
!
  interface GigabitEthernet 0/3/0/1
    ip address 10.1.3.1 255.255.255.0
    negotiation auto
!
router ospf 1
  router-id 10.2.3.4
  area 0
    interface GigabitEthernet 0/3/0/0
      !
    area 1
```
CLI Inheritance and Precedence for OSPF Version 2: Example

The following example configures the cost parameter at different hierarchical levels of the OSPF topology, and illustrates how the parameter is inherited and how only one setting takes precedence. According to the precedence rule, the most explicit configuration is used.

The cost parameter is set to 5 in router configuration mode for the OSPF process. Area 1 sets the cost to 15 and area 6 sets the cost to 30. All interfaces in area 0 inherit a cost of 5 from the OSPF process because the cost was not set in area 0 or its interfaces.

In area 1, every interface has a cost of 15 because the cost is set in area 1 and 15 overrides the value 5 that was set in router configuration mode.

Area 4 does not set the cost, but GigabitEthernet interface 0/1/0/2 sets the cost to 20. The remaining interfaces in area 4 have a cost of 5 that is inherited from the OSPF process.

Area 6 sets the cost to 30, which is inherited by GigabitEthernet interfaces 0/1/0/3 and 0/2/0/3. GigabitEthernet interface 0/3/0/3 uses the cost of 1, which is set in interface configuration mode.

```
router ospf 1
  router-id 10.5.4.3
  cost 5
  area 0
    interface GigabitEthernet 0/1/0/0
      !
    interface GigabitEthernet 0/2/0/0
      !
    interface GigabitEthernet 0/3/0/0
      !
  area 1
    cost 15
    interface GigabitEthernet 0/1/0/1
      !
    interface GigabitEthernet 0/2/0/1
      !
    interface GigabitEthernet 0/3/0/1
      !
  area 4
    interface GigabitEthernet 0/1/0/2
      cost 20
      !
    interface GigabitEthernet 0/2/0/2
      !
    interface GigabitEthernet 0/3/0/2
      !
  area 6
    cost 30
    interface GigabitEthernet 0/1/0/3
      !
    interface GigabitEthernet 0/2/0/3
      !
    interface GigabitEthernet 0/3/0/3
      cost 1
      !
```
MPLS TE for OSPF Version 2: Example

The following example shows how to configure the OSPF portion of MPLS TE. However, you still need to build an MPLS TE topology and create an MPLS TE tunnel. See the *Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide* for information.

In this example, loopback interface 0 is associated with area 0 and MPLS TE is configured within area 0.

```plaintext
interface Loopback 0
  address 10.10.10.10 255.255.255.0
!
interface GigabitEthernet 0/2/0/0
  address 10.1.2.2 255.255.255.0
!
router ospf 1
  router-id 10.10.10.10
  nsf
  auto-cost reference-bandwidth 10000
  mpls traffic-eng router-id Loopback 0
  area 0
  mpls traffic-eng
  interface GigabitEthernet 0/2/0/0
  interface Loopback 0
```

ABR with Summarization for OSPFv3: Example

The following example shows the prefix range 2300::/16 summarized from area 1 into the backbone:

```plaintext
router ospfv3 1
  router-id 192.168.0.217
  area 0
  interface GigabitEthernet 0/2/0/1
  area 1
  range 2300::/16
  interface GigabitEthernet 0/2/0/0
```

ABR Stub Area for OSPFv3: Example

The following example shows that area 1 is configured as a stub area:

```plaintext
router ospfv3 1
  router-id 10.0.0.217
  area 0
  interface GigabitEthernet 0/2/0/1
  area 1
  stub
  interface GigabitEthernet 0/2/0/0
```

ABR Totally Stub Area for OSPFv3: Example

The following example shows that area 1 is configured as a totally stub area:

