OSPFv3

This chapter describes how to configure Open Shortest Path First version 3 (OSPFv3) for IPv6 networks on the Cisco NX-OS device.

This chapter includes these sections:

- OSPFv3, on page 1
- Multi-Area adjacency, on page 7
- OSPFv3 and the IPv6 unicast RIB, on page 7
- Address family support, on page 8
- Authentication and encryption , on page 8
- Advanced features, on page 8
- Prerequisites for OSPFv3, on page 13
- Guidelines and limitations for OSPFv3, on page 13
- Default settings, on page 15
- Configure basic OSPFv3, on page 16
- Configure advanced OSPFv3, on page 21
- Configure Encryption and Authentication, on page 42
- Verify the OSPFv3 configuration, on page 54
- Monitor OSPFv3, on page 55
- Configuration examples for OSPFv3, on page 55
- Related topics, on page 56
- Additional references, on page 56

OSPFv3

OSPFv3 is a link-state routing protocol defined by the IETF, designed for IPv6 networks. Each OSPFv3 router periodically sends "hello" packets on its OSPF-enabled interfaces to discover other OSPFv3 routers. When a neighbor is detected, the routers compare information in the hello packets to verify configuration compatibility. If they are compatible, the routers form an adjacency, which means they synchronize their link-state databases to ensure identical OSPFv3 routing information. For more information, see Overview.

Adjacent routers share link-state advertisements (LSAs) that include information about the operational state of each link, the cost of the link, and any other neighbor information. The routers then flood these received LSAs out every OSPF-enabled interface so that all OSPFv3 routers eventually have identical link-state databases. When all OSPFv3 routers have identical link-state databases, the network is converged. Each router

then uses Dijkstra's Shortest Path First (SPF) algorithm to build its route table. To know more about convergence, see Convergence.

You can divide OSPFv3 networks into areas. Routers send most LSAs only within one area, which reduces the CPU and memory requirements for an OSPF-enabled router.

OSPFv3 supports IPv6. For information about OSPF for IPv4, see OSPFv2.

Comparison of OSPFv3 and OSPFv2

Much of the OSPFv3 protocol is the same as in OSPFv2. OSPFv3 is described in RFC 2740.

These are the key differences between the OSPFv3 and OSPFv2 protocols:

- OSPFv3 expands on OSPFv2 to provide support for IPv6 routing prefixes and the larger size of IPv6 addresses.
- LSAs in OSPFv3 are expressed as prefix and prefix length instead of address and mask.
- The router ID and area ID are 32-bit numbers with no relationship to IPv6 addresses.
- OSPFv3 uses link-local IPv6 addresses for neighbor discovery and other features.
- OSPFv3 can use the IPv6 authentication trailer (RFC 6506) or IPSec (RFC 4552) for authentication. However, Cisco NX-OS does not support RFC 6506.
- OSPFv3 redefines LSA types.

Hello packets

OSPFv3 routers periodically send Hello packets on every OSPF-enabled interface. The hello interval determines how frequently the router sends these Hello packets and is configured per interface. OSPFv3 uses Hello packets for these tasks:

- Neighbor discovery
- Keepalives
- Bidirectional communications
- Designated router election. For more information, see Designated Routers.

The Hello packet contains information about the originating OSPFv3 interface and router, including the assigned OSPFv3 cost of the link, the hello interval, and optional capabilities of the originating router. An OSPFv3 interface that receives these Hello packets determines if the settings are compatible with the receiving interface settings. Compatible interfaces are considered neighbors and are added to the neighbor table. See Neighbors for more information.

Hello packets also include a list of router IDs for the routers that the originating interface has communicated with. If the receiving interface sees its own router ID in this list, then bidirectional communication has been established between the two interfaces.

OSPFv3 uses Hello packets as a keepalive message to determine if a neighbor is still communicating. If a router does not receive a Hello packet by the configured dead interval (usually a multiple of the hello interval), then the neighbor is removed from the local neighbor table.

Neighbors

When an OSPFv3 interface has compatible configuration with a remote interface, both the interfaces are considered neighbors. The two OSPFv3 interfaces must match these criteria:

- Hello interval
- Dead interval
- Area ID. See the Areas.
- Optional capabilities

If there is a match, these information is entered into the neighbor table:

- Neighbor ID: The router ID of the neighbor router.
- Priority: Priority of the neighbor router. The priority is used for designated router election. See the Designated Routers.
- State: Indication of whether the neighbor has just been heard from, is in the process of setting up bidirectional communications, is sharing the link-state information, or has achieved full adjacency.
- Dead time: Indication of how long since the last Hello packet was received from this neighbor.
- Link-local IPv6 Address: The link-local IPv6 address of the neighbor.
- Designated Router: Indication of whether the neighbor has been declared the designated router or backup designated router. See the Designated Routers.
- Local interface: The local interface that received the Hello packet for this neighbor.

When the first Hello packet is received from a new neighbor, the neighbor is entered into the neighbor table in the initialization state. Once bidirectional communication is established, the neighbor state becomes two-way. ExStart and exchange states come next, as the two interfaces exchange their link-state database. Once this is all complete, the neighbor moves into the full state, which signifies full adjacency. If the neighbor fails to send any Hello packets in the dead interval, then the neighbor is moved to the down state and is no longer considered adjacent.

Adjacency

Not all OSPFv3 neighbors form full adjacencies. The establishment of adjacency depends on the network type and the presence of a Designated Router (DR). In some cases, only selected neighbors become fully adjacent and exchange Link-State Advertisements (LSAs), while others do not. For more information, see the Designated Routers.

Adjacency is established using Database Description (DD) packets, Link State Request (LSR) packets, and Link State Update (LSU) packets in OSPFv3.

The DD packet includes the link-state advertisement (LSA) headers from the link-state database of the neighbor. See the Link-State Database for more information.

The local router compares these headers with its own link-state database and determines which LSAs are new or updated. The local router sends an LSR packet for each LSA that it needs new or updated information on. The neighbor responds with an LSU packet. This exchange continues until both routers have the same link-state information.

Designated routers

In OSPFv3 networks with multiple routers, managing link-state advertisements (LSAs) efficiently is essential. Without coordination, each router could flood the network with identical LSAs, leading to unnecessary traffic. To address this, OSPFv3 may designate a single router, called the Designated Router (DR), to coordinate LSA flooding and represent the network to the wider OSPFv3 area. A Backup Designated Router (BDR) is also elected to take over if the DR becomes unavailable. See the Areas for more information on OSPFv3 area.

Network types are as follows:

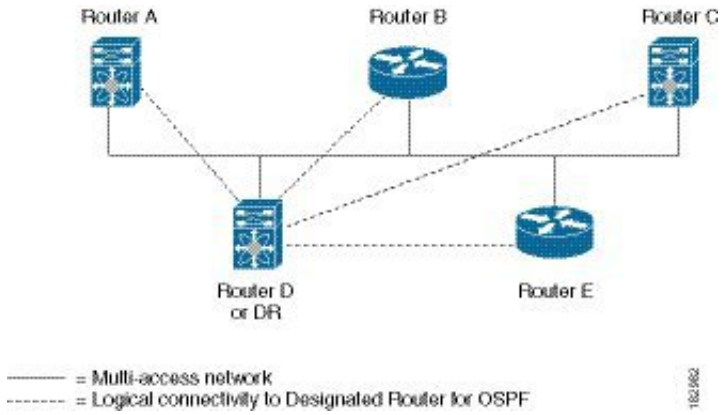
- Point-to-point:A network that exists only between two routers. All neighbors on a point-to-point network establish adjacency and there is no DR.
- Broadcast: A network with multiple routers that can communicate over a shared medium that allows broadcast traffic, such as Ethernet. OSPFv3 routers establish a DR and BDR that controls LSA flooding on the network. OSPFv3 uses the well-known IPv6 multicast addresses, FF02::5, and a MAC address of 0100.5300.0005 to communicate with neighbors.

The DR and BDR are selected based on the information in the Hello packet. When an interface sends a Hello packet, it sets the priority field and the DR and BDR field if it knows who the DR and BDR are. The routers follow an election procedure based on which routers declare themselves in the DR and BDR fields and the priority field in the Hello packet. As a final determinant, OSPFv3 chooses the highest router IDs as the DR and BDR.

All other routers establish adjacency with the DR and the BDR and use the IPv6 multicast address FF02::6 to send LSA updates to the DR and BDR. Figure Figure 1: DR in Multi-Access Network , on page 4 shows this adjacency relationship between all routers and the DR.

DRs are based on a router interface. A router might be the DR for one network and not for another network on a different interface.

Figure 1: DR in Multi-Access Network



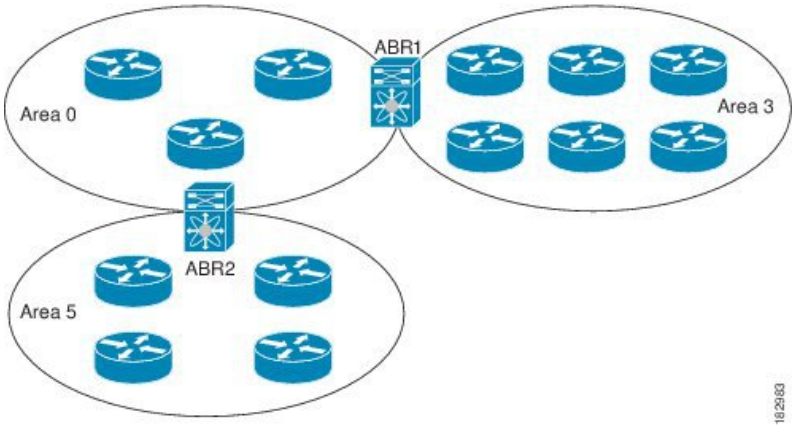
Areas

An area is a logical division of routers and links within an OSPFv3 domain that creates separate subdomains. LSA flooding is contained within an area, and the link-state database is limited to links within the area. You can assign an area ID to the interfaces within the defined area. The Area ID is a 32-bit value that can be expressed as a number or in dotted decimal notation, such as 10.2.3.1. By dividing an OSPFv3 network into areas, you can limit the CPU and memory requirements that OSPFv3 puts on the routers.

Cisco NX-OS always displays the area in dotted decimal notation.

If you define more than one area in an OSPFv3 network, you must also define the backbone area, which has the reserved area ID of 0. If you have more than one area, then one or more routers become area border routers (ABRs). An ABR connects to both the backbone area and at least one other defined area.

Figure 2: OSPFv3 Areas



The ABR has a separate link-state database for each area which it connects to. The ABR sends Inter-Area Prefix (type 3) LSAs from one connected area to the backbone area. See Route Summarization for more details.

The backbone area sends summarized information about one area to another area. In the figure Figure 2: OSPFv3 Areas, on page 5, Area 0 sends summarized information about Area 5 to Area 3.

OSPFv3 defines one other router type that is the autonomous system boundary router (ASBR). This router connects an OSPFv3 area to another autonomous system. An autonomous system is a network controlled by a single technical administration entity. OSPFv3 can redistribute its routing information into another autonomous system or receive redistributed routes from another autonomous system. For more information, see Advanced Features.

Link-State advertisements

OSPFv3 uses link-state advertisements (LSAs) to build its routing table.

Link-State advertisement types

OSPFv3 uses link-state advertisements (LSAs) to build its routing table.

The table shows the LSA types that are supported by Cisco NX-OS.

Type	Names	Description
1	Router LSA	LSA sent by every router. This LSA includes the state and cost of all links but does not include prefix information. Router LSAs trigger an SPF recalculation. Router LSAs are flooded to the local OSPFv3 area.
2	Network LSA	LSA sent by the DR. This LSA lists all routers in the multi-access network but does not include prefix information. Network LSAs trigger an SPF recalculation. See the Designated Routers section.

Type	Names	Description
3	Inter-Area Prefix LSA	LSA sent by the area border router to an external area for each destination in local area. This LSA includes the link cost from the border router to the local destination. See the Areas section.
4	Inter-Area Router LSA	LSA sent by the area border router to an external area. This LSA advertises the link cost to the ASBR only. See the Areas section.
5	AS External LSA	LSA generated by the ASBR. This LSA includes the link cost to an external autonomous system destination. AS External LSAs are flooded throughout the autonomous system. See the Areas section.
7	Type-7 LSA	LSA generated by the ASBR within an NSSA. This LSA includes the link cost to an external autonomous system destination. Type-7 LSAs are flooded only within the local NSSA. See the Areas section.
8	Link LSA	LSA sent by every router, using a link-local flooding scope. (see the Flooding and LSA Group Pacing section). This LSA includes the link-local address and IPv6 prefixes for this link.
9	Intra-Area Prefix LSA	LSA sent by every router. This LSA includes any prefix or link state changes. Intra-Area Prefix LSAs are flooded to the local OSPFv3 area. This LSA does not trigger an SPF recalculation.
11	Grace LSA	LSA sent by a restarting router, using a link-local flooding scope. This LSA is used for a graceful restart of OSPFv3. See the High Availability and Graceful Restart section.

