CHAPTER 22

Configuring OSPF

This chapter describes how to configure the adaptive security appliance to route data, perform authentication, and redistribute routing information, using the Open Shortest Path First (OSPF) routing protocol.

The chapter includes the following sections:

Information About OSPF

OSPF is an interior gateway routing protocol that uses link states rather than distance vectors for path selection. OSPF propagates link-state advertisements rather than routing table updates. Because only LSAs are exchanged instead of the entire routing tables, OSPF networks converge more quickly than RIP networks.

OSPF uses a link-state algorithm to build and calculate the shortest path to all known destinations. Each router in an OSPF area contains an identical link-state database, which is a list of each of the router usable interfaces and reachable neighbors.

The advantages of OSPF over RIP include the following:

- OSPF link-state database updates are sent less frequently than RIP updates, and the link-state database is updated instantly rather than gradually as stale information is timed out.
- Routing decisions are based on cost, which is an indication of the overhead required to send packets across a certain interface. The adaptive security appliance calculates the cost of an interface based on link bandwidth rather than the number of hops to the destination. The cost can be configured to specify preferred paths.

The disadvantage of shortest path first algorithms is that they require a lot of CPU cycles and memory.

The adaptive security appliance can run two processes of OSPF protocol simultaneously, on different sets of interfaces. You might want to run two processes if you have interfaces that use the same IP addresses (NAT allows these interfaces to coexist, but OSPF does not allow overlapping addresses). Or you might want to run one process on the inside, and another on the outside, and redistribute a subset of routes between the two processes. Similarly, you might need to segregate private addresses from public addresses.

You can redistribute routes into an OSPF routing process from another OSPF routing process, a RIP routing process, or from static and connected routes configured on OSPF-enabled interfaces.

The adaptive security appliance supports the following OSPF features:

- Support of intra-area, interarea, and external (Type I and Type II) routes.
- Support of a virtual link.
- OSPF LSA flooding.
- Authentication to OSPF packets (both password and MD5 authentication).
- Support for configuring the adaptive security appliance as a designated router or a designated backup router. The adaptive security appliance also can be set up as an ABR.
- Support for stub areas and not-so-stubby-areas.

Area boundary router Type-3 LSA filtering.

OSPF supports MD5 and clear text neighbor authentication. Authentication should be used with all routing protocols when possible because route redistribution between OSPF and other protocols (like RIP) can potentially be used by attackers to subvert routing information.

If NAT is used, if OSPF is operating on public and private areas, and if address filtering is required, then you need to run two OSPF processes—one process for the public areas and one for the private areas.

A router that has interfaces in multiple areas is called an Area Border Router (ABR). A router that acts as a gateway to redistribute traffic between routers using OSPF and routers using other routing protocols is called an Autonomous System Boundary Router (ASBR).

An ABR uses LSAs to send information about available routes to other OSPF routers. Using ABR Type 3 LSA filtering, you can have separate private and public areas with the adaptive security appliance acting as an ABR. Type 3 LSAs (inter-area routes) can be filtered from one area to other. This lets you use NAT and OSPF together without advertising private networks.

Note Only Type 3 LSAs can be filtered. If you configure the adaptive security appliance as an ASBR in a private network, it will send Type 5 LSAs describing private networks, which will get flooded to the entire AS including public areas.

If NAT is employed but OSPF is only running in public areas, then routes to public networks can be redistributed inside the private network, either as default or Type 5 AS External LSAs. However, you need to configure static routes for the private networks protected by the adaptive security appliance. Also, you should not mix public and private networks on the same adaptive security appliance interface.

You can have two OSPF routing processes, one RIP routing process, and one EIGRP routing process running on the adaptive security appliance at the same time.

Licensing Requirements for OSPF

Model	License Requirement
All models	Base License.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single context mode.

Firewall Mode Guidelines

Supported in routed mode only. Transparent mode is not supported.

IPv6 Guidelines

Does not support IPv6.

Configuring OSPF

This section describes how to enable an OSPF process on your system.

After you enable OSPF, you need to define a route map. For more information, see the "Defining Route Maps" section on page 21-1. Then you generate a default route. For more information, see the "Configuring Static and Default Routes" section on page 20-2.

After you have defined a route map for the OSPF process, you can customize the OSPF process to suit your particular needs, To learn how to customize the OSPF process on your system, see the "Customizing OSPF" section on page 22-4.