```plaintext
router ospfv3 1
  router-id 10.0.0.217
  area 0
```
interface GigabitEthernet 0/2/0/1
area 1
stub no-summary
interface GigabitEthernet 0/2/0/0

Route Redistribution for OSPFv3: Example

The following example uses prefix lists to limit the routes redistributed from other protocols. Only routes with 9898:1000 in the upper 32 bits and with prefix lengths from 32 to 64 are redistributed from BGP 42. Only routes not matching this pattern are redistributed from BGP 1956.

ipv6 prefix-list list1
seq 10 permit 9898:1000::/32 ge 32 le 64
ipv6 prefix-list list2
seq 10 deny 9898:1000::/32 ge 32 le 64
seq 20 permit ::/0 le 128
cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide

router ospfv3 1
gerouter-id 10.0.0.217
redistribute bgp 42
redistribute bgp 1956
distribute-list prefix-list list1 out bgp 42
distribute-list prefix-list list2 out bgp 1956
area 1
interface GigabitEthernet 0/2/0/0

Virtual Link Configured Through Area 1 for OSPFv3: Example

This example shows how to set up a virtual link to connect the backbone through area 1 for the OSPFv3 topology that consists of areas 0 and 1 and virtual links 10.0.0.217 and 10.0.0.212:

ABR 1 Configuration
router ospfv3 1
router-id 10.0.0.217
area 0
interface GigabitEthernet 0/2/0/1
area 1
virtual-link 10.0.0.212
interface GigabitEthernet 0/2/0/0

ABR 2 Configuration
router ospfv3 1
router-id 10.0.0.212
area 0
interface GigabitEthernet 0/3/0/1
area 1
virtual-link 10.0.0.217
interface GigabitEthernet 0/2/0/0
Virtual Link Configured with MD5 Authentication for OSPF Version 2: Example

The following examples show how to configure a virtual link to your backbone and apply MD5 authentication. You must perform the steps described on both ABRs at each end of the virtual link.

After you explicitly configure the ABRs, the configuration is inherited by all interfaces bound to that area—unless you override the values and configure them explicitly for the interface.

To understand virtual links, see the “Virtual Link and Transit Area for OSPF” section on page RC-176.

In this example, all interfaces on router ABR1 use MD5 authentication:

```
routerr ospf ABR1
  router-id 10.10.10.10
  authentication message-digest
  message-digest-key 100 md5 0 cisco
  area 0
  interface GigabitEthernet 0/2/0/1
  interface GigabitEthernet 0/3/0/0
  area 1
  interface GigabitEthernet 0/3/0/1
  virtual-link 10.10.5.5
```

In this example, only area 1 interfaces on router ABR3 use MD5 authentication:

```
routerr ospf ABR2
  router-id 10.10.5.5
  area 0
  area 1
  authentication message-digest
  message-digest-key 100 md5 0 cisco
  interface GigabitEthernet 0/9/0/1
  virtual-link 10.10.10.10
  area 3
  interface Loopback 0
  interface GigabitEthernet 0/9/0/0
```

VPN Backbone and Sham Link Configured for OSPF Version 2: Example

The following examples show how to configure a provider edge (PE) router to establish a VPN backbone and sham link connection:

```
logging console debugging
vrf vrf_1
  address-family ipv4 unicast
  import route-target
  100:1
  export route-target
  100:1
  !
  !