Link cost

Each OSPFv3 interface is assigned a link cost. The cost is an arbitrary number. By default, Cisco NX-OS assigns a cost that is the configured reference bandwidth divided by the interface bandwidth. By default, the reference bandwidth is 40 Gbps. The link cost is carried in the LSA updates for each link.

Flooding and LSA group pacing

OSPFv3 floods LSA updates to different sections of the network, depending on the LSA type. OSPFv3 uses these flooding scopes:

- Link-local: LSA is flooded only on the local link. Used for Link LSAs and Grace LSAs.
- Area-local: LSA is flooded throughout a single OSPF area only. Used for Router LSAs, Network LSAs, Inter-Area-Prefix LSAs, Inter-Area-Router LSAs, and Intra-Area-Prefix LSAs.
- AS scope: LSA is flooded throughout the routing domain. An AS scope is used for AS External LSAs.

LSA flooding guarantees that all routers in the network have identical routing information. LSA flooding depends on the OSPFv3 area configuration. See Areas to know more about area.

The LSAs are flooded based on the link-state refresh time. Each LSA has its own link-state refresh time. The default refresh time is 30 minutes.

You can control the flooding rate of LSA updates in your network by using the LSA group pacing feature. LSA group pacing can reduce high CPU or buffer utilization. This feature groups LSAs with similar link-state refresh times to allow OSPFv3 to pack multiple LSAs into an OSPFv3 Update message.

By default, LSAs with link-state refresh times within 10 seconds of each other are grouped together. You should lower this value for large link-state databases or raise it for smaller databases to optimize the OSPFv3 load on your network.

Link-State database

This database contains all the collected link-state advertisements (LSAs) and includes information on all the routes through the network. OSPFv3 uses this information to calculate the bast path to each destination and populates the routing table with these best paths. Each router maintains a link-state database for the OSPFv3 network.

LSAs are removed from the link-state database if no LSA update has been received within a set interval, called the MaxAge. Routers flood a repeat of the LSA every 30 minutes to prevent accurate link-state information from being aged out. Cisco NX-OS supports the LSA grouping feature to prevent all LSAs from refreshing at the same time. For more information, see the Flooding and LSA Group Pacing section.

Multi-Area adjacency

OSPFv3 multi-area adjacency allows you to configure a link on the primary interface that is in more than one area. This link becomes the preferred intra-area link in those areas.

Multi-area adjacency establishes a point-to-point unnumbered link in an OSPFv3 area that provides a topological path for that area. The primary adjacency uses the link to advertise an unnumbered point-to-point link in the Router LSA for the corresponding area when the neighbor state is full.

The multi-area interface 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. The multi-area interface establishes a neighbor relationship with the corresponding multi-area interface on the neighboring router. See the Configuring Multi-Area Adjacency for more information.

OSPFv3 and the IPv6 unicast RIB

OSPFv3 runs the Dijkstra shortest path first algorithm on the link-state database. This algorithm selects the best path to each destination based on the sum of all the link costs for each link in the path. The shortest path for each destination is then put in the OSPFv3 route table. When the OSPFv3 network is converged, this route table feeds into the IPv6 unicast Routing Information Base (RIB). OSPFv3 communicates with the IPv6 unicast RIB to do these:

- Add or remove routes.
- Handle route redistribution from other protocols.
- Provide convergence updates to remove stale OSPFv3 routes and for stub router advertisements. See Multiple OSPFv3 Instances for more information.

OSPFv3 also runs a modified Dijkstra algorithm for fast recalculation for Inter-Area Prefix, Inter-Area Router, AS-External, type-7, and Intra-Area Prefix (type 3, 4, 5, 7, 8) LSA changes.

Address family support

Cisco NX-OS supports multiple address families, such as unicast IPv6 and multicast IPv6. OSPFv3 features that are specific to an address family are these:

- Default routes
- Route summarization
- Route redistribution
- Filter lists for border routers
- SPF optimization

Use the **address-family ipv6 unicast** command to enter the IPv6 unicast address family configuration mode when configuring these features.

Authentication and encryption

You can configure authentication on OSPFv3 messages to prevent unauthorized or invalid routing updates in your network.

RFC 4552 provides authentication to OSPFv3 using an IPv6 Authentication Header (AH) or Encapsulating Security Payload (ESP) extension header. Cisco NX-OS supports RFC 4552 by using the IPv6 AH header to authenticate OSPFv3 packets.

Cisco NX-OS supports the IP Security (IPSec) authentication method and the Message Digest 5 (MD5) or Secure Hash Algorithm 1 (SHA-1) algorithm to authenticate OSPFv3 packets. OSPFv3 IPSec authentication supports static keys using commands.

Cisco NX-OS also supports the IPSec ESP method for both encryption and authentication of OSPFv3 messages. Encryption supports AES or 3DES algorithm for ESP encryption and SHA-1 or NULL for ESP authentication.

Beginning with Cisco NX-OS Release 10.4(1)F, Cisco NX-OS supports configuring encryption or authentication algorithms and keys using the keychain option.

You can configure IPSec encryption or authentication for an OSPFv3 process, an area, and/or an interface. The authentication configuration is inherited from process to area to interface level. If authentication is configured at all three levels, the interface configuration takes precedence over an area configurations, and an area configuration takes precedence over the process level.

Advanced features

Cisco NX-OS supports advanced OSPFv3 features that enhance the usability and scalability of OSPFv3 in the network.

Stub areas

A stub area is an area that does not allow AS External (type 5) LSAs. These LSAs are usually flooded throughout the local autonomous system to propagate external route information. You can limit the amount

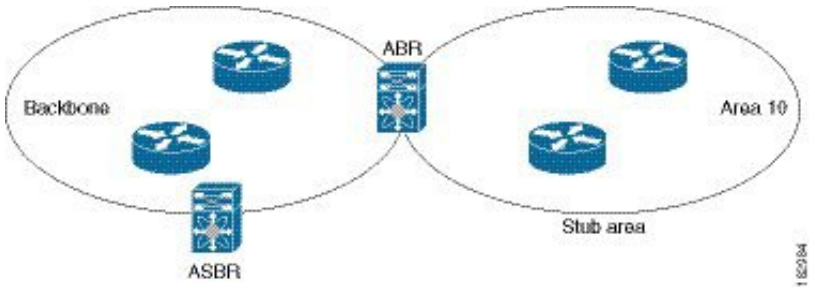
of external routing information that floods an area by making it a stub area. See Link-State advertisements, on page 5 to know more about LSA.

These are the requirements for stub area:

- All routers in the stub area are stub routers. See the Stub Routing section.
- No ASBR routers exist in the stub area.
- You cannot configure virtual links in the stub area.

The figure Stub area shows an example an OSPFv3 autonomous system where all routers in area 0.0.0.10 have to go through the ABR to reach external autonomous systems. Area 0.0.0.10 can be configured as a stub area.

Figure 3: Stub Area



Stub areas use a default route for all traffic that needs to go through the backbone area to the external autonomous system. The default route is an Inter-Area-Prefix LSA with the prefix length set to 0 for IPv6.

Not-So-Stubby Area

A Not-so-Stubby Area (NSSA) is similar to a stub area, except that an NSSA allows you to import autonomous system external routes within an NSSA using redistribution.

The NSSA ASBR redistributes these routes and generates NSSA External (type 7) LSAs that it floods throughout the NSSA. You can optionally configure the ABR that connects the NSSA to other areas to translate this NSSA External LSA to AS External (type 5) LSAs. The ABR then floods these AS External LSAs throughout the OSPFv2 autonomous system. Summarization and filtering are supported during the translation. See Link-State advertisements, on page 5 for information about NSSA External LSAs.

You can, for example, use NSSA to simplify administration if you are connecting a central site using OSPFv3 to a remote site that is using a different routing protocol. Before NSSA, the connection between the corporate site border router and a remote router could not be run as an OSPFv3 stub area because routes for the remote site could not be redistributed into a stub area. With NSSA, you can extend OSPFv3 to cover the remote connection by defining the area between the corporate router and remote router as an NSSA. (see the Configuring NSSA section).

The backbone Area 0 cannot be an NSSA

Note Beginning with Cisco NX-OS Release 9.3(1), OSPF became compliant with RFC 3101 section 2.5(3). When an Area Border Router attached to a Not-so-Stubby Area receives a default route LSA with P-bit clear, it should be ignored. OSPF had been previously adding the default route under these conditions.

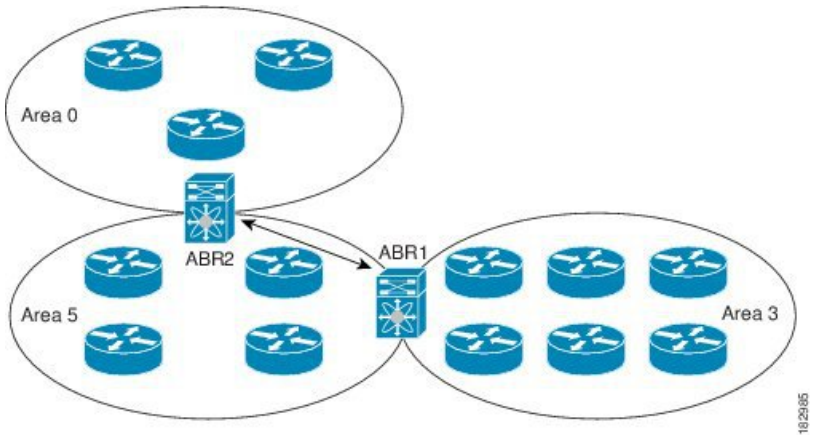
If you have already designed your networks with RFC non-compliant behavior and expect a default route to be added on NSSA ABR, you will see a change in behavior when you upgrade to Cisco NX-OS Release 9.3(1) and later.

If you decide to continue with the old behavior, you have the option to enable it with the **default-route nssa-abr pbit-clear** command. This command was implemented in Cisco NX-OS Release 9.3(1).

Virtual links

Virtual links allow you to connect an OSPFv3 area ABR to a backbone area ABR when a direct physical connection is not available. The figure Figure 4: Virtual Links, on page 10 shows a virtual link that connects Area 3 to the backbone area through Area 5.

Figure 4: Virtual Links



You can also use virtual links to temporarily recover from a partitioned area, which occurs when a link within the area fails, isolating part of the area from reaching the designated ABR to the backbone area.

Route redistribution

Route redistribution is a processes that allows OSPFv3 to learn routes from other routing protocols. You can configure OSPFv3 to assign a specific link cost to these redistributed routes or apply a default link cost to all of them.

OSPFv3 can learn routes from other routing protocols by using route redistribution. See the Route Redistribution. You configure OSPFv3 to assign a link cost for these redistributed routes or a default link cost for all redistributed routes.

Route redistribution uses route maps to control which external routes are redistributed. You must configure a route map with the redistribution to control which routes are passed into OSPFv3. A route map allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on. You can use route maps to modify parameters in the AS External (type 5) and NSSA External (type 7)

LSAs before these external routes are advertised in the local OSPFv3 autonomous system. For more information, see Configuring Route Policy Manager.

Route summarization

Route summarization simplifies route tables by replacing more-specific addresses with an address that represents all the specific addresses. For example, you can replace 2010:11:22:0:1000::1 and 2010:11:22:0:2000:679:1 with one summary address, 2010:11:22::/32.

You can use route summarization to reduce the number of unique routes that are flooded to every OSPF-enabled router because OSPFv3 shares all learned routes with every OSPF-enabled router.

Area border router (ABR) summarization is the process of aggregating routing information at the boundaries between OSPF areas. You typically perform summarization at ABRs to reduce the size of routing tables and improve network efficiency. Although you can configure summarization between any two areas, it is best to summarize routes toward the backbone area. This approach ensures that the backbone receives all summarized addresses and then injects these aggregates into other areas, optimizing routing updates and minimizing unnecessary detail across the network. There are two types of summarization:

- Inter-area route summarization
- External route summarization

Inter-area route summarization: You configure inter-area route summarization on ABRs, summarizing routes between areas in the autonomous system. To take advantage of summarization, you should assign network numbers in areas in a contiguous way to be able to lump these addresses into one range.

External route summarization: It is specific to external routes that are injected into OSPFv3 using route redistribution. Make sure that external ranges that are being summarized are contiguous. Summarizing overlapping ranges from two different routers could cause packets to be sent to the wrong destination. Configure external route summarization on ASBRs that are redistributing routes into OSPF.