To enable OSPF, you need to create an OSPF routing process, specify the range of IP addresses associated with the routing process, then assign area IDs associated with that range of IP addresses.

You can enable up to two OSPF process instances. Each OSPF process has its own associated areas and networks.

To enable OSPF, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `hostname(config)# router ospf 2`	Creates an OSPF routing process and enters router configuration mode for this OSPF process. The *process_id* is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes. If there is only one OSPF process enabled on the adaptive security appliance, then that process is selected by default. You cannot change the OSPF process ID when editing an existing area.

Step 2	**network** *ip_address mask* **area** *area_id*
	
	Example:
	`hostname(config)# router ospf 2`
	`hostname(config-router)# network 10.0.0.0 255.0.0.0 area 0`
		Defines the IP addresses on which OSPF runs and the area ID for that interface.
		
		When adding a new area, enter the area ID. You can specify the area ID as either a decimal number or an IP address. Valid decimal values range from 0-4294967295. You cannot change the area ID when editing an existing area.

Customizing OSPF

This section explains how to customize the OSPF process and includes the following topics:

Redistributing Routes Into OSPF

The adaptive security appliance can control the redistribution of routes between OSPF routing processes.

Note If you want to redistribute a route by defining which of the routes from the specified routing protocol are allowed to be redistributed into the target routing process, you must first generate a default route. See the "Configuring Static and Default Routes" section on page 20-2 and then define a route map according to the "Defining a Route Map" section on page 21-4.

To redistribute static, connected, RIP, or OSPF routes into an OSPF process, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `hostname(config)# router ospf 2`	Creates an OSPF routing process and enters router configuration mode for the OSPF process that you want to redistribute. The *process_id* is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.
Step 2	Do one of the following to redistribute the selected route type into the OSPF routing process:	
	redistribute connected **[**[**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*] **Example:** `hostname(config)# redistribute connected 5 type-1 route-map-practice`	Redistributes connected routes into the OSPF routing process.
	redistribute static [**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name* **Example:** `hostname(config)# redistribute static 5 type-1 route-map-practice`	Redistributes static routes into the OSPF routing process.

	redistribute ospf ***pid*** **[match {internal | external [1 | 2] | nssa-external [1 | 2]}]** [**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*]
	
	Example:
	`hostname(config)# route-map 1-to-2 permit`
	`hostname(config-route-map)# match metric 1`
	`hostname(config-route-map)# set metric 5`
	`hostname(config-route-map)# set metric-type type-1`
	`hostname(config-route-map)# router ospf 2`
	`hostname(config-router)# redistribute ospf 1 route-map 1-to-2`
		Allows you to redistribute routes from an OSPF routing process into another OSPF routing process.
		
		You can either use the **match** options in this command to match and set route properties, or you can use a route map. The **subnets** option does not have equivalents in the **route-map** command. If you use both a route map and **match** options in the **redistribute** command, then they must match.
		
		The example shows route redistribution from OSPF process 1 into OSPF process 2 by matching routes with a metric equal to 1. The adaptive security appliance redistributes these routes as external LSAs with a metric of 5 and a metric type of Type 1.

Command	Purpose

redistribute rip [**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*]
 	
Example:
`hostname(config)# redistribute rip 5`
`hostname(config-route-map)# match metric 1`
`hostname(config-route-map)# set metric 5`
`hostname(config-route-map)# set metric-type type-1`
`hostname(config-router)# redistribute ospf 1 route-map 1-to-2`
	Allows you to redistribute routes from a RIP routing process into the OSPF routing process.

redistribute eigrp ***as-num*** [**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*]
 	
Example:
`hostname(config)# redistribute eigrp 2`
`hostname(config-route-map)# match metric 1`
`hostname(config-route-map)# set metric 5`
`hostname(config-route-map)# set metric-type type-1`
`hostname(config-router)# redistribute ospf 1 route-map 1-to-2`
	Allows you to redistribute routes from an EIGRP routing process into the OSPF routing process.

Configuring Route Summarization When Redistributing Routes into OSPF

When routes from other protocols are redistributed into OSPF, each route is advertised individually in an external LSA. However, you can configure the adaptive security appliance to advertise a single route for all the redistributed routes that are covered by a specified network address and mask. This configuration decreases the size of the OSPF link-state database.

Routes that match the specified IP Address mask pair can be suppressed. The Tag value can be used as a match value for controlling redistribution through route maps.