interface Loopback0
  ipv4 address 2.2.2.1 255.255.255.255
  !
interface Loopback1
  vrf vrf_1
  ipv4 address 10.0.1.3 255.255.255.255
```
! interface GigabitEthernet0/2/0/2
  vrf vrf_1
  ipv4 address 100.10.10.2 255.255.255.0
! interface GigabitEthernet0/2/0/3
  ipv4 address 100.20.10.2 255.255.255.0
!
!
route-policy pass-all
  pass
end-policy
!
router ospf 1
  log adjacency changes
  router-id 2.2.2.2
  vrf vrf_1
    router-id 22.22.22.2
    domain-id type 0005 value 111122223333
    domain-tag 140
    nsf ietf
    redistribute bgp 10
    area 0
      sham-link 10.0.1.3 10.0.0.101
      interface GigabitEthernet0/2/0/2
!
!
!
router ospf 2
  router-id 2.2.2.22
  area 0
    interface Loopback0
!
  interface GigabitEthernet0/2/0/3
!
!
router bgp 10
  bgp router-id 2.2.2.1
  bgp graceful-restart restart-time 300
  bgp graceful-restart
  address-family ipv4 unicast
    redistribute connected
 !
  address-family vpnv4 unicast
  !
  neighbor 2.2.2.2
    remote-as 10
    update-source Loopback0
    address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  vrf vrf_1
    rd 100:1
      address-family ipv4 unicast
        redistribute connected route-policy pass-all
        redistribute ospf 1 match internal external
      !
    !

Where to Go Next

To configure route maps through the RPL for OSPF Version 2, see the Implementing Routing Policy on Cisco ASR 9000 Series Routers module in the Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide.

To build an MPLS TE topology, create tunnels, and configure forwarding over the tunnel for OSPF Version 2; see the Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide.

Additional References

The following sections provide references related to implementing OSPF on Cisco IOS XR software.

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSPF and OSPFv3 commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples</td>
<td>Cisco ASR 9000 Series Aggregation Services Router Routing Command Reference</td>
</tr>
<tr>
<td>MPLS TE feature information</td>
<td>Implementing MPLS Traffic Engineering on Cisco ASR 9000 Series Routers module in Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide</td>
</tr>
</tbody>
</table>

### Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>draft-ietf-ospf-multi-area-adj-07.txt</td>
<td>OSPF Multi-Area Adjacency</td>
</tr>
<tr>
<td>draft-ietf-pce-discoproto-ospf-08.txt</td>
<td>OSPF Protocol Extensions for Path Computation Element (PCE)</td>
</tr>
<tr>
<td>draft-ietf-mpls-igp-sync-00.txt</td>
<td>LDP IGP Synchronization</td>
</tr>
<tr>
<td>draft-ietf-ospf-ospfv3-graceful-restart-07.txt</td>
<td>OSPFv3 Graceful Restart</td>
</tr>
</tbody>
</table>
MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1587</td>
<td>The OSPF NSSA Option</td>
</tr>
<tr>
<td>RFC 1793</td>
<td>Extending OSPF to Support Demand Circuits</td>
</tr>
<tr>
<td>RFC 2328</td>
<td>OSPF Version 2</td>
</tr>
<tr>
<td>RFC 2370</td>
<td>The OSPF Opaque LSA Option</td>
</tr>
<tr>
<td>RFC 2740</td>
<td>OSPF for IPv6</td>
</tr>
<tr>
<td>RFC 3101</td>
<td>The OSPF Not-So-Stubby Area (NSSA) Option</td>
</tr>
<tr>
<td>RFC 3137</td>
<td>OSPF Stub Router Advertisement</td>
</tr>
<tr>
<td>RFC 3509</td>
<td>Alternative Implementations of OSPF Area Border Routers</td>
</tr>
<tr>
<td>RFC 3623</td>
<td>Graceful OSPF Restart</td>
</tr>
<tr>
<td>RFC 3630</td>
<td>Traffic Engineering (TE) Extensions to OSPF Version 2</td>
</tr>
<tr>
<td>RFC 3682</td>
<td>The Generalized TTL Security Mechanism (GTSM)</td>
</tr>
<tr>
<td>RFC 3906</td>
<td>Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels</td>
</tr>
<tr>
<td>RFC 4136</td>
<td>OSPF Refresh and Flooding Reduction in Stable Topologies</td>
</tr>
<tr>
<td>RFC 4206</td>
<td>Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)</td>
</tr>
<tr>
<td>RFC 4124</td>
<td>Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering</td>
</tr>
<tr>
<td>RFC 4576</td>
<td>Using a Link State Advertisement (LSA) Options Bit to Prevent Looping in BGP/MPLS IP Virtual Private Networks (VPNs) ownbit Extension for L3VPN</td>
</tr>
<tr>
<td>RFC 4577</td>
<td>OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)</td>
</tr>
<tr>
<td>RFC 4750</td>
<td>OSPF Version 2 Management Information Base</td>
</tr>
<tr>
<td>RFC 4811</td>
<td>OSPF Out-of-Band Link State Database (LSDB) Resynchronization</td>
</tr>
<tr>
<td>RFC 4812</td>
<td>OSPF Restart Signaling</td>
</tr>
<tr>
<td>RFC 4813</td>
<td>OSPF Link-Local Signaling</td>
</tr>
<tr>
<td>RFC 4970</td>
<td>Extensions to OSPF for Advertising Optional Router Capabilities</td>
</tr>
</tbody>
</table>
## Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>