AT&T

# OSPF Configuration Guide, 17.2.0

November 2017

Supporting AT&T Vyatta Network Operating System

# Contents

# Copyright Statement

The training materials and other content provided herein for assistance in training on the Vyatta vRouter may have references to Brocade as the Vyatta vRouter was formerly a Brocade product prior to AT&T's acquisition of Vyatta. Brocade remains a separate company and is not affiliated to AT&T.

# About This Guide

This guide describes how to configure OSPF on AT&T products that run on the AT&T Vyatta Network Operating System (referred to as a virtual router, vRouter, or router in the guide).

# OSPF Configuration

## OSPF overview

OSPF is a dynamic routing protocol that uses a link-state algorithm (Dijkstra), as opposed to protocols (such as RIP) that use a distance-vector algorithm. It is an interior gateway protocol (IGP) and operates in a single autonomous system (AS).

> **Note:** OSPF uses both link-state algorithm (Dijkstra) within an area and distance vector algorithms across areas.

In OSPF, each router advertises the state of its own links, or connections, in a link-state advertisement (LSA), which it then multicasts to other routers on the network. In addition, each router uses the LSAs that it receives from other routers to construct a directed graph with itself as the root representing the network topology.

To build its routing table, the router applies Dijkstra's Shortest Path First algorithm within an area to find the best path through the graph to each network in the topology that is within the area. This "shortest path tree" becomes the basis of the routing table. OSPF is hierarchical. In OSPF, the network is divided into "areas." Within each area, routers possess only local routing information. Routing information about other areas is calculated by using distance vector protocol and forming summarized routes. This reduces the amount of network topology information that routers have to generate and maintain, making OSPF a good choice for larger networks.

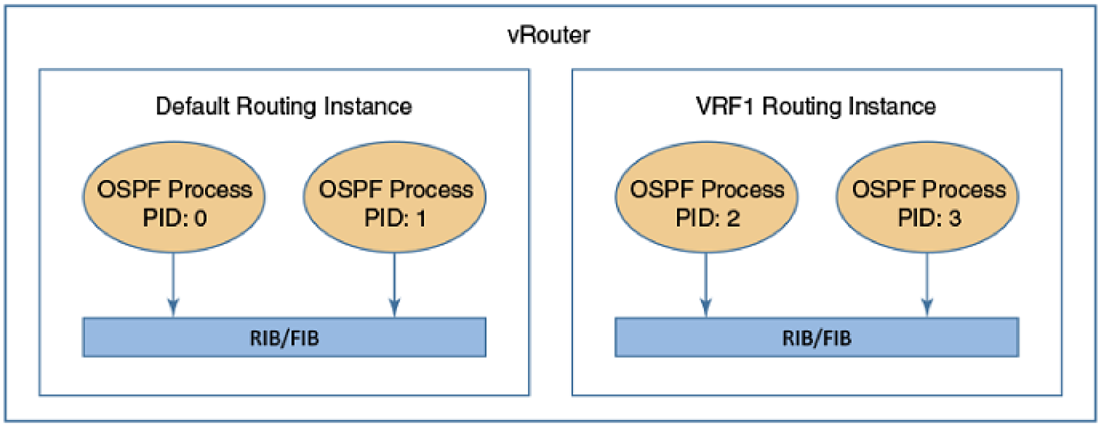
## Supported standards

The AT&Timplementation of OSPF complies with the following standard:

- RFC 2328: OSPF Version 2
- RFC4750: OSPF Version 2 Management Information Base
- RFC 3101: OSPF Not-So-Stubby Area (NSSA) Option

## Support for multiple OSPF processes

By default, when you configure OSPF without specifying a process ID (PID), the system creates an OSPF process and assigns it a PID of 0. However, you can create additional OSPF processes in the context of a routing instance by giving each process a unique PID in the range of 1 through 65535. The following figure shows multiple OSPF processes that run in different routing instances.



To set the PID for an OSPF process, use the **`process`** *pid* keyword and variable with **`protocols ospf ...`** commands, as illustrated in the following example, which shows how to configure two OSPF processes for the default routing instance.

When configuring OSPF processes, note the following:

- The PID of an OSPF process is local to the router.
- Routes can be redistributed among OSPF processes that belong to the same routing instance.
- An OSPF process maintains its own route database and does not share it with other OSPF processes.
- RIB is common to all OSPF processes defined for a routing instance.
- An interface can belong to only one OSPF process.

**Table 1: Configuring multiple OSPF processes**

| Step | Command |
|---|---|
| Set the identifier for the first OSPF process. | `vyatta@vyatta# set protocols ospf process 1` |
| Set the identifier for the second OSPF process. | `vyatta@vyatta# set protocols ospf process 2` |
| Commit the configuration. | `vyatta@vyatta# commit` |
| Display the configuration. | `vyatta@vyatta# show protocols ospf`<br>` ospf {`<br>`        process 1`<br>`        process 2`<br>` }` |

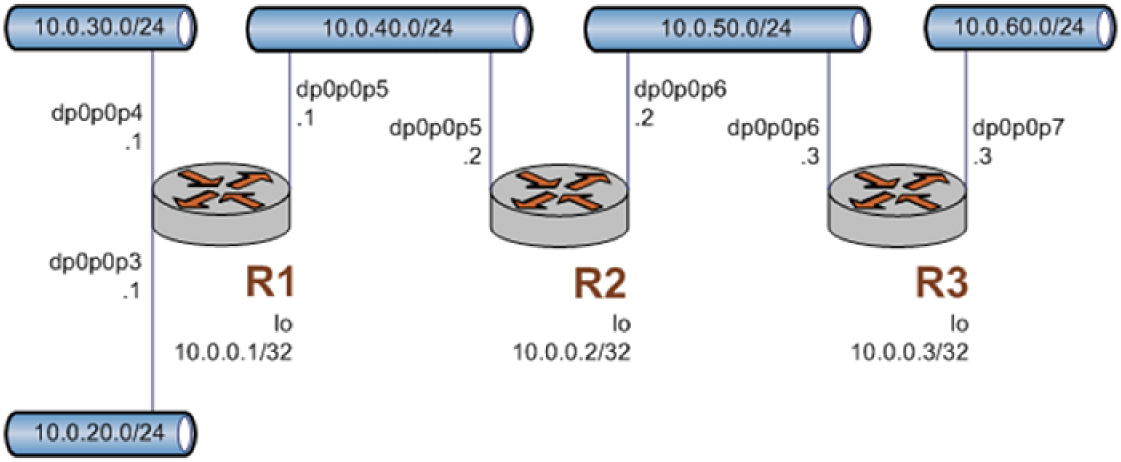
## Configuring OSPF

This section presents the following topics:

- Basic OSPF configuration *(page 12)*
- Verifying the OSPF configuration *(page 15)*

This section presents an example configuration of OSPF. The example is based on the following reference diagram.

**Figure 2: OSPF configuration diagram**



## Basic OSPF configuration

In this section, you configure OSPF on the routers that are labeled R1, R2, and R3 in the OSPF configuration diagram. The routers advertise their routes on the 10.0.40/24 and 10.0.50.0/24 networks.

For this example, it is assumed that you have already configured the router interfaces, including the loopback interfaces (lo); only the steps that are required to implement OSPF are shown.

> **Note:** Setting the router-id, loopback interface, and redistribution of router area are not mandatory requirements for basic OSPF configuration.

To create a basic OSPF configuration, perform the following steps in configuration mode.