When you configure a summary address, Cisco NX-OS automatically configures a discard route for the summary address to prevent routing black holes and route loops.

High Availability and Graceful Restart

High availabilities in Cisco NX-OS provide a multilevel architecture that ensures continuous network operation and minimal downtime. One key protocol supporting this is OSPFv3, which includes mechanisms for stateful restart and graceful restart to maintain routing stability during process interruptions.

OSPFv3 supports stateful restart, which is also referred to as non-stop routing (NSR). If OSPFv3 experiences problems, it attempts to restart from its previous run-time state. The neighbors do not register any neighbor event in this case. If the first restart is not successful and another problem occurs, OSPFv3 attempts a graceful restart.

A graceful restart, or non-stop forwarding (NSF), allows OSPFv3 to remain in the data forwarding path through a process restart. When OSPFv3 needs to perform a graceful restart, it sends a link-local Grace (type 11) LSA. This restarting OSPFv3 platform is called NSF capable.

The Grace LSA includes a grace period, which is a specified time that the neighbor OSPFv3 interfaces hold onto the LSAs from the restarting OSPFv3 interface. (Typically, OSPFv3 tears down the adjacency and discards all LSAs from a down or restarting OSPFv3 interface.) The participating neighbors, which are called

NSF helpers, keep all LSAs that originate from the restarting OSPFv3 interface as if the interface was still adjacent.

When the restarting OSPFv3 interface is operational again, it rediscovers its neighbors, establishes adjacency, and starts sending its LSA updates again. At this point, the NSF helpers recognize that the graceful restart has finished.

Stateful restart is used in these scenarios:

- First recovery attempt after the process experiences problems
- User-initiated switchover using the **system switchover** command

Graceful restart is used in these scenarios:

- Second recovery attempt after the process experiences problems within a 4-minute interval
- Manual restart of the process using the **restart ospfv3** command
- Active supervisor removal
- Active supervisor reload using the **reload module** *active-sup* command

Multiple OSPFv3 instances

Cisco NX-OS supports multiple instances of the OSPFv3 protocol. By default, every instance uses the same system router ID. You must manually configure the router ID for each instance if the instances are in the same OSPFv3 autonomous system. For the number of supported OSPFv3 instances, see the Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.

The OSPFv3 header includes an instance ID field to identify that OSPFv3 packet for a particular OSPFv3 instance. You can assign the OSPFv3 instance. The interface drops all OSPFv3 packets that do not have a matching OSPFv3 instance ID in the packet header.

Cisco NX-OS allows only one OSPFv3 instance on an interface.

SPF Optimization

The shortest path first (SPF) algorithm optimization in Cisco NX-OS is a set of techniques that

- improve the efficiency of SPF calculations,
- reduce CPU load during routing updates, and
- speed up network convergence.

Cisco NX-OS optimizes the SPF algorithm in the following ways:

- Partial SPF for Network (type 2) LSAs, Inter-Area Prefix (type 3) LSAs, and AS External (type 5) LSAs—When there is a change on any of these LSAs, Cisco NX-OS performs a faster partial calculation rather than running the whole SPF calculation.
- SPF timers—You can configure different timers for controlling SPF calculations. These timers include exponential backoff for subsequent SPF calculations. The exponential backoff limits the CPU load of multiple SPF calculations.

BFD

Bidirectional forwarding detection (BFD) is a detection protocol that provides fast forwarding-path failure detection times. BFD provides subsecond failure detection between two adjacent devices and can be less CPU-intensive than protocol hello messages, because some of the BFD load can be distributed onto the data plane on supported modules. See the Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide for more information.

Virtualization support

Cisco NX-OS supports multiple process instances of OSPFv3. Each OSPFv3 instance can support multiple virtual routing and forwarding (VRF) instances, up to the system limit. For the number of supported OSPFv3 instances, see the Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.

Prerequisites for OSPFv3

These are the prerequisites for OSPFv3:

- You must be familiar with routing fundamentals to configure OSPFv3.
- You must be logged on to the switch.
- You have configured at least one interface for IPv6 that is capable of communicating with a remote OSPFv3 neighbor.
- You have installed the Enterprise Services license.
- You have completed the OSPFv3 network strategy and planning for your network. For example, you must decide whether multiple areas are required.
- You have enabled OSPF. See the Enabling OSPFv3 topic.
- You are familiar with IPv6 addressing and basic configuration. See IPv6 Addresses for information on IPv6 routing and addressing.

Guidelines and limitations for OSPFv3

OSPFv3 has the following configuration guidelines and limitations:

- The **graceful-restart planned-only** command under OSPFv2 on reload converts to the **graceful-restart** command.

  This is not causing any impact on the functionality. If the **graceful-restart planned-only** is not in the configuration, this problem is not applicable for that device.

  This occurs when the Cisco NX-OS release is 9.3(2) and CSCvs57583 is not included in the release. A workaround is to unconfigure the **graceful-restart** command and reconfigure the old command.

- Names in the prefix-list are case-insensitive. We recommend using unique names. Do not use the same name by modifying uppercase and lowercase characters. For example, CTCPrimaryNetworks and CtcPrimaryNetworks are not two different entries.

- If you enter the **no graceful-restart planned only** command, graceful restart is disabled.
- Cisco NX-OS displays areas in dotted decimal notation regardless of whether you enter the area in decimal or dotted decimal notation.
- If you configure OSPFv3 in a virtual port channel (vPC) environment, use the following timer commands in router configuration mode on the core switch to ensure fast OSPF convergence when a vPC peer link is shut down:

```
switch(config-router)# timers throttle spf 1 50 50
switch(config-router)# timers lsa-arrival 10
```

- In scaled scenarios, when the number of interfaces and link-state advertisements in an OSPF process is large, the snmp-walk on OSPF MIB objects is expected to time out with a small-values timeout at the SNMP agent. If you observe a timeout on the querying SNMP agent while polling OSPF MIB objects, increase the timeout value on the polling SNMP agent.
- The following guidelines and limitations apply to the administrative distance feature:
  - When an OSPF route has two or more equal cost paths, configuring the administrative distance is non-deterministic for the **match ip route-source** command.
  - For matching route sources in OSPFv3 routes, you must configure **match ip route-source** instead of **match ipv6 route-source** because the route sources and router IDs for OSPFv3 are IPv4 addresses.
  - Configuring the administrative distance is supported only for the **match route-type**, **match ipv6 address prefix-list**, and **match ip route-source prefix-list** commands. The other match statements are ignored.
  - The discard route is always assigned an administrative distance of 220. No configuration in the table map applies to OSPF discard routes.
  - There is no preference among the **match route-type**, **match ipv6 address**, and **match ip route-source** commands for setting the administrative distance of OSPF routes. In this way, the behavior of the table map for setting the administrative distance in Cisco NX-OS OSPF is different from the behavior in Cisco IOS OSPF.
- If you configure the **delay restore** *seconds* command in vPC configuration mode and if the VLANs on the multichassis EtherChannel trunk (MCT) are announced by OSPFv2 or OSPFv3 using switch virtual interfaces (SVIs), those SVIs are announced with MAX_LINK_COST on the vPC secondary node during the configured time. As a result, all route or host programming completes after the vPC synchronization operation (on a peer reload of the secondary vPC node) before attracting traffic. This behavior allows for minimal packet loss for any north-to-south traffic.
- If you configure the same *area-id* for the primary area and any multiarea, the configuration is accepted without displaying an error. When you configure the primary area and any multiareas, do not use the same *area-id.*

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

- If you use the **network** *ip address mask* command under OSPF, an error message will be displayed, and you will be prompted to enable OSPF under an interface with **area** *area id* command.

- It is recommended that you use the OSPF default timers (hello-interval:10 and dead-interval:40). For better convergence time, you can use the BFD along with OSPF. This combination will give sub-second link/adjacency flaps detection and very low convergence time.
- While OSPF support are aggressive timers, these are not commended as aggressive timers will bring the adjancency down quickly as well as cause CPU churn. We recommend you to use the default timers and use BFD (Bidirectional Forwarding Detection) to get sub-second failure detection.
- The Route Policy Manager (RPM) does not support IPv6 redistribution for filtering BGP route updates based on the BGP community attribute using community lists in a route map. This functionality is only available for IPv4 redistribution.
- Beginning with Cisco NX-OS Release 10.3(1)F, OSPFv3 is supported on the Cisco Nexus 9808 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, OSPFv3 is supported on the Cisco Nexus 9804 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, the keychain support is provided for OSPFv3 encryption and authentication commands on the Cisco NX-OS switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, OSPFv3 is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with 9808 and 9804 switches.

Default settings

The table lists the default settings for OSPFv3 parameters.

Table 1: Default OSPFv3 Parameters

Parameters	Default
Administrative distance	110
Hello interval	10 seconds
Dead interval	40 seconds
Discard routes	Enabled
Graceful restart grace period	60 seconds
Graceful restart notify period	15 seconds
OSPFv3 feature	Disabled
Stub router advertisement announce time	600 seconds
Reference bandwidth for link cost calculation	40 Gb/s
LSA minimal arrival time	1000 milliseconds
LSA group pacing	10 seconds
SPF calculation initial delay time	200 milliseconds
SPF calculation minimum hold time	1000 milliseconds

Parameters	Default
SPF calculation maximum wait time	5000 milliseconds

Configure basic OSPFv3

Configure OSPFv3 after you have designed your OSPFv3 network.

Enable OSPFv3

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the [**no**] **feature ospfv3** command to enable OSPFv3.

Example:

```
switch(config)# feature ospfv3
```

Using the **no** keyword with this command disables the OSPFv3 feature and removes all associated configuration.

Step 3 (Optional) Use the **show feature** command to see see enabled and disabled features.

Example:

```
switch(config)# show feature
```

Step 4 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config)# copy running-config
startup-config
```

Create an OSPFv3 instance

The first step in configuring OSPFv3 is to create an instance or OSPFv3 instance. You assign a unique instance tag for this OSPFv3 instance. The instance tag can be any string. For each OSPFv3 instance, you can also configure the following optional parameters:

- Router ID—Configures the router ID for this OSPFv3 instance. If you do not use this parameter, the router ID selection algorithm is used. , see the Router IDs section.
- Administrative distance—Rates the trustworthiness of a routing information source. For more information, see the Administrative Distance section.

- Log adjacency changes—Creates a system message whenever an OSPFv3 neighbor changes its state.
- Name lookup—Translates OSPF router IDs to hostnames, either by looking up the local hosts database or querying DNS names in IPv6.
- Maximum paths—Sets the maximum number of equal paths that OSPFv3 installs in the route table for a particular destination. Use this parameter for load balancing between multiple paths.
- Reference bandwidth—Controls the calculated OSPFv3 cost metric for a network. The calculated cost is the reference bandwidth divided by the interface bandwidth. You can override the calculated cost by assigning a link cost when a network is added to the OSPFv3 instance. For more information, see the Configuring Networks in OSPFv3 section.

For more information about OSPFv3 instance parameters, see the Configuring Advanced OSPFv3 section.

Before you begin

You must enable OSPFv3 (see the Enabling OSPFv3 section).

- Ensure that the OSPFv3 instance tag that you plan on using is not already in use on this router.
- Use the **show ospfv3** *instance-tag* command to verify that the instance tag is not in use.
- OSPFv3 must be able to obtain a router identifier (for example, a configured loopback address) or you must configure the router ID option.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the [**no**] **router ospfv3** *instance-tag* command to create a new OSPFv3 instance with the configured instance tag.

Example:

```
switch(config)# router ospfv3 201
 switch(config-router)#
```

Note
The **no router ospfv3** *instance tag* command does not remove OSPF configuration in interface mode. You must manually remove any OSPFv3 commands configured in interface mode.

Step 3 (Optional) Use the **router-id** *ip-address* command to configures the OSPFv3 router ID.

Example:

```
switch(config-router)# router-id
 192.0.2.1
```

This ID uses the dotted decimal notation and identifies this OSPFv3 instance and must exist on a configured interface in the system.

Step 4 (Optional) Use the **show ipv6 ospfv3** *instance-tag* command to see OSPFv3 information.

Example:

```
switch(config-router)# show ipv6 ospfv3
 201
```

Step 5 (Optional) Use the **log-adjacency-changes** [**detail**] command to generate a system message whenever a neighbor changes state.

Example:

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

Step 6 (Optional) Use the **passive-interface default** command to suppress routing updates on all interfaces.

Example:

```
switch(config-router)# passive-interface
 default
```

This command is overridden by the VRF or interface command mode configuration.

Step 7 (Optional) Use the **distance** *number* command to configure the administrative distance for this OSPFv3 instance.

Example:

```
switch(config-router-af)# distance 25
```

The range is from 1 to 255. The default is 110.

Step 8 (Optional) Use the **maximum-paths** *paths* command to configure the maximum number of equal OSPFv3 paths to a destination in the route table.

Example:

```
switch(config-router-af)# maximum-paths
 4
```

The range is from 1 to 16. The default is 8. This command is used for load balancing.

Step 9 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch(config)# copy running-config startup-config
```

Example

This example shows how to create an OSPFv3 instance:

```
switch# <userinput>configure terminal</userinput>
switch(config)# <userinput>router ospfv3 201</userinput>
 switch(config-router)# <userinput>copy running-config startup-config</userinput>
```

Configure networks in OSPFv3

You can configure a network to OSPFv3 by associating it through the interface that the router uses to connect to that network (see the Neighbors section). You can add all networks to the default backbone area (Area 0), or you can create new areas using any decimal number or an IP address.