To configure the software advertisement on one summary route for all redistributed routes covered by a network address and mask, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `hostname(config)# router ospf 1`	Creates an OSPF routing process and enters router configuration mode for this OSPF process. The *process_id* is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2	**summary-address** *ip_address mask* [**not-advertise**] [**tag** *tag*]
	
	Example:
	`hostname(config)# router ospf 1`
	`hostname(config-router)# summary-address 10.1.0.0 255.255.0.0`
		Sets the summary address.
		
		In this example, the summary address 10.1.0.0 includes addresses 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. Only the 10.1.0.0 address is advertised in an external link-state advertisement.

Configuring Route Summarization Between OSPF Areas

Route summarization is the consolidation of advertised addresses. This feature causes a single summary route to be advertised to other areas by an area boundary router. In OSPF, an area boundary router advertises networks in one area into another area. If the network numbers in an area are assigned in a way such that they are contiguous, you can configure the area boundary router to advertise a summary route that covers all the individual networks within the area that fall into the specified range.

To define an address range for route summarization, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `hostname(config)# router ospf 1`	Creates an OSPF routing process and enters router configuration mode for this OSPF process. The *process_id* is an internally used identifier for this routing process. It can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2	**area** *area-id* **range** *ip-address mask* [**advertise** | **not-advertise**]
	
	Example:
	`hostname(config)# router ospf 1`
	`hostname(config-router)# area 17 range 12.1.0.0 255.255.0.0`
		Sets the address range.
		
		In this example, the address range is set between OSPF areas.

Configuring OSPF Interface Parameters

You can alter some interface-specific OSPF parameters as necessary.

Prerequisites

You are not required to alter any of these parameters, but the following interface parameters must be consistent across all routers in an attached network: **ospf hello-interval**, **ospf dead-interval**, and **ospf authentication-key**. Be sure that if you configure any of these parameters, the configurations for all routers on your network have compatible values.

To configure OSPF interface parameters, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `hostname(config)# router ospf 2`	Creates an OSPF routing process and enters router configuration mode for the OSPF process that you want to redistribute. The *process_id* is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2	**network** *ip_address mask* **area** *area_id*
	
	Example:
	`hostname(config)# router ospf 2`
	`hostname(config-router)# network 10.0.0.0 255.0.0.0 area 0`
		Defines the IP addresses on which OSPF runs and the area ID for that interface.

Step 3	`hostname(config)#` **interface** *interface_name* **Example:** `hostname(config)# interface my_interface`	Allows you to enter interface configuration mode.
Step 4	Do one of the following to configure optional OSPF interface parameters:	
	ospf authentication [**message-digest** | **null**] **Example:** `hostname(config-interface)# ospf authentication message-digest`	Specifies the authentication type for an interface.

Command	Purpose
ospf authentication-key *key* **Example:** `hostname(config-interface)# ospf authentication-key cisco`	Allows you to assign a password to be used by neighboring OSPF routers on a network segment that is using the OSPF simple password authentication. The *key* can be any continuous string of characters up to 8 bytes in length. The password created by this command is used as a key that is inserted directly into the OSPF header when the adaptive security appliance software originates routing protocol packets. A separate password can be assigned to each network on a per-interface basis. All neighboring routers on the same network must have the same password to be able to exchange OSPF information.
ospf cost *cost* **Example:** `hostname(config-interface)# ospf cost 20`	Allows you to explicitly specify the cost of sending a packet on an OSPF interface. The *cost* is an integer from 1 to 65535. In this example, the cost is set to 20.
ospf dead-interval *seconds* **Example:** `hostname(config-interface)# ospf dead-interval 40`	Allows you to set the number of seconds that a device must wait before it declares a neighbor OSPF router down because it has not received a hello packet. The value must be the same for all nodes on the network. In this example, the dead interval is set to 40.
ospf hello-interval *seconds* **Example:** `hostname(config-interface)# ospf hello-interval 10`	Allows you to specify the length of time between the hello packets that the adaptive security appliance sends on an OSPF interface. The value must be the same for all nodes on the network. In this example, the hello interval is set to 10.
ospf message-digest-key *key_id* **md5** *key* **Example:** `hostname(config-interface)# ospf message-digest-key 1 md5 cisco`	Enables OSPF MD5 authentication. The following values can be set: • *key_id*—An identifier in the range from 1 to 255. • *key*—Alphanumeric password of up to 16 bytes. Usually, one key per interface is used to generate authentication information when sending packets and to authenticate incoming packets. The same key identifier on the neighbor router must have the same key value. We recommend that you not keep more than one key per interface. Every time you add a new key, you should remove the old key to prevent the local system from continuing to communicate with a hostile system that knows the old key. Removing the old key also reduces overhead during rollover.
ospf priority *number_value* **Example:** `hostname(config-interface)# ospf priority 20`	Allows you to set the priority to help determine the OSPF designated router for a network. The *number_value* is between 0 to 255. In this example, the priority number value is set to 20.