**Table 2: Configuring OSPF**

| Router | Step | Command |
|---|---|---|
| R1 | Set the identifier of the OSPF router to that of the loopback address. | `vyatta@R1# set protocols ospf process 3 parameters router-id 10.0.0.1` |
| R1 | Run OSPF on the 10.0.40.0/24 network. | `vyatta@R1# set protocols ospf process 3 area 0 network 10.0.40.0/24` |
| R1 | Redistribute the networks connected to OSPF. | `vyatta@R2# set protocols ospf process 3 redistribute connected` |
| R1 | Commit the configuration. | `vyatta@R1# commit` |
| R1 | Display the configuration. | `vyatta@R1# show protocols ospf {`<br>`process 3 {`<br>`area 0 {`<br>`network 10.0.40.0/24`<br>`}`<br>`parameters {`<br>`router-id 10.0.0.1`<br>`}`<br>`redistribute {`<br>`connected`<br>`}`<br>`}`<br>`}` |
| R2 | Set the identifier of the OSPF router to that of the loopback address. | `vyatta@R2# set protocols ospf parameters router-id 10.0.0.2` |

| Router | Step | Command |
|---|---|---|
| R2 | Run OSPF on the 10.0.40.0/24 network. | `vyatta@R2# set protocols ospf process 4 area 0 network 10.0.40.0/24` |
| R2 | Run OSPF on the 10.0.0.2/32 network. | `vyatta@R2# set protocols ospf process 4 area 0 network 10.0.0.2/32` |
| R2 | Redistribute the networks connected to OSPF. | `vyatta@R2# set protocols ospf process 4 redistribute connected` |
| R2 | Redistribute connected routes from OSPF process 5 to OSPF process 4. | `vyatta@R2# set protocols ospf process 4 redistribute ospf process-id 5` |
| R2 | Run OSPF on the 10.0.50.0/24 network. | `vyatta@R2# set protocols ospf process 5 area 0 network 10.0.50.0/24` |
| R2 | Redistribute the networks connected to OSPF. | `vyatta@R2# set protocols ospf process 5 redistribute connected` |
| R2 | Redistribute connected routes from OSPF process 4 to OSPF process 5. | `vyatta@R2# set protocols ospf process 5 redistribute ospf process-id 4` |
| R2 | Commit the configuration. | `vyatta@R2# commit` |

<table>
<tr><th>Router</th><th>Step</th><th>Command</th></tr>
<tr><td>R2</td><td>Display the configuration.</td><td>

```
vyatta@R2# show protocols
 ospf {
        parameters {
                router-id
 10.0.0.2
        }
        process 4 {
                area 0 {

 network 10.0.40.0/24

 network 10.0.0.2/32
                }
                redistribute
 {

 connected
                        ospf
 {

    process-id 5
                        }
                }
        }
        process 5 {
                area 0 {

 network 10.0.50.0/24
                }
                redistribute
 {

 connected
                        ospf
 {

    process-id 4
                        }
                }
        }
 }
```

</td></tr>
<tr><td>R3</td><td>Set the identifier of the OSPF router to that of the loopback address.</td><td>

```
vyatta@R3# set protocols
 ospf parameters router-id
 10.0.0.3
```

</td></tr>
<tr><td>R3</td><td>Run OSPF on the 10.0.50.0/24 network.</td><td>

```
vyatta@R3# set protocols
 ospf area 0 network
 10.0.50.0/24
```

</td></tr>
<tr><td>R3</td><td>Redistribute the networks connected to OSPF.</td><td>

```
vyatta@R3# set protocols
 ospf redistribute connected
```

</td></tr>
<tr><td>R3</td><td>Commit the configuration.</td><td>

```
vyatta@R3# commit
```

</td></tr>
</table>

| Router | Step | Command |
|---|---|---|
| R3 | Display the configuration. | `vyatta@R3# show protocols`<br>`ospf {`<br>`        area 0 {`<br>`                network`<br>`10.0.50.0/24`<br>`        }`<br>`        parameters {`<br>`                router-id`<br>`10.0.0.3`<br>`        }`<br>`        redistribute {`<br>`                connected`<br>`        }`<br>`}` |

## Verifying the OSPF configuration

You can use the following operational mode commands to verify the OSPF configuration for this example.

### Verifying the OSPF configuration for R2

The following example shows the output of the `show ip route` command for R2. The output shows that routes to 10.0.0.1/32, 10.0.0.3/32, 10.0.20.0/24, 10.0.30.0/24, and 10.0.60.0/24, which have been learned through OSPF processes 4 and 5, are present in the routing table and are the selected routes that are copied to the FIB. The 10.0.0.2/32, 10.0.40.0/24, and 10.0.50.0/24 networks are directly connected to R2. Directly connected routes have a preference over OSPF-discovered routes.

```
vyatta@R2:~$ show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       L - LDP, V - RSVP-TE, I - IGP shortcut
       > - selected route, * - FIB route, p - stale info

IP Route Table for Routing-instance "default"
O    *> 10.0.0.1/32 [110/1] via 10.0.40.1, dp0p0p5, 00:36:37
C    *> 10.0.0.2/32 is directly connected, lo
O    *> 10.0.0.3/32 [110/1] via 10.0.50.3, dp0p0p6, 00:26:45
O E2 *> 10.0.20.0/24 [110/20] via 10.0.40.1, dp0p0p5, 00:36:37
O E2 *> 10.0.30.0/24 [110/20] via 10.0.40.1, dp0p0p5, 00:36:37
C    *> 10.0.40.0/24 is directly connected, dp0p0p5
C    *> 10.0.50.0/24 is directly connected, dp0p0p6
O E2 *> 10.0.60.0/24 [110/20] via 10.0.50.3, dp0p0p6, 00:26:45
C    *> 127.0.0.0/8 is directly connected, lo
vyatta@R2:~$
```

The following example shows the output of the `show ip ospf neighbor` command. The output shows that there are no OSPF neighbors in the default process (process 0).

```
vyatta@R2:~$ show ip ospf neighbor
Total number of full neighbors: 0
OSPF process 0 routing-instance(default):
Neighbor ID     Pri   State            Dead Time   Address         Interface
```

The following example shows the output of the `show ip ospf process 4 neighbor` command. The output shows that there is only one OSPF neighbor for OSPF process 4.

```
vyatta@R2:~$ show ip ospf process 4 neighbor
Total number of full neighbors: 1
OSPF process 4 routing-instance(default):
Neighbor ID     Pri   State            Dead Time   Address         Interface
10.0.0.1          1   Full/Backup      00:00:36    10.0.40.1       dp0p0p5
```

The following example shows the output of the `show ip ospf process 5 neighbor` command. The output shows that there is only one OSPF neighbor for the configured OSPF process 5.

```
vyatta@R2:~$ show ip ospf process 5 neighbor
Total number of full neighbors: 1
OSPF process 5 routing-instance(default):
Neighbor ID     Pri   State            Dead Time   Address         Interface
10.0.0.3          1   Full/DR          00:00:38    10.0.50.3       dp0p0p6
```

The following example shows the output of the `show ip route ospf` command. The output shows the OSPF routes in the routing table.

```
vyatta@R2:~$ show ip route ospf
IP Route Table for Routing-instance "default"
O    *> 10.0.0.1/32 [110/1] via 10.0.40.1, dp0p0p5, 00:36:45
O    *> 10.0.0.3/32 [110/1] via 10.0.50.3, dp0p0p6, 00:26:53
O E2 *> 10.0.20.0/24 [110/20] via 10.0.40.1, dp0p0p5, 00:36:45
O E2 *> 10.0.30.0/24 [110/20] via 10.0.40.1, dp0p0p5, 00:36:45
O E2 *> 10.0.60.0/24 [110/20] via 10.0.50.3, dp0p0p6, 00:26:53
```

The following example shows the output of the `show ip ospf route` command. The output shows that no routes have been learned by the default OSPF process (process 0).

```
vyatta@R2:~$ show ip ospf route
vyatta@R2:~$
```