All areas must connect to the backbone area either directly or through a virtual link.

OSPFv3 is not enabled on an interface until you configure a valid IPv6 address for that interface.

Before you begin

You must enable OSPFv3 (see the Enabling OSPFv3 section).

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the **interface***interface-type slot/port* command to enter interface configuration mode.

Example:

```
switch(config)# interface ethernet 1/2
 switch(config-if)#
```

Step 3 Use the **ipv6 address***ipv6-prefix/length* command to assign an IPv6 address to this interface.

Example:

```
switch(config-if)# ipv6 address
 2001:0DB8::1/48
```

Step 4 Use the **ipv6 router ospfv3***instance-tag***area***area-id* [**secondaries none**] command to add the interface to the OSPFv3 instance and area.

Example:

```
switch(config-if)# ipv6 router ospfv3
 201 area 0
```

Step 5 (Optional) Use the **show ipv6 ospfv3***instance-tag***interface***interface-type slot/port* command to see OSPFv3 information.

Example:

```
switch(config-if)# show ipv6 ospfv3 201
 interface ethernet 1/2
```

Step 6 (Optional) Use the **ospfv3 cost***number* command to configure the OSPFv3 cost metric for this interface.

Example:

```
switch(config-if)# ospfv3 cost 25
```

The default is to calculate a cost metric, based on the reference bandwidth and interface bandwidth. The range is from 1 to 65535.

Step 7 (Optional) Use the **ospfv3 dead-interval***seconds* command to configure the OSPFv3 dead interval, in seconds.

Example:

```
switch(config-if)# ospfv3 dead-interval 50
```

The range is from 1 to 65535. The default is four times the hello interval, in seconds.

Step 8 (Optional) Use the **ospfv3 hello-interval***seconds* command to configure the OSPFv3 hello interval, in seconds.

Example:

```
switch(config-if)# ospfv3 hello-interval
 25
```

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

Step 9 (Optional) Use the **ospfv3 instance***instance* command to configure the OSPFv3 instance ID.

Example:

```
switch(config-if)# ospfv3 instance 25
```

The range is from 0 to 255. The default is 0. The instance ID is link-local in scope.

Step 10 (Optional) Use the **ospfv3 mtu-ignore** command to configure OSPFv3 to ignore any IP maximum transmission unit (MTU) mismatch with a neighbor.

Example:

```
switch(config-if)# ospfv3 mtu-ignore
```

The default is to not establish adjacency if the neighbor MTU does not match the local interface MTU.

Step 11 (Optional) Use the **ospfv3 network** {**broadcast** | **point-point**} command to set the OSPFv3 network type.

Example:

```
switch(config-if)# ospfv3 network
 broadcast
```

Step 12 (Optional) Use the [**default** | **no**] **ospfv3 passive-interface** command to suppress routing updates on the interface.

Example:

```
switch(config-if)# ospfv3
 passive-interface
```

This command overrides the router or VRF command mode configuration. The **default** option removes this interface mode command and reverts to the router or VRF configuration, if present.

Step 13 (Optional) Use the **ospfv3 priority***number* command to configure the OSPFv3 priority, used to determine the DR for an area.

Example:

```
switch(config-if)# ospfv3 priority 25
```

The range is from 0 to 255. The default is 1. See the Designated Routers section.

Step 14 (Optional) Use the **ospfv3 shutdown** command to shut down the OSPFv3 instance on this interface.

Example:

```
switch(config-if)# ospfv3 shutdown
```

Step 15 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch(config)# copy running-config startup-config
```

Example

This example shows how to add a network area 0.0.0.10 in OSPFv3 instance 201:

```
switch# configure terminal
 switch(config)# interface ethernet 1/2
 switch(config-if)# ipv6 address 2001:0DB8::1/48
 switch(config-if)# ipv6 router ospfv3 201 area 0.0.0.10
 switch(config-if)# copy running-config startup-config
```

Configure advanced OSPFv3

Configure OSPFv3 after you have designed your OSPFv3 network.

Configure filter lists for border routers

You can separate your OSPFv3 domain into a series of areas that contain related networks. All areas must connect to the backbone area through an area border router (ABR). OSPFv3 domains can connect to external domains as well through an autonomous system border router (ASBR). See the Areas section.

ABRs have the following optional configuration parameters:

- Area range—Configures route summarization between areas. For more information, see the Configuring Route Summarization section.
- Filter list—Filters the Inter-Area Prefix (type 3) LSAs on an ABR that are allowed in from an external area.

ASBRs also support filter lists.

Before you begin

Create the route map that the filter list uses to filter IP prefixes in incoming or outgoing Inter-Area Prefix (type 3) LSAs. See Configuring Route Policy Manager.

Procedure

Step 1 Use the **router ospfv3** *instance-tag* command to create a new OSPFv3 instance with the configured instance tag.

Example:

```
switch(config)# router ospfv3 201
 switch(config-router)#
```

Step 2 Use the **address-family ipv6 unicast** command to enter IPv6 unicast address family mode.

Example:

```
switch(config-router)# address-family
 ipv6 unicast
 switch(config-router-af)#
```

Step 3 Use the **area** *area-id* **filter-list route-map** *map-name* {**in | out**} command to filter incoming or outgoing Inter-Area Prefix (type 3) LSAs on an ABR.

Example:

```
switch(config-router-af)# area 0.0.0.10
 filter-list route-map FilterLSAs in
```

Step 4 (Optional) Use the **show ipv6 ospfv3 policy statistics area** *id* **filter-list** {**in** | **out**} command to see OSPFv3 policy information.

Example:

```
switch(config-router-af)# show ipv6 ospfv3
 policy statistics area 0.0.0.10
 filter-list in
```

Step 5 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config-router-af)# copy running-config
 startup-config
```

Example

This example shows how to configure a filter list for a route map:

```
switch# <userinput>configure terminal</userinput>
 switch(config)# <userinput>router ospfv3 201</userinput>
 switch(config-router)# <userinput>address-family ipv6 unicast</userinput>
 switch(config-router-af)# <userinput>area 0.0.0.10 filter-list route-map FilterLSAs
in</userinput>
 switch(config-router-af)# <userinput>copy running-config startup-config</userinput>
```

Configure stub areas

You can configure a stub area for part of an OSPFv3 domain where external traffic is not necessary. Stub areas block AS External (type 5) LSAs, limiting unnecessary routing to and from selected networks. See the Stub Area section. You can optionally block all summary routes from going into the stub area.

Before you begin

- You must enable OSPF (see the Enabling OSPFv3 section).
- Ensure that there are no virtual links or ASBRs in the proposed stub area.

Procedure

Step 1 Use the **router ospfv3** *instance-tag* command to create a new OSPFv3 instance with the configured instance tag.

Example:

```
switch(config)# router ospfv3 201
 switch(config-router)#
```

Step 2 Use the **area** *area-id* **stub** command to create this area as a stub area.

Example:

```
switch(config-router)# area 0.0.0.10
 stub
```

Step 3 (Optional) Use the **address-family ipv6 unicast** command to enter IPv6 unicast address family mode.

Example:

```
switch(config-router)# address-family
 ipv6 unicast
 switch(config-router-af)#
```

Step 4 (Optional) Use the **area** *area-id* **default cost** *cost* command to set the cost metric for the default summary route sent into this stub area.

Example:

```
switch(config-router-af)# area 0.0.0.10
 default-cost 25
```

The range is from 0 to 16777215.

Step 5 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config-router-af)# copy running-config
 startup-config
```

Example

This shows how to create a stub area that blocks all summary route updates:

```
switch# <userinput>configure terminal</userinput>
 switch(config)# <userinput>router ospfv3 201</userinput>
 switch(config-router)# <userinput>area 0.0.0.10 stub no-summary</userinput>
 switch(config-router)# <userinput>copy running-config startup-config</userinput>
```

Configure a totally stubby area

You can create a totally stubby area and prevent all summary route updates from going into the stub area.

To create a totally stubby area, use the following command in router configuration mode:

Procedure

Use the **area** *area-id* **stub no-summary** command to create this area as a totally stubby area.

Example:

```
switch(config-router)# area 20 stub
 no-summary
```

Configure NSSA

You can configure an NSSA for part of an OSPFv3 domain where limited external traffic is required.

You can optionally translate this external traffic to an AS External (type 5) LSA and flood the OSPFv3 domain with this routing information. An NSSA can be configured with the following optional parameters:

- No redistribution—Redistributes routes that bypass the NSSA to other areas in the OSPFv3 autonomous system. Use this option when the NSSA ASBR is also an ABR.
- Default information originate—Generates a Type-7 LSA for a default route to the external autonomous system. Use this option on an NSSA ASBR if the ASBR contains the default route in the routing table. This option can be used on an NSSA ABR whether or not the ABR contains the default route in the routing table.
- Route map—Filters the external routes so that only those routes you want are flooded throughout the NSSA and other areas.
- No summary—Blocks all summary routes from flooding the NSSA. Use this option on the NSSA ABR.
- Translate—Translates Type-7 LSAs to AS External (type 5) LSAs for areas outside the NSSA. Use this command on an NSSA ABR to flood the redistributed routes throughout the OSPFv3 autonomous system. You can optionally suppress the forwarding address in these AS External LSAs.

Note The translate option requires a separate **area***area-id***nssa** command, preceded by the **area***area-id***nssa** command that creates the NSSA and configures the other options.

Before you begin

You must enable OSPF (see the Enabling OSPFv3 section).

- Ensure that there are no virtual links in the proposed NSSA and that it is not the backbone area.

Procedure

Step 1 Use the **router ospfv3***instance-tag* command to create a new OSPFv3 instance with the configured instance tag.

Example:

```
switch(config)# router ospfv3 201
  switch(config-router)#
```

Step 2 Use the **area***area-id***nssa** [**no-redistribution**][**default-information-originate**] [**route-map***map-name*] [**no-summary**] command to create this area as an NSSA.

Example:

```
switch(config-router)# area 0.0.0.10
  nssa
```

Step 3 (Optional) Use the **area***area-id***nssa translate type7** {**always | never**} [**suppress-fa**] command to configure the NSSA to translate AS External (type 7) LSAs to NSSA External (type 5) LSAs.

Example:

```
switch(config-router)# area 0.0.0.10
  nssa translate type7 always
```

Step 4 (Optional) Use the **address-family ipv6 unicast** command to enter IPv6 unicast address family mode.

Example:

```
switch(config-router)# address-family
  ipv6 unicast
  switch(config-router-af)#
```

Step 5 (Optional) Use the **area***area-id***default cost***cost* command to set the cost metric for the default summary route sent into this NSSA.

Example:

```
switch(config-router-af)# area 0.0.0.10
  default-cost 25
```

The range is from 0 to 16777215.

Step 6 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config-router-af)# copy running-config
  startup-config
```

Example

This example shows how to create an NSSA that blocks all summary route updates:

```
switch# <userinput>configure terminal</userinput>
 switch(config)# <userinput>router ospfv3 201</userinput>
 switch(config-router)# <userinput>area 0.0.0.10 nssa no-summary</userinput>
 switch(config-router)# <userinput>copy running-config startup-config</userinput>
```

This example shows how to create an NSSA that generates a default route:

```
switch# <userinput>configure terminal</userinput>
 switch(config)# <userinput>router ospfv3 201</userinput>
 switch(config-router)# <userinput>area 0.0.0.10 nssa default-info-originate</userinput>
 switch(config-router)# <userinput>copy running-config startup-config</userinput>
```

This example shows how to create an NSSA that filters external routes and blocks all summary route updates:

```
switch# <userinput>configure terminal</userinput>
 switch(config)# <userinput>router ospfv3 201</userinput>
 switch(config-router)# <userinput>area 0.0.0.10 nssa route-map ExternalFilter
no-summary</userinput>
 switch(config-router)# <userinput>copy running-config startup-config</userinput>
```

<!--CSCvv84492-->This example shows how to create an NSSA and then configure the NSSA to always translate AS External (type 7) LSAs to NSSA External (type 5) LSAs:

```
switch# <userinput>configure terminal</userinput>
 switch(config)# <userinput>router ospfv3 201</userinput>
 switch(config-router)# <userinput>area 0.0.0.10 nssa</userinput>
 switch(config-router)# <userinput>area 0.0.0.10 nssa translate type 7 always</userinput>
 switch(config-router)# <userinput>copy running-config startup-config</userinput>
```

This example shows how to create an NSSA that blocks all summary route updates:

```
switch# <userinput>configure terminal</userinput>
 switch(config)# <userinput>router ospfv3 201</userinput>
 switch(config-router)# <userinput>area 0.0.0.10 nssa no-summary</userinput>
 switch(config-router)# <userinput>copy running-config startup-config</userinput>
```

Configure multi-area adjacency

You can add more than one area to an existing OSPFv3 interface. The additional logical interfaces support multi-area adjacency.