Command	Purpose
ospf retransmit-interval *seconds* **Example:** `hostname(config-interface)# ospf retransmit-interval seconds`	Allows you to specify the number of seconds between LSA retransmissions for adjacencies belonging to an OSPF interface. The value for *seconds* must be greater than the expected round-trip delay between any two routers on the attached network. The range is from 1 to 65535 seconds. The default value is 5 seconds. In this example, the retransmit-interval value is set to 15.
ospf transmit-delay *seconds* **Example:** `hostname(config-interface)# ospf transmit-delay 5`	Sets the estimated number of seconds required to send a link-state update packet on an OSPF interface. The *seconds* value is from 1 to 65535 seconds. The default value is 1 second. In this example, the transmit-delay is 5 seconds.
ospf network point-to-point non-broadcast **Example:** `hostname(config-interface)# ospf network point-to-point non-broadcast`	Specifies the interface as a point-to-point, non-broadcast network. When you designate an interface as point-to-point, nonbroadcast, you must manually define the OSPF neighbor; dynamic neighbor discovery is not possible. See the "Defining Static OSPF Neighbors" section on page 22-12, for more information. Additionally, you can only define one OSPF neighbor on that interface.

Configuring OSPF Area Parameters

You can configure several OSPF area parameters. These area parameters (shown in the following task list) include setting authentication, defining stub areas, and assigning specific costs to the default summary route. Authentication provides password-based protection against unauthorized access to an area.

Stub areas are areas into which information on external routes is not sent. Instead, there is a default external route generated by the ABR, into the stub area for destinations outside the autonomous system. To take advantage of the OSPF stub area support, default routing must be used in the stub area. To further reduce the number of LSAs sent into a stub area, you can configure the **no-summary** keyword of the **area stub** command on the ABR to prevent it from sending summary link advertisement (LSA Type 3) into the stub area.

To specify area parameters for your network, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `hostname(config)# router ospf 2`	Creates an OSPF routing process and enters router configuration mode for the OSPF process that you want to redistribute. The *process_id* is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.
Step 2	Do one of the following to configure optional OSPF area parameters:	
	area *area-id* **authentication** **Example:** `hostname(config-router)# area 0 authentication`	Enables authentication for an OSPF area.
	area *area-id* **authentication message-digest** **Example:** `hostname(config-router)# area 0 authentication message-digest`	Enables MD5 authentication for an OSPF area.

Configuring OSPF NSSA

The OSPF implementation of an NSSA is similar to an OSPF stub area. NSSA does not flood Type 5 external LSAs from the core into the area, but it can import autonomous system external routes in a limited way within the area.

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

You can simplify administration if you are an ISP or a network administrator that must connect a central site using OSPF to a remote site that is using a different routing protocol using NSSA.

Before the implementation of NSSA, the connection between the corporate site border router and the remote router could not be run as an OSPF stub area because routes for the remote site could not be redistributed into the stub area, and two routing protocols needed to be maintained. A simple protocol such as RIP was usually run and handled the redistribution. With NSSA, you can extend OSPF to cover the remote connection by defining the area between the corporate router and the remote router as an NSSA.

Before you use this feature, consider these guidelines:

- You can set a Type 7 default route that can be used to reach external destinations. When configured, the router generates a Type 7 default into the NSSA or the NSSA area boundary router.
- Every router within the same area must agree that the area is NSSA; otherwise, the routers will not be able to communicate.

To specify area parameters for your network as needed to configure OSPF NSSA, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `hostname(config)# router ospf 2`	Creates an OSPF routing process and enters router configuration mode for the OSPF process that you want to redistribute. The *process_id* is an internally used identifier for this routing process. It can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.
Step 2	Do one of the following to configure optional OSPF NSSA parameters:	
	area *area-id* **nssa** [**no-redistribution**] [**default-information-originate**] **Example:** `hostname(config-router)# area 0 nssa`	Defines an NSSA area.

	summary-address *ip_address mask* [**not-advertise**] [**tag** *tag*]
	
	Example:
	`hostname(config)# router ospf 1`
	`hostname(config-router)# summary-address 10.1.0.0 255.255.0.0`
		Sets the summary address and helps reduce the size of the routing table. Using this command for OSPF causes an OSPF ASBR to advertise one external route as an aggregate for all redistributed routes that are covered by the address.
		