The following example shows the output of the `show ip ospf process 4 route` command. The output shows the routes that have been learned by OSPF process 4.

```
vyatta@R2:~$ show ip ospf process 4 route

OSPF process 4 routing-instance: default
Codes: C - connected, D - Discard, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2

O  10.0.0.1/32 [1] via 10.0.40.1, dp0p0p5, Area 0.0.0.0
C  10.0.0.2/32 [0] is directly connected, lo, Area 0.0.0.0
E2 10.0.20.0/24 [1/20] via 10.0.40.1, dp0p0p5
E2 10.0.30.0/24 [1/20] via 10.0.40.1, dp0p0p5
C  10.0.40.0/24 [1] is directly connected, dp0p0p5, Area 0.0.0.0
```

The following example shows the output of the `show ip ospf process 5 route` command. The output shows the routes that have been learned by OSPF process 5.

```
vyatta@R2:~$ show ip ospf process 5 route
OSPF process 5 routing-instance: default
Codes: C - connected, D - Discard, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2

O  10.0.0.3/32 [1] via 10.0.50.3, dp0p0p6, Area 0.0.0.0
C  10.0.50.0/24 [1] is directly connected, dp0p0p6, Area 0.0.0.0
E2 10.0.60.0/24 [1/20] via 10.0.50.3, dp0p0p6
```

### Verifying the OSPF configuration for R3

The following example shows the output of the `show ip route` command for R3. The output shows that routes to 10.0.0.1/32, 10.0.0.2/32, 10.0.20.0/24, 10.0.30.0/24, and 10.0.40.0/24 have been learned through OSPF (and are the selected routes). The 10.0.0.3/32, 10.0.50.0/24, and 10.0.60.0/24 networks are directly connected to R3. Directly connected routes are always preferred over OSPF-discovered routes.

```
vyatta@R3:~$ show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       L - LDP, V - RSVP-TE, I - IGP shortcut
       > - selected route, * - FIB route, p - stale info

IP Route Table for Routing-instance "default"
O E2 *> 10.0.0.1/32 [110/20] via 10.0.50.2, dp0p0p6, 00:14:26
O E2 *> 10.0.0.2/32 [110/20] via 10.0.50.2, dp0p0p6, 00:10:15
C    *> 10.0.0.3/32 is directly connected, lo
O E2 *> 10.0.20.0/24 [110/20] via 10.0.50.2, dp0p0p6, 00:14:26
O E2 *> 10.0.30.0/24 [110/20] via 10.0.50.2, dp0p0p6, 00:14:26
O E2 *> 10.0.40.0/24 [110/20] via 10.0.50.2, dp0p0p6, 00:10:15
C    *> 10.0.50.0/24 is directly connected, dp0p0p6
C    *> 10.0.60.0/24 is directly connected, dp0p0p7
C    *> 127.0.0.0/8 is directly connected, lo
vyatta@R3:~$
```

The following example shows the output of the `show ip route ospf` command for router R3. This output shows OSPF routes only.

```
vyatta@R3:~$ show ip route ospf
IP Route Table for Routing-instance "default"
O E2 *> 10.0.0.1/32 [110/20] via 10.0.50.2, dp0p0p6, 00:14:30
O E2 *> 10.0.0.2/32 [110/20] via 10.0.50.2, dp0p0p6, 00:10:19
O E2 *> 10.0.20.0/24 [110/20] via 10.0.50.2, dp0p0p6, 00:14:30
O E2 *> 10.0.30.0/24 [110/20] via 10.0.50.2, dp0p0p6, 00:14:30
O E2 *> 10.0.40.0/24 [110/20] via 10.0.50.2, dp0p0p6, 00:10:19
vyatta@R3:~$
```

The following example shows the output of the `show ip ospf neighbor` command for R3.

```
vyatta@R3:~$ show ip ospf neighbor

Total number of full neighbors: 1
OSPF process 0 routing-instance(default):
Neighbor ID     Pri   State            Dead Time   Address         Interface
10.0.0.2          1   Full/Backup      00:00:33    10.0.50.2       dp0p0p6
vyatta@R3:~$
```

The following commands ping IP addresses of hosts in networks connected to R1. Pinging these hosts confirms that the OSPF configuration is working and that you can reach a remote network.

```
vyatta@R3:~$ ping 10.0.20.1
PING 10.0.20.1 (10.0.20.1) 56(84) bytes of data.
64 bytes from 10.0.20.1: icmp_seq=1 ttl=63 time=0.752 ms
64 bytes from 10.0.20.1: icmp_seq=2 ttl=63 time=0.595 ms
64 bytes from 10.0.20.1: icmp_seq=3 ttl=63 time=0.598 ms
^C
--- 10.0.20.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.595/0.648/0.752/0.076 ms

vyatta@R3:~$ ping 10.0.30.1
PING 10.0.30.1 (10.0.30.1) 56(84) bytes of data.
64 bytes from 10.0.30.1: icmp_seq=1 ttl=63 time=0.752 ms
64 bytes from 10.0.30.1: icmp_seq=2 ttl=63 time=0.993 ms
64 bytes from 10.0.30.1: icmp_seq=3 ttl=63 time=0.685 ms
^C
--- 10.0.30.1 ping statistics ---
```

```
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.685/0.810/0.993/0.132 ms
```

## Removing router-id overlap

If the vRouter detects that multiple OSPF or OSPFv3 processes that are running in a routing instance have the same router ID, the vRouter sends router-id overlap notifications to syslog. The vRouter also displays router-id overlap notifications in the output of the **`show ip ospf`** and **`show ipv6 ospfv3`** commands.

In the following example, the lines that start with the "**" characters indicate an overlap in the router IDs that were allocated by the vRouter.

```
vyatta@vyatta# set protocols ospf
[edit]
vyatta@vyatta# set protocols ospf process 7
[edit]
vyatta@vyatta# commit
[edit]
vyatta@vyatta# run show ip ospf
 Routing Process "ospf 0" with ID 1.1.1.1
** Allocated Router ID 1.1.1.1 overlaps with one or more ospf processes - please reconfigure **
 Process uptime is 4 minutes
 Process bound to routing-instance default
 Conforms to RFC2328, and RFC1583 Compatibility flag is disabled
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Does not support Restarting
 SPF schedule delay initial 0 secs 500 msecs
 SPF schedule delay min 0 secs 500 msecs
 SPF schedule delay max 50 secs 0 msecs
 Refresh timer 10 secs
 Number of incoming current DD exchange neighbors 0/64
 Number of outgoing current DD exchange neighbors 0/64
 Initial LSA throttle delay 0 secs 0 msecs
 Minimum hold time for LSA throttle 5 secs 0 msecs
 Maximum wait time for LSA throttle 5 secs 0 msecs
 Minimum LSA arrival 1 secs 0 msecs
 Number of external LSA 0. Checksum 0x000000
 Number of opaque AS LSA 0. Checksum 0x000000
 Number of non-default external LSA 0
 External LSA database is unlimited.
 Number of LSA originated 0
 Number of LSA received 0
 Number of areas attached to this router: 0

[edit]
vyatta@vyatta# run show ip ospf process 7
 Routing Process "ospf 7" with ID 1.1.1.1
** Allocated Router ID 1.1.1.1 overlaps with one or more ospf processes - please reconfigure **
 Process uptime is 12 minutes
...
[edit]
```

The following example shows an overlap between the router ID that the vRouter allocated to process 0 and the router ID that a user configured for process 4.

```
vyatta@vyatta# set protocols ospf process 4 parameters router-id 1.1.1.1
vyatta@vyatta# set protocols ospf
vyatta@vyatta# commit
vyatta@vyatta# run show ip ospf
 Routing Process "ospf 0" with ID 1.1.1.1
** Allocated Router ID 1.1.1.1 overlaps with one or more ospf processes - please reconfigure **
 Process uptime is 27 minutes
...
vyatta@vyatta# run show ip ospf process 4
```

```
 Routing Process "ospf 4" with ID 1.1.1.1
** Configured Router ID 1.1.1.1 overlaps with one or more ospf processes - please reconfigure **
 Process uptime is 6 minutes
 ...
```