Before you begin

- You must enable OSPF (see the Enabling OSPFv3 section).
- Ensure that you have configured a primary area for the interface (see the Configuring Networks in OSPFv3 section).

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the **interface** *interface-type slot/port* command to enter interface configuration mode.

Example:

```
switch(config)# interface ethernet 1/2
  switch(config-if)#
```

Step 3 Use the **ipv6 router ospfv3** *instance-tag* **multi-area** *area-id* command to add the interface to another area.

Example:

```
switch(config-if)# ipv6 router ospfv3
  201 multi-area 3
```

Step 4 (Optional) Use the **show ipv6 ospfv3** *instance-tag* **interface** *interface-type slot/port* command to see OSPFv3 information.

Example:

```
switch(config-if)# show ipv6 ospfv3 201
  interface ethernet 1/2
```

Step 5 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch(config)# copy running-config startup-config
```

Example

This example shows how to add a second area to an OSPFv3 interface:

```
switch# <userinput>configure terminal</userinput>
 switch(config)# <userinput>interface ethernet 1/2</userinput>
 switch(config-if)# <userinput>ipv6 address 2001:0DB8::1/48</userinput>
 switch(config-if)# <userinput>ipv6 ospfv3 201 area 0.0.0.10</userinput>
 switch(config-if)# <userinput>ipv6 ospfv3 201 multi-area 20</userinput>
 switch(config-if)# <userinput>copy running-config startup-config</userinput>
```

Configure virtual links

A virtual link connects an isolated area to the backbone area through an intermediate area. See the Virtual Links section. You can configure the following optional parameters for a virtual link:

- Dead interval—Sets the time that a neighbor waits for a Hello packet before declaring the local router as dead and tearing down adjacencies.
- Hello interval—Sets the time between successive Hello packets.
- Retransmit interval—Sets the estimated time between successive LSAs.
- Transmit delay—Sets the estimated time to transmit an LSA to a neighbor.

Note You must configure the virtual link on both routers involved before the link becomes active.

Before you begin

You must enable OSPF. For more information, see the Enabling OSPFv3.

Procedure

Step 1 Use the **router ospfv3** *instance-tag* command to create a new OSPFv3 instance with the configured instance tag.

Example:

```
switch(config)# router ospfv3 201
  switch(config-router)#
```

Step 2 Use the **area** *area-id* **virtual-link** *router-id* command to create one end of a virtual link to a remote router.

Example:

```
switch(config-router)# area 0.0.0.10
  virtual-link 2001:0DB8::1
  switch(config-router-vlink)#
```

You must create the virtual link on that remote router to complete the link.

Step 3 (Optional) Use the **show ipv6 ospfv3 virtual-link** [**brief**] command to see see OSPFv3 virtual link information.

Example:

```
switch(config-router-vlink)# show ipv6 ospfv3
  virtual-link
```

Step 4 (Optional) Use the **dead-interval** *seconds* command to configure the OSPFv3 dead interval, in seconds.

Example:

```
switch(config-router-vlink)#
  dead-interval 50
```

The range is from 1 to 65535. The default is four times the hello interval, in seconds.

Step 5 (Optional) Use the **hello-interval** *seconds* command to configure the OSPFv3 hello interval, in seconds.

Example:

```
switch(config-router-vlink)#
  hello-interval 25
```

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

Step 6 (Optional) Use the **retransmit-interval** *seconds* command to configure the OSPFv3 retransmit interval, in seconds.

Example:

```
switch(config-router-vlink)#
  retransmit-interval 50
```

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

Step 7 (Optional) Use the **transmit-delay** *seconds* command to configure the OSPFv3 transmit-delay, in seconds.

Example:

```
switch(config-router-vlink)#
  transmit-delay 2
```

The range is from 1 to 450. The default is 1.

Step 8 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config-router-vlink)#
  copy running-config startup-config
```

Example

These examples show how to create a simple virtual link between two ABRs:

Configuration for ABR 1 (router ID 2001:0DB8::1) is as follows:

```
switch# <userinput>configure terminal</userinput>
 switch(config)# <userinput>router ospfv3 201</userinput>
 switch(config-router)# <userinput>area 0.0.0.10 virtual-link 2001:0DB8::10</userinput>
 switch(config-router-vlink)# <userinput>copy running-config startup-config</userinput>
```

Configuration for ABR 2 (router ID 2001:0DB8::10) is as follows:

```
switch# <userinput>configure terminal</userinput>
 switch(config)# <userinput>router ospfv3 201</userinput>
 switch(config-router)# <userinput>area 0.0.0.10 virtual-link 2001:0DB8::1</userinput>
 switch(config-router-vlink)# <userinput>copy running-config startup-config</userinput>
```

Configure redistribution

You can redistribute routes learned from other routing protocols into an OSPFv3 autonomous system through the ASBR.

You can configure the following optional parameters for route redistribution in OSPF:

- Default information originate—Generates an AS External (type 5) LSA for a default route to the external autonomous system.

Note Default information originate ignores **match** statements in the optional route map.

- Default metric—Sets all redistributed routes to the same cost metric.

Note If you redistribute static routes, Cisco NX-OS requires the **default-information originate** command to successfully redistribute the default static route starting in 7.0(3)I7(6).

Before you begin

You must enable OSPF (see the Enabling OSPFv3 section).

- Create the necessary route maps used for redistribution.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the **router ospfv3***instance-tag* command to create a new OSPFv3 instance with the configured instance tag.

Example:

```
switch(config)# router ospfv3 201
  switch(config-router)#
```

Step 3 Use the **address-family ipv6 unicast** command to enter IPv6 unicast address family mode.

Example:

```
switch(config-router)# address-family
  ipv6 unicast
  switch(config-router-af)#
```

Step 4 Use the **redistribute** {**bgp***id* | **direct** | **isis** *id* | **rip** *id* | **static** | **dhcpv6**} **route-map** *map-name* command to redistribute the selected protocol into OSPFv3 through the configured route map.

Example:

```
switch(config-router-af)# redistribute
  bgp route-map FilterExternalBGP
```

Note
If you redistribute static routes, Cisco NX-OS requires the **default-information originate** command to successfully redistribute the default static route starting in 7.0(3)I7(6).

Step 5 Use the **default-information originate** [**always**] [**route-map** *map-name*] command to create a default route into this OSPFv3 domain if the default route exists in the RIB.

Example:

```
switch(config-router-af)#
  default-information-originate route-map DefaultRouteFilter
```

Use the following optional keywords:

- **always** —Always generates the default route of 0.0.0. even if the route does not exist in the RIB.
- **route-map**—Generates the default route if the route map returns true.

Note
This command ignores **match** statements in the route map.

Step 6 Use the **default-metric** *cost* command to set the cost metric for the redistributed routes.

Example:

```
switch(config-router-af)# default-metric
  25
```

The range is from 1 to 16777214. This command does not apply to directly connected routes. Use a route map to set the default metric for directly connected routes.

Step 7 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config-router-af)# copy running-config
  startup-config
```

Example

This example shows how to redistribute the Border Gateway Protocol (BGP) into OSPFv3:

```
switch# <userinput>configure terminal</userinput>
 switch(config)# <userinput>router ospfv3 201</userinput>
 switch(config-router)# <userinput>address-family ipv6 unicast</userinput>
 switch(config-router-af)# <userinput>redistribute bgp route-map FilterExternalBGP</userinput>

 switch(config-router-af)# <userinput>copy running-config startup-config</userinput>
```

Limit the number of redistributed routes

Route redistribution can add many routes to the OSPFv3 route table. You can configure a maximum limit to the number of routes accepted from external protocols. OSPFv3 provides the following options to configure redistributed route limits:

- Fixed limit—Logs a message when OSPFv3 reaches the configured maximum. OSPFv3 does not accept any more redistributed routes. You can optionally configure a threshold percentage of the maximum where OSPFv3 logs a warning when that threshold is passed.
- Warning only—Logs a warning only when OSPFv3 reaches the maximum. OSPFv3 continues to accept redistributed routes.
- Withdraw—Starts the configured timeout period when OSPFv3 reaches the maximum. After the timeout period, OSPFv3 requests all redistributed routes if the current number of redistributed routes is less than the maximum limit. If the current number of redistributed routes is at the maximum limit, OSPFv3 withdraws all redistributed routes. You must clear this condition before OSPFv3 accepts more redistributed routes. You can optionally configure the timeout period.

Before you begin

You must enable OSPF (see the Enabling OSPFv3 section).

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the **router ospfv3** *instance-tag* command to create a new OSPFv3 instance with the configured instance tag.

Example:

```
switch(config)# router ospfv3 201
  switch(config-router)#
```

Step 3 Use the **address-family ipv6 unicast** command to enter IPv6 unicast address family mode.

Example:

```
switch(config-router)# address-family
  ipv6 unicast
  switch(config-router-af)#
```

Step 4 Use the **redistribute** {**bgp***id* | **direct** | **isis** *id* | **rip** *id* | **static**} **route-map** *map-name* command to redistribute the selected protocol into OSPFv3 through the configured route map.

Example:

```
switch(config-router-af)# redistribute
  bgp route-map FilterExternalBGP
```

Step 5 Use the **redistribute maximum-prefix***max* [*threshold*] [**warning-only** | **withdraw** [*num-retries timemout*]] command to specify a maximum number of prefixes that OSPFv2 distributes.

Example:

```
switch(config-router-af)# redistribute
  maximum-prefix 1000 75 warning-only
```

The range is from 0 to 65536. Optionally, specifies the following:

- *threshold*—Percent of maximum prefixes that triggers a warning message.
- **warning-only**—Logs a warning message when the maximum number of prefixes is exceeded.
- **withdraw**—Withdraws all redistributed routes and optionally tries to retrieve the redistributed routes. The *num-retries* range is from 1 to 12. The *timeout* range is from 60 to 600 seconds. The default is 300 seconds.

Step 6 (Optional) Use the **show running-config ospfv3** command to see the OSPFv3 configuration.

Example:

```
switch(config-router-af)# show
 running-config ospf
```

Step 7 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch(config)# copy running-config startup-config
```

Example

This example shows how to limit the number of redistributed routes into OSPF:

```
switch# <userinput>configure terminal</userinput>
 switch(config)# <userinput>router ospfv3 201</userinput>
 switch(config-router)# <userinput>address-family ipv6 unicast</userinput>
 switch(config-router-af)# <userinput>redistribute bgp route-map FilterExternalBGP</userinput>

 switch(config-router-af)# <userinput>redistribute maximum-prefix 1000 75</userinput>
```

Configure route summarization

You can configure route summarization for inter-area networks by configuring an address range that is summarized. You can also configure route summarization for external, redistributed routes by configuring a summary address for those routes on an ASBR. For more information, see the Route Summarization section.

Before you begin

You must enable OSPF (see the Enabling OSPFv3 section).

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the **router ospfv3***instance-tag* command to Creates a new OSPFv3 instance with the configured instance tag.

Example:

```
switch(config)# router ospfv3 201
switch(config-router)#
```

Step 3 Use the **address-family ipv6 unicast** command to enter IPv6 unicast address family mode.

Example:

```
switch(config-router)# address-family
ipv6 unicast
switch(config-router-af)#
```

Step 4 Use the **area** *area-id* **range** *ipv6-prefix/length* [**no-advertise**] [**cost** *cost*] command to create a summary address on an ABR for a range of addresses and optionally advertises this summary address in a Inter-Area Prefix (type 3) LSA.

Example:

```
switch(config-router-af)# area 0.0.0.10
range 2001:0DB8::/48 advertise
```

The cost range is from 0 to 16777215.