		In this example, the summary address 10.1.0.0 includes addresses 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. Only the 10.1.0.0 address is advertised in an external link-state advertisement

Note OSPF does not support summary-address 0.0.0.0 0.0.0.0.

Defining Static OSPF Neighbors

You need to define static OSPF neighbors to advertise OSPF routes over a point-to-point, non-broadcast network. This lets you broadcast OSPF advertisements across an existing VPN connection without having to encapsulate the advertisements in a GRE tunnel.

Before you begin, you must create a static route to the OSPF neighbor. See Chapter 20, "Configuring Static and Default Routes," for more information about creating static routes.

To define a static OSPF neighbor, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `hostname(config)# router ospf 2`	Creates an OSPF routing process and enters router configuration mode for this OSPF process. The *process_id* is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.
Step 2	**neighbor** *addr* [**interface** *if_name*] **Example:** `hostname(config-router)# neighbor 255.255.0.0 [interface my_interface]`	Defines the OSPF neighborhood. The *addr* argument is the IP address of the OSPF neighbor. The *if_name* is the interface used to communicate with the neighbor. If the OSPF neighbor is not on the same network as any of the directly-connected interfaces, you must specify the **interface**.

Configuring Route Calculation Timers

You can configure the delay time between when OSPF receives a topology change and when it starts an SPF calculation. You also can configure the hold time between two consecutive SPF calculations.

To configure route calculation timers, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `hostname(config)# router ospf 2`	Creates an OSPF routing process and enters router configuration mode for this OSPF process. The *process_id* is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.
Step 2	**timers spf** *spf-delay spf-holdtime* **Example:** `hostname(config-router)# timers spf 10 120`	Configures the route calculation times. The *spf-delay* is the delay time (in seconds) between when OSPF receives a topology change and when it starts an SPF calculation. It can be an integer from 0 to 65535. The default time is 5 seconds. A value of 0 means that there is no delay; that is, the SPF calculation is started immediately. The *spf-holdtime* is the minimum time (in seconds) between two consecutive SPF calculations. It can be an integer from 0 to 65535. The default time is 10 seconds. A value of 0 means that there is no delay; that is, two SPF calculations can be done, one immediately after the other.

Logging Neighbors Going Up or Down

By default, a system message is generated when an OSPF neighbor goes up or down.

Configure this command if you want to know about OSPF neighbors going up or down without turning on the **debug ospf adjacency** command. The **log-adj-changes** router configuration command provides a higher level view of the peer relationship with less output. Configure the **log-adj-changes detail** command if you want to see messages for each state change.

To log neighbors going up or down, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**router ospf** *process_id* **Example:** `hostname(config)# router ospf 2`	Creates an OSPF routing process and enters router configuration mode for this OSPF process. The *process_id* is an internally used identifier for this routing process and can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.
Step 2	**log-adj-changes** [**detail**] **Example:** `hostname(config-router)# log-adj-changes [detail]`	This step configures logging for neighbors going up or down.

Restarting the OSPF Process

To remove the entire OSPF configuration that you have enabled, enter the following command:

Command	Purpose
clear ospf *pid* {**process** | **redistribution** | **counters** [**neighbor** [*neighbor-interface*] [*neighbor-id*]]}	Removes the entire OSPF configuration that you have enabled. After the configuration is cleared, you must reconfigure OSPF using the **router ospf** command.
Example: `hostname(config)# clear ospf`	

Configuration Example for OSPF

The following example shows how to enable and configure OSPF with various optional processes:

Step 1 Enable OSPF:

```
hostname(config)# router ospf 2
hostname(config-router)# network 10.0.0.0 255.0.0.0 area 0
```

Step 2 (Optional) Redistribute routes from one OSPF process to another OSPF process:

```
hostname(config)# route-map 1-to-2 permit
hostname(config-route-map)# match metric 1
hostname(config-route-map)# set metric 5
hostname(config-route-map)# set metric-type type-1
hostname(config-route-map)# router ospf 2
hostname(config-router)# redistribute ospf 1 route-map 1-to-2
```