The following example shows overlap in router IDs allocated to OSPFv3 processes by the vRouter.

```
vyatta@vyatta# set protocols ospfv3
[edit]
vyatta@vyatta# set protocols ospfv3 process 1
[edit]
vyatta@vyatta# commit
[edit]
vyatta@vyatta# run show ipv6 ospfv3
 Routing Process "OSPFv3 (*null*)" with ID 10.18.170.230
** Allocated Router ID 10.18.170.230 overlaps with one or more processes - please reconfigure **
 Process uptime is 0 minute
 Process bound to routing-instance default
 SPF schedule delay initial 0.500 secs
 SPF schedule delay min 0.500 secs
 SPF schedule delay max 50.0 secs
 Minimum LSA interval 5 secs, Minimum LSA arrival 1 secs
 Number of incoming current DD exchange neighbors 0/5
 Number of outgoing current DD exchange neighbors 0/5
 Number of external LSA 0. Checksum Sum 0x0000
 Number of AS-Scoped Unknown LSA 0
 Number of LSA originated 0
 Number of LSA received 0
 Number of areas in this router is 0

[edit]
vyatta@vyatta# run show ipv6 ospfv3 process 1
 Routing Process "OSPFv3 (1)" with ID 10.18.170.230
** Allocated Router ID 10.18.170.230 overlaps with one or more processes - please reconfigure **
 Process uptime is 1 minute
 ...
[edit]
```

The following example shows an overlap between the router ID that the vRouter allocated to OSPFv3 process 3 and the router ID that a user configured for OSPFv3 process 0.

```
vyatta@vyatta# set protocols ospfv3 router-id 1.1.1.1
[edit]
vyatta@vyatta# set protocols ospfv3 process 3
[edit]
vyatta@vyatta# commit
[edit]
vyatta@vyatta# run show ipv6 ospfv3
 Routing Process "OSPFv3 (*null*)" with ID 1.1.1.1
** Configured Router ID 10.18.170.230 overlaps with one or more processes - please reconfigure **
 Process uptime is 0 minute
...
[edit]
vyatta@vyatta# run show ipv6 ospfv3 process 3
 Routing Process "OSPFv3 (3)" with ID 1.1.1.1
** Allocated Router ID 1.1.1.1 overlaps with one or more processes - please reconfigure **
 Process uptime is 1 minute
 ...
[edit]
```

To remove router-id overlap, use the **`set protocols ospf parameters router-id`** and **`set protocols ospfv3 router-id`** commands to configure unique router IDs for OSPF and OSPFv3 processes, respectively.

# OSPF Commands

## monitor protocol ospf disable all

Disables the generation of all debug messages.

**Syntax:**

```
monitor protocol ospf[ process pid ] disable all
```

***pid***

Specifies the process ID (PID) of an IPv4 OSPF process.

**Operational mode**

Use this command to disable the generation of all debug messages.

## monitor protocol ospf disable bfd

Disables the generation of BFD debug messages.

**Syntax:**

```
monitor protocol ospf [ process pid ] disable bfd
```

***pid***

Specifies the PID of an IPv4 OSPF process.

**Operational mode**

Use this command to disable the generation of BFD debug messages.

## monitor protocol ospf disable database-timer rate-limit

Disables the generation of debug messages that are related to the OSPF database timer.

**Syntax:**

```
monitor protocol ospf [ process pid ] disable database-timer rate-limit
```

**process *pid***

Specifies the PID of an IPv4 OSPF process.

**rate-limit**

Disables the debugging of the timer.

**Operational mode**

Use this command to disable the generation of debug messages that are related to the OSPF database timer.

## monitor protocol ospf disable events

Disables the generation of debug messages that are related to OSPF events.

**Syntax:**

```
monitor protocol ospf [ process pid ] disable events [ abr | asbr | lsa | nssa | os | router | vlink ]
```

**process *pid***

Specifies the PID of an IPv4 OSPF process.

**abr**

Disables debugging of area border router (ABR) events.

**asbr**

Disables debugging of autonomous system boundary router (ASBR) events.

**lsa**
: Disables debugging of link-state advertisement (LSA) events.

**nssa**
: Disables debugging of not-so-stubby area (NSSA) events.

**os**
: Disables debugging of OS-interaction events.

**router**
: Disables debugging of other router events.

**vlink**
: Disables debugging of virtual-link events.

**Operational mode**

Use this command to disable the generation of debug messages that are related to OSPF events.

# monitor protocol ospf disable ifsm

Disables the generation of debug messages that are related to OSPF IFSM events.

**Syntax:**

```
monitor protocol ospf [ process pid ] disable ifsm [ events | status | timers ]
```

OSPF IFSM event debugging is disabled.

**process** ***pid***
: Specifies the PID of an IPv4 OSPF process.

**events**
: Disables debugging of IFSM events.

**status**
: Disables debugging of IFSM status.

**timers**
: Disables debugging of IFSM timers.

**Operational mode**

Use this command to disable the generation of debug messages that are related to OSPF Interface Finite State Machine (IFSM).

# monitor protocol ospf disable lsa

Disables the generation of debug messages that are related to OSPF LSAs.

**Syntax:**

```
monitor protocol ospf [ process pid ] disable lsa [ flooding | generate | install | maxage | refresh ]
```

When used with no option, this command disables debugging for all OSPF LSA activity.

**process** ***pid***
: Specifies the PID of an IPv4 OSPF process.

**flooding**
: Disables debugging of OSPF LSA flood events.

**generate**
: Disables debugging of OSPF LSA generation.

**install**
: Disables debugging of OSPF LSA installation.

**maxage**
: Disables debugging of maximum-age-related OSPF LSA processing.

**refresh**
: Disables debugging of OSPF LSA refreshes.

**Operational mode**

Use this command to disable the generation of debug messages that are related to OSPF link-state advertisements (LSAs).

# monitor protocol ospf disable nfsm

Disables the generation of debug messages that are related to OSPF NFSMs.

**Syntax:**

```
monitor protocol ospf [ process pid ] disable nfsm [ events | status | timers ]
```

OSPF NFSM debugging is disabled.

`process` ***pid***
: Specifies the PID of an IPv4 OSPF process.

`events`
: Disables debugging of OSPF NFSM events.

`status`
: Disables debugging of OSPF NFSM status.

`timers`
: Disables debugging of OSPF NFSM timers.

**Operational mode**

Use this command to disable the generation of debug messages that are related to OSPF Neighbor State Machines (NFSMs).

# monitor protocol ospf disable nsm

Disables the generation of debug messages that are related to OSPF NSM.

**Syntax:**

```
monitor protocol ospf [ process pid ] disable nsm  interface
```

OSPF NSM debugging is disabled.

`process` ***pid***
: Specifies the PID of an IPv4 OSPF process.

`interface`
: Disables debugging of OSPF NSM interface events.

**Operational mode**

Use this command to disable the generation of debug messages related to the OSPF NSM.

# monitor protocol ospf disable packet dd

Disables the generation of debug messages that are related to OSPF DD packets.

**Syntax:**

```
monitor protocol ospf [ process pid ] disable packet dd [ detail | recv [ detail ] | send [ detail ] ]
```

Debug messages are disabled for OSPF DD packets at a medium level of detail.

`process` ***pid***
: Specifies the PID of an IPv4 OSPF process.

`detail`
: Disables detailed debug messages for OSPF DD packets, both sent and received.

`recv`
: Disables debug messages for received OSPF DD packets.

`detail`
: Disables detailed debug messages for received OSPF DD packets.

`send`
: Disables debug messages for transmitted OSPF DD packets.

`detail`
: Disables detailed debug messages for transmitted OSPF DD packets.