Step 5 Use the **summary-address** *ipv6-prefix/length* [**no-advertise**] [**tag** *tag*] command to

Example:

```
switch(config-router-af)#
summary-address 2001:0DB8::/48 tag 2
```

Creates a summary address on an ASBR for a range of addresses and optionally assigns a tag for this summary address that can be used for redistribution with route maps.

Step 6 (Optional) Use the **show ipv6 ospfv3 summary-address** command to see the information about OSPFv3 summary addresses.

Example:

```
switch(config-router-af)# show ipv6 ospfv3
summary-address
```

Step 7 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch(config)# copy running-config startup-config
```

Example

This example shows how to create summary addresses between areas on an ABR:

```
switch# <userinput>configure terminal</userinput>
switch(config)# <userinput>router ospfv3 201</userinput>
switch(config-router)# <userinput>address-family ipv6 unicast</userinput>
switch(config-router-af)# <userinput>area 0.0.0.10 range 2001:0DB8::/48</userinput>
switch(config-router-af)# <userinput>copy running-config startup-config</userinput>
```

This example shows how to create summary addresses on an ASBR:

```
switch# <userinput>configure terminal</userinput>
switch(config)# <userinput>router ospfv3 201</userinput>
switch(config-router)# <userinput>address-family ipv6 unicast</userinput>
switch(config-router-af)# <userinput>summary-address 2001:0DB8::/48</userinput>
switch(config-router-af)# <userinput>no discard route internal</userinput>
switch(config-router-af)# <userinput>copy running-config startup-config</userinput>
```

Configure the administrative distance of routes

You can set the administrative distance of routes added by OSPFv3 into the RIB.

The administrative distance is a rating of the trustworthiness of a routing information source. A higher value indicates a lower trust rating. Typically, a route can be learned through more than one routing protocol. The administrative distance is used to discriminate between routes learned from more than one routing protocol. The route with the lowest administrative distance is installed in the IP routing table.

Before you begin

Ensure that you have enabled OSPF (see the OSPFv3, on page 1 section).

- See the guidelines and limitations for this feature in the Guidelines and limitations for OSPFv3, on page 13 section.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the **router ospfv3** *instance-tag* command to create a new OSPFv3 instance with the configured instance tag.

Example:

```
switch(config)# router ospfv3 201
switch(config-router)#
```

Step 3 Use the **address-family ipv6 unicast** command to enter IPv6 unicast address family mode.

Example:

```
switch(config-router)# address-family
ipv6 unicast
switch(config-router-af)#
```

Step 4 Use the [**no**] **table-map** *map-name* command to configure the policy for filtering or modifying OSPFv3 routes before sending them to the RIB.

Example:

```
switch(config-router-af)# table-map foo
```

You can enter up to 63 alphanumeric characters for the map name.

Step 5 Use the **exit** command to exit router address-family configuration mode.

Example:

```
switch(config-router-af)# exit
switch(config-router)#
```

Step 6 Use the **exit** command to exit router configuration mode.

Example:

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

Step 7 Use the **route-map** *map-name* [**permit** | **deny**] [*seq*] command to create a route map or enters route-map configuration mode for an existing route map.

Example:

```
switch(config)# route-map foo permit 10
switch(config-route-map)#
```

Use *seq* to order the entries in a route map.

Note
The **permit** option enables you to set the distance. If you use the **deny** option, the default distance is applied.

Step 8 Use the **match route-type** *route-type* command to match against route type.

Example:

```
switch(config-route-map)# match
route-type external
```

These are the route types:

- external—The external route (BGP, EIGRP, and OSPF type 1 or 2)
- inter-area—The OSPF inter-area route
- internal—The internal route (including the OSPF intra- or inter-area)
- intra-area—The OSPF intra-area route
- nssa-external—The NSSA external route (OSPF type 1 or 2)
- type-1—The OSPF external type 1 route
- type-2—The OSPF external type 2 route

Step 9 Use the **match ip route-source prefix-list** *name* command to match the IPv6 route source address or router ID of a route to one or more IP prefix lists.

Example:

```
switch(config-route-map)# match ip
route-source prefix-list p1
```

Use the **ip prefix-list** command to create the prefix list.

Note
For OSPFv3, the router ID is 4 bytes.

Step 10 Use the **match ipv6 address prefix-list** *name* command to match against one or more IPv6 prefix lists.

Example:

```
switch(config-route-map)# match ipv6
address prefix-list p1
```

Use the **ip prefix-list** command to create the prefix list.

Step 11 Use the **set distance** *value* command to set the administrative distance of routes for OSPFv3.

Example:

```
switch(config-route-map)# set distance
150
```

The range is from 1 to 255.

Step 12 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config-route-map)# copy
running-config startup-config
```

Example

This example shows how to configure the OSPFv3 administrative distance for inter-area routes to 150, for external routes to 200, and for all prefixes in prefix list p1 to 190:

```
switch# <userinput>configure terminal</userinput>
switch(config)# <userinput>router ospfv3 201</userinput>
switch(config-router)# <userinput>address-family ipv6 unicast</userinput>
switch(config-router-af)# <userinput>table-map foo</userinput>
switch(config-router)# <userinput>exit</userinput>
switch(config)# <userinput>exit</userinput>
switch(config)# <userinput>route-map foo permit 10</userinput>
switch(config-route-map)# <userinput>match route-type inter-area</userinput>
switch(config-route-map)# <userinput>set distance 150</userinput>
switch(config)# <userinput>route-map foo permit 20</userinput>
switch(config-route-map)# <userinput>match route-type external</userinput>
switch(config-route-map)# <userinput>set distance 200</userinput>
switch(config)# <userinput>route-map foo permit 30</userinput>
switch(config-route-map)# <userinput>match ip route-source prefix-list p1</userinput>
switch(config-route-map)# <userinput>match ipv6 address prefix-list p1</userinput>
switch(config-route-map)# <userinput>set distance 190</userinput>
switch(config-route-map)# <userinput>copy running-config startup-config</userinput>
```

Modify the default timers

OSPFv3 includes a number of timers that control the behavior of protocol messages and shortest path first (SPF) calculations. OSPFv3 includes the following optional timer parameters:

- LSA arrival time—Sets the minimum interval allowed between LSAs arriving from a neighbor. LSAs that arrive faster than this time are dropped.
- Pacing LSAs—Sets the interval at which LSAs are collected into a group and refreshed, checksummed, or aged. This timer controls how frequently LSA updates occur and optimizes how many are sent in an LSA update message (see the Flooding and LSA Group Pacing section).

- Throttle LSAs—Sets rate limits for generating LSAs. This timer controls how frequently LSAs are generated after a topology change occurs.
- Throttle SPF calculation—Controls how frequently the SPF calculation is run.

At the interface level, you can also control the following timers:

- Retransmit interval—Sets the estimated time between successive LSAs.
- Transmit delay—Sets the estimated time to transmit an LSA to a neighbor.

See the Configuring Networks in OSPFv3 section for information on the hello interval and dead timer.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the **router ospfv3***instance-tag* command to create a new OSPFv3 instance with the configured instance tag.

Example:

```
switch(config)# router ospfv3 201
 switch(config-router)#
```

Step 3 Use the **timers lsa-arrival***msec* command to set the LSA arrival time in milliseconds.

Example:

```
switch(config-router)# timers
 lsa-arrival 2000
```

The range is from 10 to 600000. The default is 1000 milliseconds.

Step 4 Use the **timers lsa-group-pacing***seconds* command to set the interval in seconds for grouping LSAs.

Example:

```
switch(config-router)# timers
 lsa-group-pacing 200
```

The range is from 1 to 1800. The default is 10 seconds.

Step 5 Use the **timers throttle lsa***start-time hold-interval max-time* command to set the rate limit in milliseconds for generating LSAs.

Example:

```
switch(config-router)# timers
 throttle lsa network 350 5000 6000
```

You can configure theses timers:

- *start-time*—The range is from 0 to 5000 milliseconds. The default value is 0 milliseconds.
- *hold-interval*—The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds.
- *max-time*—The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds.

Step 6 Use the **address-family ipv6 unicast** command to enter IPv6 unicast address family mode.

Example:

```
switch(config-router)# address-family
 ipv6 unicast
 switch(config-router-af)#
```

Step 7 Use the **timers throttle spf***delay-time hold-time max-time* command to set the SPF best path schedule.

Example:

```
switch(config-router-af)# timers throttle
 spf 3000 2000
```

Sets the SPF best path schedule in seconds between SPF best path calculations with these timers:

- *delay-time*—The range is from 1 to 600,000 milliseconds. The default value is 200 milliseconds.
- *hold-time*—The range is from 1 to 600,000 milliseconds. The default value is 1000 milliseconds.
- *max-wait* —The range is from 1 to 600,000 milliseconds. The default value is 5000 milliseconds.

Step 8 Use the **interface***type slot/port* command to enter interface configuration mode.

Example:

```
switch(config)# interface ethernet 1/2
 switch(config-if)#
```

Step 9 Use the **ospfv3 retransmit-interval***seconds* command to set the estimated time in seconds between LSAs transmitted from this interface.

Example:

```
switch(config-if)# ospfv3
 retransmit-interval 30
```

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

Step 10 Use the **ospfv3 transmit-delay***seconds* command to set the estimated time in seconds to transmit an LSA to a neighbor.

Example:

```
switch(config-if)# ospfv3
 transmit-delay 600
```

The range is from 1 to 450. The default is 1.

Step 11 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch(config)# copy running-config startup-config
```

Example

This example shows how to control LSA flooding with the lsa-group-pacing option:

```
switch# <userinput>configure terminal</userinput>
switch(config)# <userinput>router ospfv3 201</userinput>
```

```
switch(config-router)# <userinput>timers lsa-group-pacing 300</userinput>
switch(config-router)# <userinput>copy running-config startup-config</userinput>
```

Configure graceful restart

Graceful restart is enabled by default. You can configure the following optional parameters for graceful restart in an OSPFv3 instance:

- Grace period—Configures how long neighbors should wait after a graceful restart has started before tearing down adjacencies.
- Helper mode disabled—Disables helper mode on the local OSPFv3 instance. OSPFv3 does not participate in the graceful restart of a neighbor.
- Planned graceful restart only—Configures OSPFv3 to support graceful restart only in the event of a planned restart.

Before you begin

You must enable OSPFv3 (see the Enabling OSPFv3 section).

- Ensure that all neighbors are configured for graceful restart with matching optional parameters set.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the **router ospfv3***instance-tag* command to create a new OSPFv3 instance with the configured instance tag.

Example:

```
switch(config)# router ospfv3 201
switch(config-router)#
```

Step 3 Use the **graceful-restart** command to enable a graceful restart. A graceful restart is enabled by default.

Example:

```
switch(config-router)# graceful-restart
```

Step 4 Use the **graceful-restart grace-period***seconds* command to set the grace period, in seconds.

Example:

```
switch(config-router)# graceful-restart
grace-period 120
```

The range is from 5 to 1800 seconds. The default is 60 seconds.

Step 5 Use the **graceful-restart helper-disable** command to disable helper mode.

Example:

```
switch(config-router)# graceful-restart
helper-disable
```

It's enabled by default.

Step 6 Use the **graceful-restart planned-only** command to configure graceful restart for planned restarts only.

Example:

```
switch(config-router)# graceful-restart
planned-only
```

Step 7 (Optional) Use the **show ipv6 ospfv3***instance-tag* command to see OPSFv3 information.

Example:

```
switch(config-router)# show ipv6 ospfv3
201
```

Step 8 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch(config)# copy running-config startup-config
```

Example

This example shows how to enable a graceful restart if it has been disabled and set the grace period to 120 seconds:

```
switch# <userinput>configure terminal</userinput>
switch(config)# <userinput>router ospfv3 201</userinput>
switch(config-router)# <userinput>graceful restart</userinput>
switch(config-router)# <userinput>graceful-restart grace-period 120</userinput>
switch(config-router)# <userinput>copy running-config startup-config</userinput>
```

Restart an OSPFv3 instance

You can restart an OSPv3 instance. This action clears all neighbors for the instance.

To restart an OSPFv3 instance and remove all associated neighbors, use the following command:

Procedure

Use the **restart ospfv3** *instance-tag* command to restart the OSPFv3 instance and removes all neighbors.

Example:

```
switch(config)# restart ospfv3 201
```

Configure OSPFv3 with virtualization

You can configure multiple OSPFv3 instances. You can also create multiple VRFs within the virtual device context (VDC) and use the same or multiple OSPFv3 instances in each VRF. You assign an OSPFv3 interface to a VRF.

Note Configure all other parameters for an interface after you configure the VRF for an interface. Configuring a VRF for an interface deletes all the configuration for that interface.