Step 3 Configure OSPF interface parameters (optional):

```
hostname(config)# router ospf 2
hostname(config-router)# network 10.0.0.0 255.0.0.0 area 0
hostname(config-router)# interface inside
hostname(config-interface)# ospf cost 20
hostname(config-interface)# ospf retransmit-interval 15
hostname(config-interface)# ospf transmit-delay 10
hostname(config-interface)# ospf priority 20
hostname(config-interface)# ospf hello-interval 10
hostname(config-interface)# ospf dead-interval 40
hostname(config-interface)# ospf authentication-key cisco
hostname(config-interface)# ospf message-digest-key 1 md5 cisco
hostname(config-interface)# ospf authentication message-digest
```

Step 4 (Optional) Configure OSPF area parameters:

```
hostname(config)# router ospf 2
hostname(config-router)# area 0 authentication
hostname(config-router)# area 0 authentication message-digest
hostname(config-router)# area 17 stub
hostname(config-router)# area 17 default-cost 20
```

Step 5 (Optional) Configure the route calculation timers and show the log neighbor up and down messages:

```
hostname(config-router)# timers spf 10 120
hostname(config-router)# log-adj-changes [detail]
```

Step 6 Restart the OSPF process:

```
hostname(config)# clear ospf pid {process | redistribution | counters
[neighbor [neighbor-interface] [neighbor-id]]}
```

Step 7 (Optional) Show the results of your OSPF configuration:

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

```
hostname(config)# show ospf

Routing Process "ospf 2" with ID 10.1.89.2 and Domain ID 0.0.0.2
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 5. Checksum Sum 0x 26da6
Number of opaque AS LSA 0. Checksum Sum 0x     0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
External flood list length 0
    Area BACKBONE(0)
        Number of interfaces in this area is 1
        Area has no authentication
        SPF algorithm executed 2 times
        Area ranges are
        Number of LSA 5. Checksum Sum 0x 209a3
        Number of opaque link LSA 0. Checksum Sum 0x     0
```

```
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0
```

Monitoring OSPF

You can display specific statistics such as the contents of IP routing tables, caches, and databases. You can also use the information provided to determine resource utilization and solve network problems. You can also display information about node reachability and discover the routing path that your device packets are taking through the network.

To monitor or display various OSPF routing statistics, enter one of the following commands:

Command	Purpose
show ospf [*process-id [area-id]*]	Displays general information about OSPF routing processes.
show ospf border-routers	Displays the internal OSPF routing table entries to the ABR and ASBR.
show ospf [*process-id* [*area-id*]] **database**	Displays lists of information related to the OSPF database for a specific router.
show ospf flood-list *if-name*	Displays a list of LSAs waiting to be flooded over an interface (to observe OSPF packet pacing). OSPF update packets are automatically paced so they are not sent less than 33 milliseconds apart. Without pacing, some update packets could get lost in situations where the link is slow, a neighbor could not receive the updates quickly enough, or the router could run out of buffer space. For example, without pacing packets might be dropped if either of the following topologies exist: • A fast router is connected to a slower router over a point-to-point link. • During flooding, several neighbors send updates to a single router at the same time. Pacing is also used between resends to increase efficiency and minimize lost retransmissions. You also can display the LSAs waiting to be sent out an interface. The benefit of the pacing is that OSPF update and retransmission packets are sent more efficiently. There are no configuration tasks for this feature; it occurs automatically.
show ospf interface [*if_name*]	Displays OSPF-related interface information.

Command	Purpose
show ospf neighbor [*interface-name*] [*neighbor-id*] [**detail**]	Displays OSPF neighbor information on a per-interface basis.
show ospf request-list *neighbor if_name*	Displays a list of all LSAs requested by a router.
show ospf retransmission-list *neighbor if_name*	Displays a list of all LSAs waiting to be resent.
show ospf [*process-id*] **summary-address**	Displays a list of all summary address redistribution information configured under an OSPF process.
show ospf [*process-id*] **virtual-links**	Displays OSPF-related virtual links information.

Feature History for OSPF

Table 22-1 lists each feature change and the platform release in which it was implemented.

Table 22-1 Feature History for Static and Default Routes

Feature Name	Platform Releases	Feature Information
OSPF Support	7.0(1)	Support was added for route data, perform authentication, redistribute and monitor routing information, using the Open Shortest Path First (OSPF) routing protocol. The **route ospf** command was introduced to route data, perform authentication, redistribute and monitor routing information, using the Open Shortest Path First (OSPF) routing protocol.