### Operational mode

Use this command to disable the generation of debug messages that are related to OSPF Database Description (DD) packets.

## monitor protocol ospf disable packet detail

Disables the generation of debug messages that are related to OSPF detailed packets.

**Syntax:**

```
monitor protocol ospf [ process pid ] disable packet detail
```

Debug messages are disabled for OSPF detailed packets.

**process *pid***
Specifies the PID of an IPv4 OSPF process.

### Operational mode

Use this command to disable the generation of debug messages that are related to OSPF detailed packets.

## monitor protocol ospf disable packet hello

Disables the generation of debug messages that are related to OSPF hello packets.

**Syntax:**

```
monitor protocol ospf [ process pid ] disable packet hello [ detail | recv [ detail ] | send [ detail ]
]
```

Debug messages are disabled for OSPF hello packets at a medium level of detail.

**process *pid***
Specifies the PID of an IPv4 OSPF process.

**detail**
Disables detailed debug messages for OSPF hello packets, both sent and received.

**recv**
Disables debug messages for received OSPF hello packets.

**detail**
Disables detailed debug messages for received OSPF hello packets.

**send**
Disables debug messages for transmitted OSPF hello packets.

**detail**
Disables detailed debug messages for transmitted OSPF hello packets.

### Operational mode

Use this command to disable the generation of debug messages that are related to OSPF hello packets.

## monitor protocol ospf disable packet ls-ack

Disables the generation of debug messages that are related to OSPF LS Ack packets.

**Syntax:**

```
monitor protocol ospf [ process pid ] disable packet ls-ack [ detail | recv [ detail ] | send [ detail ]
]
```

Debug messages are disabled for OSPF LS Ack packets at a medium level of detail.

**process *pid***
Specifies the PID of an IPv4 OSPF process.

**detail**
Disables detailed debug messages for OSPF LS Ack packets, both sent and received.

**recv**
Disables debug messages for received OSPF LS Ack packets.

**detail**
Disables detailed debug messages for received OSPF LS Ack packets.

**send**
Disables debug messages for transmitted OSPF LS Ack packets.

**detail**
Disables detailed debug messages for transmitted OSPF LS Ack packets.

### Operational mode

Use this command to disable the generation of debug messages related to OSPF link-state acknowledgment (LS Ack) packets.

## monitor protocol ospf disable packet ls-request

Disables the generation of debug messages that are related to OSPF LSR packets.

### Syntax:

```
monitor protocol ospf [ process pid ] disable packet ls-request [ detail | recv [ detail ] | send [
detail ] ]
```

Debug messages are disabled for OSPF LSR packets at a medium level of detail.

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

**detail**
Disables detailed debug messages for OSPF LSR packets, both sent and received.

**recv**
Disables debug messages for received OSPF LSR packets.

**detail**
Disables detailed debug messages for received OSPF LSR packets.

**send**
Disables debug messages for transmitted OSPF LSR packets.

**detail**
Disables detailed debug messages for transmitted OSPF LSR packets.

### Operational mode

Use this command to disable the generation of debug messages that are related to OSPF link-state request (LSR) packets.

## monitor protocol ospf disable packet ls-update

Disables the generation of debug messages that are related to OSPF LSU packets.

### Syntax:

```
monitor protocol ospf [ process pid ] disable packet ls-update [ detail | recv [ detail ] | send [
detail ] ]
```

Debug messages are disabled for OSPF LSU packets at a medium level of detail.

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

**detail**
Disables detailed debug messages for OSPF LSU packets, both sent and received.

**recv**
Disables debug messages for received OSPF LSU packets.

**detail**
Disables detailed debug messages for received OSPF LSU packets.

**send**
Disables debug messages for transmitted OSPF LSU packets.

**detail**
Disables detailed debug messages for transmitted OSPF LSU packets.

**Operational mode**

Use this command to disable the generation of debug messages that are related to OSPF link-state update (LSU) packets.

## monitor protocol ospf disable packet recv

Disables the generation of debug messages that are related to OSPF received packets.

**Syntax:**

```
monitor protocol ospf [ process pid ] disable packet recv
```

Debug messages are disabled for OSPF received packets.

`process` ***pid***
Specifies the PID of an IPv4 OSPF process.

**Operational mode**

Use this command to disable the generation of debug messages that are related to OSPF received packets.

## monitor protocol ospf disable packet send

Disables the generation of debug messages that are related to OSPF sent packets.

**Syntax:**

```
monitor protocol ospf [ process pid ] disable packet send
```

Debug messages are disabled for OSPF sent packets.

`process` ***pid***
Specifies the PID of an IPv4 OSPF process.

**Operational mode**

Use this command to disable the generation of debug messages that are related to OSPF sent packets.

## monitor protocol ospf disable rib

Disables the generation of debug messages that are related to OSPF RIB messages.

**Syntax:**

```
monitor protocol ospf [ process pid ] disable rib [ interface | redistribute ]
```

***pid***
Specifies the ID, which is a string of alphanumeric characters, of an OSPF process.

**interface**
Disbales debugging of OSPF RIB interface messages.

**redistribute**
Disables debugging of OSPF RIB redistribute messages.

**Operational mode**

Use this command to disable the generation of debug messages that are related to OSPF RIB messages.

## monitor protocol ospf disable route

Disables the generation of debug messages that are related to OSPF route calculations.

**Syntax:**

```
monitor protocol ospf[ process pid ] disable route [ ase | ia | install | spf ]
```

Debug messages are disabled for OSPF route calculations.

`process` ***pid***

Specifies the PID of an IPv4 OSPF process.

`ase`

Disables detailed debugging of OSPF external route calculations.

`ia`

Disables detailed debugging of OSPF interarea route calculations.

`install`

Disables detailed debugging of OSPF route installation.

`spf`

Disables detailed debugging of OSPF Shortest Path First calculation.

**Operational mode**

Use this command to disable the generation of debug messages that are related to OSPF route calculations.

# monitor protocol ospf enable all

Enables the generation of all OSPF debug messages.

**Syntax:**

```
monitor protocol ospf [ process pid ] enable all
```

***pid***

Specifies the ID, which is a string of alphanumeric characters, of an OSPF process.

**Operational mode**

Use this command to enable the generation of all OSPF debug messages.

# monitor protocol ospf enable bfd

Enables the generation of BFD debug messages.

**Syntax:**

```
monitor protocol ospf [ process pid ] enable bfd
```

***pid***

Specifies the ID, which is a string of alphanumeric characters, of an OSPF process.

**Operational mode**

Use this command to enable the generation of BFD debug messages.

# monitor protocol ospf enable database-timer

Enables the generation of debug messages that are related to the OSPF database timer.

**Syntax:**

```
monitor protocol ospf [ process pid ] enable database-timer [ rate-limit ]
```

`process` ***pid***

Specifies the PID of an IPv4 OSPF process.

`rate-limit`

Enables debugging of the timer.

**Operational mode**

Use this command to enable the generation of debug messages that are related to the OSPF database timer.

# monitor protocol ospf enable events

Enables the generation of debug messages that are related to OSPF events.

**Syntax:**

```
monitor protocol ospf [ process pid ] enable events [ abr | asbr | lsa | nssa | os | router | vlink ]
```

**process *pid***
Specifies the PID of an IPv4 OSPF process.

**abr**
Enables debugging of Area Border Router (ABR) events.

**asbr**
Enables debugging of Autonomous System Boundary Router (ASBR) events.

**lsa**
Enables debugging of link-state advertisement (LSA) events.

**nssa**
Enables debugging of not-so-stubby area (NSSA) events.

**os**
Enables debugging of OS-interaction events.

**router**
Enables debugging of other router events.

**vlink**
Enables debugging of virtual-link events.

### Operational mode

Use this command to enable the generation of debug messages that are related to OSPF events.