Before you begin

You must enable OSPFv3 (see the Enabling OSPFv3 section).

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the **vrf context***vrf-name* command to create a new VRF and enters VRF configuration mode.

Example:

```
switch(config)# vrf context
RemoteOfficeVRF
switch(config-vrf)#
```

Step 3 Use the **router ospfv3***instance-tag* command to create a new OSPFv3 instance with the configured instance tag.

Example:

```
switch(config)# router ospfv3 201
switch(config-router)#
```

Step 4 Use the **vrf***vrf-name* command to enter router VRF configuration mode.

Example:

```
switch(config-router)# vrf
RemoteOfficeVRF
switch(config-router-vrf)#
```

Step 5 (Optional) Use the **maximum-paths***paths* command to configure the maximum number of equal OSPFv3 paths to a destination in the route table for this VRF.

Example:

```
switch(config-router-vrf)# maximum-paths
4
```

Use this command for load balancing.

Step 6 Use the **interface***type slot/port* command to enter interface configuration mode.

Example:

```
switch(config)# interface ethernet 1/2
switch(config-if)#
```

Step 7 Use the **vrf member***vrf-name* command to add this interface to a VRF.

Example:

```
switch(config-if)# vrf member
RemoteOfficeVRF
```

Step 8 Use the **ipv6 address***ipv6-prefix/length* command to configure an IP address for this interface.

Example:

```
switch(config-if)# ipv6 address
2001:0DB8::1/48
```

You must do this step after you assign this interface to a VRF.

Step 9 Use the **ipv6 ospfv3***instance-tag***area***area-id* command to assign this interface to the OSPFv3 instance and area configured.

Example:

```
switch(config-if)# ipv6 ospfv3 201
area 0
```

Step 10 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch(config)# copy running-config startup-config
```

Example

This example shows how to create a VRF and add an interface to the VRF:

```
switch# <userinput>configure terminal</userinput>
switch(config)# <userinput>vrf context NewVRF</userinput>
switch(config-vrf)# <userinput>exit</userinput>
switch(config)# <userinput>router ospfv3 201</userinput>
switch(config-router)# <userinput>exit</userinput>
switch(config)# <userinput>interface ethernet 1/2</userinput>
switch(config-if)# <userinput>vrf member NewVRF</userinput>
switch(config-if)# <userinput>ipv6 address 2001:0DB8::1/48</userinput>
switch(config-if)# <userinput>ipv6 ospfv3 201 area 0</userinput>
switch(config-if)# <userinput>copy running-config startup-config</userinput>
```

Configure Encryption and Authentication

Beginning with Cisco Nexus Release 10.2(1), you can encrypt and authenticate OSPFv3 messages using ESP encapsulation. OSPFv3 depends on IPSec for secure connection. IPSec supports two encapsulation types: Authentication Header (AH) and Encapsulating Security Payload (ESP). RFC4552 'Authentication/Confidentiality for OSPFv3' covers both aspects. ESP configuration provides both encryption and authentication for OSPFv3 messages.

Beginning with Cisco Nexus Release 10.4(1)F, the encryption and authentication algorithms and keys can be configured using the keychain option.

Procedure

Step 1 The following are the limitations:

- Only IPSec transport mode is supported and tunnel mode is not supported.
- AH and ESP configurations together are not allowed on an interface. Though two different interfaces can have AH and ESP.
- Non-disruptive rekeying as defined in section 10 of RFC 4552 is not supported.
- The following Encryption Algorithms will be supported under ESP:
  - AES-CBC (128 bit)
  - AES 192 bit and AES 256 bit will not be supported in this release.
  - 3DES-CBC
  - NULL
- The following Authentications will be supported under ESP:
  - SHA-1
  - NULL
- Both Encryption and Authentication algorithms cannot be configured NULL in one ESP CLI.
- An interface which is part of multiple areas uses the same ESP parameters as the parent.
- On SPI conflict during configuration, error will be thrown to user and configuration will not be saved. So, while changing the ESP configuration the user must use different SPI for a new configuration.
- Max 128 SA/SPI values can be configured per OSPFv3 process.

Step 2 You can configure ESP at the following levels:

- Router
- Area
- Interface
- Virtual Links

Configuring OSPFv3 encryption at router level

You can configure OSPFv3 ESP to encrypt and authenticate OSPFv3 packets at the router level using the following commands.

For information on how to configure a keychain, see **Configuring Keychain Management** of *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

Before you begin

Enable OSPFv3 feature.

Enable authentication package.

Procedure

Step 1 Enter the global configuration mode:

switch# **configure terminal**

Step 2 Enable OSPFv3:

switch(config)# **feature ospfv3**

Step 3 Enable authentication package:

switch(config)# **feature imp**

Step 4 Create a new OSPFv3 instance with the configured instance tag:

switch(config)# **router ospfv3** *instance-tag*

Step 5 Enable IPSec ESP encryption:

switch(config-router)# **encryption ipsec spi** *spi_id* **esp** [*encrypt_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *enc_keychain_name* | **null**] **authentication** [*auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**]

You can specify the security policy index through *spi_id* and define the encryption algorithm through *encrypt_algorithm* which can be 3DES, AES 128 or null. Numbers 0, 3, and 7 specify the format of the *key*. You can define the authentication algorithm through *auth_algorithm* which can be SHA-1 or NULL.

You can also configure keys and algorithms using the **key-chain** option.

Step 6 (Optional) Display OSPFv3 information:

switch(config)# **show running-config ospfv3**

Configure OSPFv3 encryption at area level

You can configure OSPFv3 ESP to encrypt and authenticate OSPFv3 packets at the area level using the following commands.

For information on how to configure a keychain, see **Configuring Keychain Management** of *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

Before you begin

Enable OSPFv3 feature.

Enable authentication package.

Procedure

Step 1 Enter the global configuration mode:

switch# **configure terminal**

Step 2 Enable OSPFv3:

switch(config)# **feature ospfv3**

Step 3 Enable the authentication package:

switch(config)# **feature imp**

Step 4 Create a new OSPFv3 instance with the configured instance tag:

switch(config)# **router ospfv3** *instance-tag*

Step 5 Enable IPSec ESP Encryption:

switch(config-router)#**area** *area-num* **encryption ipsec spi** *spi_val* **esp** *encrypt_algorithm* [**0** | **3** | **7***key* | **key-chain** *enc_keychain_name* | **null**] **authentication** *auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**]

You can specify the security policy index through *spi_id* and define the encryption algorithm through *encrypt_algorithm* which can be 3DES, AES 128 or null. Numbers 0, 3, 6 and 7 specify the format of the *key*. You can define the authentication algorithm through *auth_algorithm* which can be SHA-1 or NULL or key-chain.

You can also configure keys and algorithms using the **key-chain** option.

Step 6 (Optional) Display OSPFv3 information:

switch(config)# **show running-config ospfv3**

Configure OSPFv3 encryption at interface level

You can configure OSPFv3 ESP to encrypt and authenticate OSPFv3 packets at the interface level using the following commands.

For information on how to configure a keychain, see **Configuring Keychain Management** of *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

Before you begin

You must enable OSPFv3.

- Enable authentication package.

Procedure

Step 1 Enter the global configuration mode:

switch# **configure terminal**

Step 2 Enable OSPFv3:

switch(config)# **feature ospfv3**

Step 3 Enables the authentication mode:

switch(config)# **feature imp**

Step 4 Enters the interface configuration mode:

switch(config)# **interface ethernet** *interface*

Step 5 Specify the OSPFv3 instance and area for the interface:

switch(config-if)# **ipv6 router ospfv3** *instance-tag* **area** *area-id*

Step 6 Enable IPSec ESP Encryption:

switch(config-if)# **ospfv3 encryption ipsec spi** *spi_id* **esp** *encrypt_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *enc_keychain_name* | **null**] **authentication** *auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**]

You can specify the security policy index through *spi_id* and define the encryption algorithm through *encrypt_algorithm* which can be 3DES, AES 128 or null. Numbers 0, 3 and 7 specify the format of the *key*. You can define the authentication algorithm through *auth_algorithm* which can be SHA-1 or NULL.

You can also configure keys and algorithms using the key-chain option.

Step 7 (Optional) Display the running configuration on the interface:

switch(config-if)#**show run interface** *interface*

Example

The following example shows how to enable security for Ethernet interface 3/2:

```
switch# configure terminal
switch(config)# feature ospfv3
switch(config)# feature imp
switch(config)# interface ethernet 3/2
switch(config-if)# ipv6 router ospfv3 1 area 0.0.0.0
switch(config-if)# ospfv3 encryption ipsec spi 444
esp  Specify encryption parameters
switch(config-if)# ospfv3 encryption ipsec spi 444 esp
3des  Use the triple DES algorithim
aes   Use the AES algorithim
<!--NXOS1-307-->key-chain  Encryption password key-chain
null  Use NULL authentication
switch(config-if)# ospfv3 encryption ipsec spi 444 esp aes
128  Use the 128-bit AES algorithim
switch(config-if)# ospfv3 encryption ipsec spi 444 esp aes 128
```

```
0     Specifies an UNENCRYPTED encryption key will follow
3     Specifies an 3DES ENCRYPTED encryption key will follow
7     Specifies a Cisco type 7  ENCRYPTED encryption key will follow
WORD  The UNENCRYPTED (cleartext) encryption key
switch(config-if)# ospfv3 encryption ipsec spi 444 esp aes 128
12345678123456781234567812345678 authentication null
switch(config-if)# sh ospfv3 interface
Ethernet3/2 is up, line protocol is up
IPv6 address 1:1:1:1::2/64
Process ID 1 VRF default, Instance ID 0, area 0.0.0.0
Enabled by interface configuration
State DOWN, Network type BROADCAST, cost 40
ESP Encryption AES, Authentication NULL, SPI 444, ConnId 444
switch(config-if)#
```

Configure OSPFv3 encryption for virtual links

You can configure OSPFv3 ESP to encrypt and authenticate OSPFv3 packets for virtual links using the following commands.

For information on how to configure a keychain, see **Configuring Keychain Management** of *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

Before you begin

Enable OSPFv3 feature.

Enable authentication package.

Procedure

Step 1 Enter the global configuration mode:

switch# **configure terminal**

Step 2 Enable OSPFv3:

switch(config)# **feature ospfv3**

Step 3 Enable the authentication package:

switch(config)# **feature imp**

Step 4 Create a new OSPFv3 instance with the configured instance tag:

switch(config)#**router ospfv3** *instance-tag*

Step 5 Creates one end of a virtual link to a remote router. You must create the virtual link on that remote router to complete the link.

switch(config-router)# **area** *area-id* **virtual-link** *router-id*

Step 6 Enable IPSec ESP Encryption:

switch(config-router-vlink)# **encryption ipsec spi** *spi_id* **esp** *encrypt_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *enc_keychain_name* | **null**] **authentication** *auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**]

You can specify the security policy index through *spi_id* and define the encryption algorithm through *encrypt_algorithm* which can be 3DES, AES 128 or null. Numbers 0, 3 and 7 specify the format of the *key*. You can define the authentication algorithm through *auth_algorithm* which can be SHA-1 or NULL.

You can also configure keys and algorithms using the **key-chain** option.

Step 7 (Optional) Display OSPFv3 information:

switch(config)# **show running-config ospfv3**

Configuration Example

The following example shows how to encrypt Virtual links:

```
switch(config)# feature ospfv3
switch(config)# feature imp
switch(config-if)# router ospfv3 1
switch(config-router)# area 0.0.0.1 virtual-link 3.3.3.3
switch(config-router-vlink)# encryption ipsec spi ?
<256-4294967295> SPI Value
switch(config-router-vlink)# encryption ipsec spi 256 esp ?
3des Use the triple DES algorithim
aes Use the AES algorithim
key-chain  Encryption password key-chain
null Use NULL authentication
switch(config-router-vlink)# encryption ipsec spi 256 esp aes 128
123456789A123456789B123456789C12 authentication ?
null Use NULL authentication
sha1 Use the SHA1 algorithim
switch(config-router-vlink)# encryption ipsec spi 256 esp aes 128
123456789A123456789B123456789C12 authentication null
```

Note To permit multiple OSPFv3 neighbors to have IPsec ESP, the following policy-map has to be applied for a control-plane:

```
ipv6 access-list copp-acl-ipsec
10 permit ahp any any
20 permit esp any any

class-map type control-plane match-any copp-class-critical-customized-copp
match access-group name copp-acl-ipsec
policy-map type control-plane customized-copp
class copp-class-critical-customized-copp
police cir 36000 kbps bc 1280000 bytes conform transmit violate drop
control-plane
service-policy input customized-copp
```

Configure OSPFv3 authentication at router level

You can configure OSPFv3 ESP to authenticate OSPFv3 packets at the router level using the following commands.

For information on how to configure a keychain, see **Configuring Keychain Management** of *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

Before you begin

Ensure that you have enabled OSPFv3, for more information see Enabling OSPFv3 section.

Procedure

Step 1 **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Enters global configuration mode.

Step 2 **feature ospfv3**

Example:

```
switch(config)# feature ospfv3
```

Enables OSPFv3.

Step 3 **feature imp**

Example:

```
switch(config)# feature imp
```

Enables authentication mode.

Step 4 **router ospfv3 instance-tag**

Example:

```
switch(config)# router ospfv3 100
switch(config-router)#
```

Creates a new OSPFv3 instance with the configured instance tag.

Step 5 [**no**] **authentication** {**ipsec spi** *spi_id* [*auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**]

Example:

For authentication algorithm and key option:

```
switch(config-router)# authentication ipsec spi 475 md5 11111111111111112222222222222222
```

For keychain option:

```
switch(config-router)# authentication ipsec spi 333 key-chain test1
```

Configures OSPFv3 IPSec authentication at the process (or VRF) level.

The spi argument specifies the security parameter index (SPI). The range is from 256 to 4294967295.

The auth argument specifies the type of authentication. The supported values are md5 or sha1.

0 configures the password in cleartext. 3 configures the pass key as 3DES encrypted. 7 configures the key as Cisco type 7 encrypted.

If the cleartext option (0) is used, the key argument must be 32 characters long for md5 or 40 characters long for sha1.

Beginning with Cisco NX-OS Release 10.4(1)F, the **key-chain** option is provided to configure key and algorithm.

Use the **no** form of this command to disable the OSPFv3 IPSec authentication.

Step 6 (Optional) **show running-config ospfv3**

Example:

```
switch(config)# show running-config ospfv3
```

Displays the OSPFv3 authentication configuration information.

Step 7 (Optional) **copy running-config startup-config**

Example:

```
switch(config)# copy running-config
startup-config
```

Saves this configuration change.

Configure OSPFv3 authentication at area level

You can configure OSPFv3 ESP to authenticate OSPFv3 packets at the area level using the following commands.

For information on how to configure a keychain, see **Configuring Keychain Management** of *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

Before you begin

Ensure that you have enabled OSPFv3, for more information see Enabling OSPFv3 section.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Use the **feature ospfv3** command to enable OSPFv3.

Example:

```
switch(config)# feature ospfv3
```

Step 3 Use the **feature imp** command to enable authentication mode.

Example:

```
switch(config)# feature imp
```

Step 4 Use the **router ospfv3 instance-tag** command to create a new OSPFv3 instance with the configured instance tag.

Example:

```
switch(config)# router ospfv3 100
                        switch(config-router)#
```

Step 5 Use the [**no**] **area** *area-id-ip* **authentication** {**ipsec spi** *spi_id*[*auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**] command to configure OSPFv3 IPSec authentication at the area level.

Example:

For authentication algorithm and key option:

```
switch(config-router)# area 0 authentication ipsec spi 475 md5 11111111111111112222222222222222
```

For keychain option:

```
switch(config-router)# area 0 authentication ipsec spi 333 key-chain test1
```

The spi argument specifies the security parameter index (SPI). The range is from 256 to 4294967295.

The auth argument specifies the type of authentication. The supported values are MD5 or SHA-1.

0 configures the password in cleartext. 3 configures the pass key as 3DES encrypted. 7 configures the key as Cisco Type-7 encrypted.

If the cleartext option (0) is used, the key argument must be 32 characters long for MD5 or 40 characters long for SHA-1.

Beginning with Cisco NX-OS Release 10.4(1)F, the **key-chain** option is provided to configure key and algorithm.

Use the **no** form of this command to disable the OSPFv3 IPSec authentication.

Step 6 Use the **show running-config ospfv3** command to see the OSPFv3 authentication configuration information.

Example:

```
switch(config)# show running-config ospfv3
```

Step 7 Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config)# copy running-config startup-config
```

Configure OSPFv3 authentication at interface level

You can configure OSPFv3 ESP to authenticate OSPFv3 packets at interval level using the following commands.

For information on how to configure a keychain, see **Configuring Keychain Management** of *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

Before you begin

Ensure that you have enabled OSPFv3, for more information see Enabling OSPFv3 section.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Use the **interface***interface-type slot/port* command to enter interface configuration mode.

Example:

```
switch(config)# interface ethernet 1/1
                        switch(config-if)#
```

Step 3 Use the [**no**] **ospfv3 authentication** {**disable** | **ipsec spi** *spi_id* {**md5** *akey* | **sha1** *akey* | **key-chain** *keychain_ah*}} command to configure OSPFv3 IPSec authentication for the specified interface.

Example:

For authentication algorithm and key option:

```
switch(config-if)# ospfv3 authentication ipsec spi 475 md5 11111111111111112222222222222222
```

For keychain option:

```
switch(config-if)# ospfv3 authentication ipsec spi 333 key-chain test1
```

The spi argument specifies the security parameter index (SPI). The range is from 256 to 4294967295.

The auth argument specifies the type of authentication. The supported values are MD5 or SHA-1.

0 configures the password in cleartext. 3 configures the pass key as 3DES encrypted. 7 configures the key as Cisco Type-7 encrypted.

If the cleartext option (0) is used, the key argument must be 32 characters long for MD5 or 40 characters long for SHA-1.

Beginning with Cisco NX-OS Release 10.4(1)F, the **key-chain** option is provided to configure key and algorithm.

Use the **no** form of this command to disable the OSPFv3 IPSec authentication.

Step 4 Use the **show running-config ospfv3** command to see the OSPFv3 authentication configuration information.

Example:

```
switch(config)# show running-config ospfv3
```

Step 5 Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config)# copy running-config startup-config
```

Configure OSPFv3 authentication at virtual links level

You can configure OSPFv3 ESP to authenticate OSPFv3 packets at the virtual link level using the following commands.

For information on how to configure a keychain, see **Configuring Keychain Management** of *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

Before you begin

Ensure that you have enabled OSPFv3, for more information see Enabling OSPFv3 section.

Procedure

Step 1 Use the **configure terminal**

Example:

```
switch# configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Use the **feature ospfv3** command to enable OSPFv3.

Example:

```
switch(config)# feature ospfv3
```

Step 3 Use the **feature imp** command to enable authentication mode.

Example:

```
switch(config)# feature imp
```

Step 4 Use the **router ospfv3 instance-tag** command to create a new OSPFv3 instance with the configured instance tag.

Example:

```
switch(config)# router ospfv3 100
                        switch(config-router)#
```

Step 5 Use the **area** *area-id* **virtual-link** *router-id* command to create one end of a virtual link to a remote router.

Example:

```
switch(config-router)# area 0.0.0.10 virtual-link 2001:0DB8::1
                        switch(config-router-vlink)#
```

You must create the virtual link on that remote router to complete the link.

Step 6 Use the [**no**] **authentication** {**ipsec spi** *spi_id* [*auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**] command to configure OSPFv3 IPSec authentication at the virtual link level.

Example:

For authentication algorithm and key option:

```
switch(config-router-vlink)# authentication ipsec spi 475 md5 11111111111111122222222222222222
```

For keychain option:

```
switch(config-router-vlink)# authentication ipsec spi 333 key-chain test1
```

The spi argument specifies the security parameter index (SPI). The range is from 256 to 4294967295.

The auth argument specifies the type of authentication. The supported values are MD5 or SHA-1.

0 configures the password in cleartext. 3 configures the pass key as 3DES encrypted. 7 configures the key as Cisco Type-7 encrypted.

If the cleartext option (0) is used, the key argument must be 32 characters long for MD5 or 40 characters long for SHA-1.

Beginning with Cisco NX-OS Release 10.4(1)F, the **key-chain** option is provided to configure key and algorithm.

Use the **no** form of this command to disable the OSPFv3 IPSec authentication.

Step 7 (Optional) Use the **show running-config ospfv3** command to see the OSPFv3 authentication configuration information.

Example:

```
switch(config)# show running-config ospfv3
```

Step 8 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config)# copy running-config startup-config
```

Verify the OSPFv3 configuration

To display the OSPFv3 configuration, perform one of the following tasks:

Command	Purpose
show ipv6 ospfv3 [instance-tag] [vrf vrf-name]	Displays information about one or more OSPFv3 routing instances. The output includes the following area-level counts: • Interfaces in this area—A count of all interfaces added to this area (configured interfaces). • Active interfaces—A count of all interfaces considered to be in router link states and SPF (UP interfaces). • Passive interfaces—A count of all interfaces considered to be OSPF passive (no adjacencies will be formed). • Loopback interfaces—A count of all local loopback interfaces.
show ipv6 ospfv3 border-routers	Displays the internal OSPF routing table entries to an ABR and ASBR.
show ipv6 ospfv3 database	Displays lists of information related to the OSPFv3 database for a specific router.
show ipv6 ospfv3 interface *type number* [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays the OSPFv3 interface information.
show ipv6 ospfv3 neighbors	Displays the neighbor information. Use the **clear ospfv3 neighbors** command to remove adjacency with all neighbors.
show ipv6 ospfv3 request-list	Displays a list of LSAs requested by a router.
show ipv6 ospfv3 retransmission-list	Displays a list of LSAs waiting to be retransmitted.

Command	Purpose
show ipv6 ospfv3 summary-address	Displays a list of all summary address redistribution information configured under an OSPFv3 instance.
show ospfv3 process	Displays the OSPFv3 authentication configuration at the process level.
show ospfv3 interface *interface-type slot/port*	Displays the OSPFv3 authentication configuration at the interface level.
show running-configuration ospfv3	Displays the current running OSPFv3 configuration.

Monitor OSPFv3

To display OSPFv3 statistics, use the following commands:

Command	Purpose
show ipv6 ospfv3 memory	Displays the OSPFv3 memory usage statistics.
show ipv6 ospfv3 policy statistics area *area-id* **filter-list** {**in** | **out**} [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays the OSPFv3 route policy statistics for an area.
show ipv6 ospfv3 policy statistics redistribute {**bgp** *id* | **direct** | **isis** *id* | **rip** *id* | **static vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays the OSPFv3 route policy statistics.
show ipv6 ospfv3 statistics [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays the OSPFv3 event counters.
show ipv6 ospfv3 traffic *interface-type number* [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays the OSPFv3 packet counters.

Configuration examples for OSPFv3

This example shows how to configure OSPFv3:

```
This example shows how to configure OSPFv3:
        feature ospfv3
        router ospfv3 201
        router-id 290.0.2.1

        interface ethernet 1/2
        ipv6 address 2001:0DB8::1/48
        ipv6 ospfv3 201 area 0.0.0.10
```

This example shows how to configure OSPFv3 encryption using **key-chain** option:

```
switch(config-if)# ospfv3 encryption ipsec spi 333 esp ?
        3des       Use the triple DES algorithim
        aes        Use the AES algorithim
        key-chain  Encryption password key-chain
        null       Use NULL authentication
```

```
        switch(config-if)# ospfv3 encryption ipsec spi 333 esp key-chain ?
        WORD  Encryption key-chain name (Max Size 63)
        switch(config-if)# ospfv3 encryption ipsec spi 333 esp key-chain test1 ?
        authentication  Specify authentication parameters
       switch(config-if)# ospfv3 encryption ipsec spi 333 esp key-chain test1 authentication
?
        key-chain  Authentication password key-chain
        null       Use NULL authentication
      switch(config-if)# ospfv3 encryption ipsec spi 333 esp key-chain test1 authentication
key-chain ?
        WORD Authentication key-chain name (Max Size 63)
      switch(config-if)# ospfv3 encryption ipsec spi 333 esp key-chain test1 authentication
key-chain test2 ?
        <CR>
        switch(config-router)# sh ospfv3
        Routing Process 2 with ID 20.20.10.2 VRF default
        Routing Process Instance Number 1
        Install discard route for summarized internal routes.
        ESP Encryption 3DES, Authentication SHA1, SPI 334, ConnId 334
        ESP keychains: Encr test_key_chain_01(ready), Auth test1(ready)
        Number of new LSAs originated : 3
        Number of new LSAs received : 0
```

Related topics

The following topics can give more information on OSPF:

- OSPFv2
- Configuring Route Policy Manager

Additional references

For additional information related to implementing OSPF, see the following sections:

MIBs

MIBs	MIBs Link
MIBs related to OSPFv3	To locate and download supported MIBs, go to the following URL: https://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/mib/quickreference/cisco-nexus-7000-series-and-9000-series-nx-os-mib-quick-reference.html#con_67262