## monitor protocol ospf enable ifsm

Enables the generation of debug messages that are related to OSPF IFSM events.

### Syntax:

```
monitor protocol ospf[ process pid ] enable ifsm [ events | status | timers ]
```

**process *pid***
Specifies the PID of an IPv4 OSPF process.

**events**
Enables debugging of IFSM events.

**status**
Enables debugging of IFSM status.

**timers**
Enables debugging of IFSM timers.

### Operational mode

Use this command to enable the generation of debug messages that are related to OSPF Interface Finite State Machine (IFSM) events.

## monitor protocol ospf enable lsa

Enables the generation of debug messages that are related to OSPF LSAs.

### Syntax:

```
monitor protocol ospf[ process pid ] enable lsa [ flooding | generate | install | maxage | refresh ]
```

When used with no option, this command enables debugging for all OSPF LSA activity.

**process *pid***
Specifies the PID of an IPv4 OSPF process.

**flooding**
Enables debugging of OSPF LSA flood events.

**generate**
Enables debugging of OSPF LSA generation.

**install**
Enables debugging of OSPF LSA installation.

**maxage**

Enables debugging of maximum-age processing that is related to OSPF LSAs.

`refresh`

Enables debugging of OSPF LSA refreshes.

**Operational mode**

Use this command to enable the generation of debug messages that are related to OSPF LSAs.

# monitor protocol ospf enable nfsm

Enables the generation of debug messages that are related to OSPF NFSMs.

**Syntax:**

```
monitor protocol ospf[ process pid ] enable nfsm [ events | status | timers ]
```

`process` ***pid***

Specifies the PID of an IPv4 OSPF process.

`events`

Enables debugging of OSPF NFSM events.

`status`

Enables debugging of OSPF NFSM status.

`timers`

Enables debugging of OSPF NFSM timers.

**Operational mode**

Use this command to enable the generation of debug messages that are related to OSPF Neighbor Finite State Machine (NFSM) events.

# monitor protocol ospf enable nsm

Enables the generation of debug messages that are related to OSPF NSM events.

**Syntax:**

```
monitor protocol ospf [ process pid ] enable nsm  interface
```

`process` ***pid***

Specifies the PID of an IPv4 OSPF process.

`interface`

Enables debugging of OSPF NSM interface events.

**Operational mode**

Use this command to enable the generation of debug messages that are related to OSPF Network Services Module (NSM) events.

# monitor protocol ospf enable packet dd

Enables the generation of debug messages that are related to OSPF DD packets.

**Syntax:**

```
monitor protocol ospf [ process pid ] enable packet dd [ detail | recv [ detail ] | send [ detail ] ]
```

Debug messages are generated for OSPF DD packets at a medium level of detail.

`process` ***pid***

Specifies the PID of an IPv4 OSPF process.

`detail`

Generates detailed debug messages for OSPF DD packets, both sent and received.

`recv`

Generates debug messages for received OSPF DD packets.

`detail`

Generates detailed debug messages for received OSPF DD packets.

**send**
Generates debug messages for transmitted OSPF DD packets.

**detail**
Generates detailed debug messages for transmitted OSPF DD packets.

### Operational mode

Use this command to enable the generation of debug messages that are related to OSPF Database Description (DD) packets. OSPF DD packets provide a summary (digest) of each link-state advertisement in the link-state databases. OSPF routers exchange these packets to keep data synchronized.

## monitor protocol ospf enable packet detail

Enables the generation of debug messages that are related to OSPF detailed packets.

### Syntax:

```
monitor protocol ospf [ process pid ] enable packet detail
```

**process *pid***
Specifies the PID of an IPv4 OSPF process.

### Operational mode

Use this command to enable the generation of debug messages that are related to OSPF detailed packets.

## monitor protocol ospf enable packet hello

Enables the generation of debug messages that are related to OSPF hello packets.

### Syntax:

```
monitor protocol ospf [ process pid ] enable packet hello [ detail | recv [ detail ] | send [ detail ] ]
```

Debug messages are generated for OSPF hello packets at a medium level of detail.

**process *pid***
Specifies the PID of an IPv4 OSPF process.

**detail**
Generates detailed debug messages for OSPF hello packets, both sent and received.

**recv**
Generates debug messages for received OSPF hello packets.

**detail**
Generates detailed debug messages for received OSPF hello packets.

**send**
Generates debug messages for transmitted OSPF hello packets.

**detail**
Generates detailed debug messages for transmitted OSPF hello packets.

### Operational mode

Use this command to enable the generation of debug messages that are related to OSPF hello packets. OSPF hello packets are sent at intervals to discover neighbors and ensure that neighbors can be reached. Hello packets include information about certain OSPF timers, the designated router, the backup designated router (BDR), and known neighbors.

## monitor protocol ospf enable packet ls-ack

Enables the generation of debug messages that are related to OSPF LSA Ack packets.

### Syntax:

```
monitor protocol ospf [ process pid ] enable packet ls-ack [ detail | recv [ detail ] | send [ detail ]
]
```

Debug messages are generated for OSPF LS Ack packets at a medium level of detail.

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

**detail**
Generates detailed debug messages for OSPF LS Ack packets, both sent and received.

**recv**
Generates debug messages for received OSPF LS Ack packets.

**detail**
Generates detailed debug messages for received OSPF LS Ack packets.

**send**
Generates debug messages for transmitted OSPF LS Ack packets.

**detail**
Generates detailed debug messages for transmitted OSPF LS Ack packets.

**Operational mode**

Use this command to enable the generation of debug messages that are related to OSPF link-state acknowledgment (LSA) packets. LS Ack packets are sent to OSPF neighbors to acknowledge receipt of a link-state advertisement packet from a neighbor.

# monitor protocol ospf enable packet ls-request

Enables the generation of debug messages that are related to OSPF LSR packets.

**Syntax:**

```
monitor protocol ospf [ process pid ] enable packet ls-request [ detail | recv [ detail ] | send [
detail ] ]
```

Debug messages are generated for OSPF LSR packets at a medium level of detail.

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

**detail**
Generates detailed debug messages for OSPF LSR packets, both sent and received.

**recv**
Generates debug messages for received OSPF LSR packets.

**detail**
Generates detailed debug messages for received OSPF LSR packets.

**send**
Generates debug messages for transmitted OSPF LSR packets.

**detail**
Generates detailed debug messages for transmitted OSPF LSR packets.

**Operational mode**

Use this command to enable the generation of debug messages that are related to OSPF link-state request (LSR) packets. After exchanging Database Description packets, neighboring OSPF routers determine which link-state advertisements (LSAs) are missing from the local link-state database. The local router sends an LSR packet to the neighbor to request the missing LSAs.

# monitor protocol ospf enable packet ls-update

Enables the generation of debug messages that are related to OSPF LSU packets.

**Syntax:**

```
monitor protocol ospf [ process pid ] enable packet ls-update [ detail | recv [ detail ] | send [ detail
] ]
```

Debug messages are generated for OSPF LSU packets at a medium level of detail.

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

**detail**

Generates detailed debug messages for OSPF LSU packets, both sent and received.

**`recv`**
Generates debug messages for received OSPF LSU packets.

**`detail`**
Generates detailed debug messages for received OSPF LSU packets.

**`send`**
Generates debug messages for transmitted OSPF LSU packets.

**`detail`**
Generates detailed debug messages for transmitted OSPF LSU packets.

### Operational mode

Use this command to enable the generation of debug messages that are related to OSPF link-state update (LSU) packets. LSU packets send any required link-state advertisements updates to an OSPF neighbor.

## monitor protocol ospf enable packet recv

Enables the generation of debug messages that are related to received OSPF packets.

**Syntax:**

```
monitor protocol ospf [ process pid ] enable packet recv
```

**`process`** ***pid***
Specifies the PID of an IPv4 OSPF process.

### Operational mode

Use this command to enable the generation of debug messages that are related to received OSPF packets.

## monitor protocol ospf enable packet send

Enables the generation of debug messages that are related to OSPF sent packets.

**Syntax:**

```
monitor protocol ospf [ process pid ] enable packet send
```

**`process`** ***pid***
Specifies the PID of an IPv4 OSPF process.

### Operational mode

Use this command to enable the generation of debug messages that are related to OSPF sent packets.

## monitor protocol ospf enable rib

Enables the generation of debug messages that are related to OSPF RIB messages.

**Syntax:**

```
monitor protocol ospf [ process pid ] enable rib [ interface | redistribute ]
```

**`process`** ***pid***
Specifies the PID of an IPv4 OSPF process.

**`interface`**
Enables debugging of OSPF RIB interface messages.

**`redistribute`**
Enables debugging of OSPF RIB redistribute messages.

### Operational mode

Use this command to enable the generation of debug messages that are related to OSPF RIB messages.

## monitor protocol ospf enable route

Enables the generation of debug messages that are related to OSPF route calculations.

**Syntax:**

```
monitor protocol ospf [ process pid ] enable route [ ase | ia | install | spf ]
```

`process` ***pid***

Specifies the PID of an IPv4 OSPF process.

`ase`

Enables detailed debugging of OSPF external route calculations.

`ia`

Enables detailed debugging of OSPF interarea route calculations.

`install`

Enables detailed debugging of OSPF route installation.

`spf`

Enables detailed debugging of OSPF Shortest Path First calculation.

**Operational mode**

Use this command to enable the generation of debug messages that are related to OSPF route calculations.

## protocols ospf

Enables the OSPF routing protocol on the router.

**Syntax:**

```
set protocols ospf [ process pid ]
```

**Syntax:**

```
delete protocols ospf [ process pid ]
```

**Syntax:**

```
show protocols ospf
```

`process` ***pid***

Specifies the PID of an IPv4 OSPF process. The PID is a number that ranges from 1 through 65535.

**Configuration mode**

```
protocols {
       ospf {
              process pid
       }
}
```

Use this command to enable the Open Shortest Path First (OSPF) routing protocol on the system. You can enable multiple OSPF processes by assigning PIDs to these processes.

Use the `set` form of this command to enable the OSPF routing protocol.

Use the `delete` form of this command to disable the OSPF routing protocol and remove all OSPF configuration.

Use the `show` form of this command to display the configuration of the OSPF routing protocol.

## protocols ospf auto-cost reference-bandwidth <bandwidth>

Directs the system to use the reference-bandwidth method for calculating the administrative cost.

**Syntax:**

`set protocols ospf [` **`process`** *pid* `]` **`auto-cost reference-bandwidth`** *bandwidth*

**Syntax:**
`delete protocols ospf [` **`process`** *pid* `]` **`auto-cost reference-bandwidth`**

**Syntax:**
`show protocols ospf [` **`process`** *pid* `]` **`auto-cost reference-bandwidth`**

The default reference bandwidth is $10^8$.

**`process`** ***pid***
Specifies the PID of an IPv4 OSPF process.

***bandwidth***
The reference-bandwidth rate in megabits per second (Mbps). The rate ranges from 1 through 4294967.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                       auto-cost {
                              reference-bandwidth bandwidth
                       }
               }
       }
}
```

Use this command to set a reference-bandwidth rate for calculating the OSPF cost. The OSPF rate is calculated as the reference bandwidth divided by the actual bandwidth.

An explicitly set cost for an area overrides automatically calculated rates.

Use the `set` form of this command to set the reference-bandwidth rate for calculating the OSPF cost.

Use the `delete` form of this command to restore the default reference-bandwidth rate, which is $10^8$ Mbps, for calculating the OSPF cost.

Use the `show` form of this command to display the (current) reference-bandwidth rate for OSPF cost.

## protocols ospf default-information originate

Sets the characteristics of an external default route that is originated into an OSPF routing domain.

**Syntax:**
`set protocols ospf [` **`process`** *pid* `]` **`default-information originate`** `{` **`always`** `|` **`metric`** *metric* `|` **`metric-type`** *type* `|` **`route-map`** *map-name* `}`

**Syntax:**
`delete protocols ospf [` **`process`** *pid* `]` **`default-information originate`** `[` **`always`** `|` **`metric`** `|` **`metric-type`** `|` **`route-map`** `]`

**Syntax:**
`show protocols ospf [` **`process`** *pid* `]` **`default-information originate`** `[` **`always`** `|` **`metric`** `|` **`metric-type`** `|` **`route-map`** `]`

By default, the system does not generate an external default route into the OSPF routing domain. When enabled to do so, the default route depends on the type of area into which the default route is being advertised.

- In stub areas, a Type 3 link-state advertisement is generated with a metric of 1 and the metric type is ignored.
- In not-so-stubby areas (NSSAs) that are configured to import summary advertisements, a Type 7 LSA with a metric of 1 and a metric type of 2 is generated.

- In NSSAs that are configured not to import summary advertisements, a Type 3 LSA with metric of 1 and the metric type is ignored.
- In a totally stub area, only a default route is advertised and allowed into the area. Summary and external routes are not advertised in the area.

**`process`** ***pid***

Specifies the PID of an IPv4 OSPF process.

**`always`**

Always advertises the default route.

**`metric`** ***metric***

Specifies the metric to apply to the default route. The metric ranges from 0 through 16777214.

**`metric-type`** ***type***

The metric type of an external route, which specifies how cost is calculated, to associate with the Type 5 default link-state advertisement (LSA). The metric type is either of the following:

1: Adds internal cost to external cost. (E1 routes use the redistributed cost plus the cost to the ASBR.)

2: Uses only external cost. (E2 routes use only the redistributed cost.) In case of two ASBRs advertising the same E2 cost, the internal cost to ASBR is used to break the tie.

The default type is 2.

***map-name***

The default route is generated if the specified route map is satisfied.

### Configuration mode

```
protocols {
 ospf {
  default-information {
   originate {
    always
    metric metric
    metric-type type
    route-map map-name
   }
  }
 }
}
```

Use this command to redistribute the default route (0.0.0.0) into an OSPF routing domain.

If routes are redistributed in this way, the router automatically becomes an Autonomous System Boundary Router (ASBR). The router must have a default route configured before it can generate one, unless the always keyword is specified.

Use the `set` form of this command to enable generation of an external default route into the OSPF routing domain.

Use the `delete` form of this command to disable generation of an external default route into the OSPF routing domain or to restore default parameter values.

Use the `show` form of this command to display default route distribution configuration. Use this command to redistribute the default route (0.0.0.0) into an OSPF routing domain.

> **Note:** When extended ACLs are used directly or in a route map, the route must be specified in the source field of the ACL and the net mask of the route must be specified in the destination field of the ACL. The inverse mask of the source field is applied to the prefix prior to the comparison with the source. The inverse mask for the destination field is applied to the net mask of the prefix prior to the comparison with the net-mask as specified in the destination field. Effectively, the prefix must match the source with the application of inverse mask of the source field. And the net mask for the prefix must match the destination with the application of the inverse mask of the destination field.

# protocols ospf default-metric <metric>

Sets default metric to be applied to routes being redistributed into OSPF.

**Syntax:**
set protocols ospf [ **process** *pid* ] **default-metric** *metric*

**Syntax:**
delete protocols ospf [ **process** *pid* ] **default-metric** *metric*

**Syntax:**
show protocols ospf [ **process** *pid* ] **default-metric**

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

***metric***
The metric to apply to routes from other protocols that are redistributed into OSPF. The metric ranges from 1 through 16777214.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                      default-metric metric
               }
       }
}
```

Use this command to set the default metric to apply to routes from other protocols that are redistributed into OSPF.

Use the `set` form of this command to set the default OSPF metric.

Use the `delete` form of this command to restore the default value for default metric.

Use the `show` form of this command to display the default OSPF metric.

# protocols ospf distance

Sets the OSPF administrative distance by route type.

**Syntax:**
set protocols ospf [ **process** *pid* ] **distance** { **global** *global* | **ospf** [ **external** *external* | **inter-area** *inter* | **intra-area** *intra* ] }

**Syntax:**
delete protocols ospf [ **process** *pid* ] **distance** [ **global** | **ospf** [ **external** | **inter-area** | **intra-area** ] ]

**Syntax:**
show protocols ospf [ **process** *pid* ] **distance** [ **global** | **ospf** [ **external** | **inter-area** | **intra-area** ] ]

The default administrative distance for OSPF routes is 110.

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

**global** ***global***
Sets the administrative distance for all routes. The distance ranges from 1 through 255.

**external** ***external***

Sets the OSPF administrative distance for external routes (routes learned from another protocol by redistribution). The distance ranges from 1 through 255. The default distance is 110.

`inter-area` ***inter***

Sets the OSPF administrative distance for interarea routes (routes to another area). The distance ranges from 1 through 255. The default is distance 110.

`intra-area` ***intra***

Sets the OSPF administrative distance for intra-area routes (routes within an area). The distance ranges from 1 through 255. The default distance is 110.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                      distance {
                              global global
                              ospf {
                                     external external
                                     inter-area inter
                                     intra-area intra
                              }
                      }
               }
       }
}
```

Use this command to set the administrative distance for OSPF routes.

The administrative distance indicates the trustworthiness of a router or group of routers as a source of routing information. In general, as the distance increases, the entity becomes less trusted. An administrative distance of 1 usually represents a directly connected network, and an administrative distance of 255 indicates that the routing source is unreliable or unknown. The administrative distance that is conventionally applied to OSPF is 110.

Use the `set` form of this command to set the administrative distance.

Use the `delete` form of this command to restore the default administrative distance, which is 110.

Use the `show` form of this command to display the administrative distance.

## protocols ospf distribute-list <list-number>

Specifies an access list to filter networks in routing updates.

**Syntax:**

`set protocols ospf [ process` *pid* `] distribute-list` *list-number* `[ export` *type* `| import ]`

**Syntax:**

`delete protocols ospf [ process` *pid* `] distribute-list` *list-number* `[ export` *type* `| import ]`

**Syntax:**

`show protocols ospf [ process` *pid* `] distribute-list` *list-number*

`process` ***pid***

Specifies the PID of an IPv4 OSPF process.

***list-number***

The number of an access list to filter networks in routing updates. The number is a number from one of the following ranges of numbers:

1-99: IP standard access list.

100-199: IP extended access list.

1300-1999: IP standard access list (expanded range).

2000-2699: IP extended access list (expanded range).

**`export`** ***type***

Specifies the type of export routes to filter. The type is one of the following: `bgp`, `connected`, `kernel`, `rip`, or `static`. Multiple types can be specified by creating additional `export` configuration nodes.

**`import`**

The filter for incoming routing updates.

### Configuration mode

```
protocols {
       ospf {
               process pid {
                     distribute-list list-number {
                            export type
                            import
                     }
               }
       }
}
```

Use this command to specify an access list to filter networks in routing updates.

> **Note:** When extended ACLs are used directly or in a route map, the route must be specified in the source field of the ACL and the net mask of the route must be specified in the destination field of the ACL.
>
> The inverse mask of the source field is applied to the prefix prior to the comparison with the source. The inverse mask for the destination field is applied to the net mask of the prefix prior to the comparison with the net-mask as specified in the destination field.
>
> Effectively, the prefix must match the source with the application of inverse mask of the source field. And the net mask for the prefix must match the destination with the application of the inverse mask of the destination field.

Use the `set` form of this command to specify an access list to filter networks in routing updates.

Use the `delete` form of this command to remove an access list from filtering networks in routing updates.

Use the `show` form of this command to display the configuration.

## protocols ospf log

Enables OSPF protocol logs.

### Syntax:

`set protocols ospf` [ **`process`** *pid* ] **`log`** { **`all`** | **`bfd`** | **`database-timer`** | **`events`** | **`ifsm`** | **`lsa`** | **`nfsm`** | **`nsm`** | **`packet`** | **`rib`** | **`route`** }

### Syntax:

`delete protocols ospf` [ **`process`** *pid* ] **`log`** { **`all`** | **`bfd`** | **`database-timer`** | **`events`** | **`ifsm`** | **`lsa`** | **`nfsm`** | **`nsm`** | **`packet`** | **`rib`** | **`route`** }

### Syntax:

`show protocols ospf` [ **`process`** *pid* ] **`log`** { **`all`** | **`bfd`** | **`database-timer`** | **`events`** | **`ifsm`** | **`lsa`** | **`nfsm`** | **`nsm`** | **`packet`** | **`rib`** | **`route`** }

None

**`process`** ***pid***

Specifies the PID of an IPv4 OSPF process.

**`all`**

Enables all OSPF logs.

**`bfd`**

Enables only OSPF BFD logs.

**`database-timer`**

Enables only OSPF database-timer logs.

**`events`**

Enables only OSPF event logs.

**`ifsm`**

Enables only OSPF IFSM logs.

**`lsa`**

Enables only OSPF LSA logs.

**`nfsm`**

Enables only OSPF NFSM logs.

**`nsm`**

Enables only OSPF NSM logs.

**`packet`**

Enables only OSPF packet logs.

**`rib`**

Enables only OSPF RIB logs.

**`route`**

Enables only OSPF route logs.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                       log {
                               all
                               bfd
                               database-timer
                               events
                               ifsm
                               lsa
                               nfsm
                               nsm
                               packet
                               rib
                               route
                       }
               }
       }
}
```

Use the `set` form of this command to enable Open Shortest Path First (OSPF) logs.

Use the `delete` form of this command to remove OSPF logs.

Use the `show` form of this command to view OSPF logs.

# protocols ospf log events

Enables OSPF event logs.

**Syntax:**

```
set protocols ospf [ process pid ] log events { abr | all | asbr | lsa | nssa | os | router | vlink }
```

**Syntax:**

```
delete protocols ospf [ process pid ] log ifsm { abr | all | asbr | lsa | nssa | os | router | vlink }
```

**Syntax:**

```
show protocols ospf [ process pid ] log ifsm { abr | all | asbr | lsa | nssa | os | router | vlink }
```

None

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

**abr**
Enables OSPF event ABR logs.

**all**
Enables all OSPF event logs

**asbr**
Enables only OSPF event ASBR logs.

**lsa**
Enables only OSPF event LSA logs.

**nssa**
Enables only OSPF event NSSA logs.

**os**
Enables only OSPF event OS logs.

**router**
Enables only OSPF event router logs.

**vlink**
Enables only OSPF event vlink logs.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                       log {
                               events {
                                       abr
                                       all
                                       asbr
                                       lsa
                                       nssa
                                       os
                                       router
                                       vlink
                               }
                       }
               }
       }
}
```

Use the `set` form of this command to enable Open Shortest Path First (OSPF) event logs.

Use the `delete` form of this command to remove OSPF event logs.

Use the `show` form of this command to view OSPF event logs.

# protocols ospf log ifsm

Enables OSPF IFSM logs.

**Syntax:**

```
set protocols ospf [ process pid ] log ifsm { all | events | status | timers }
```

**Syntax:**

```
delete protocols ospf [ process pid ] log ifsm { all | events | status | timers }
```

**Syntax:**

```
show protocols ospf [ process pid ] log ifsm { all | events | status | timers }
```

None

`process` ***pid***
: Specifies the PID of an IPv4 OSPF process.

**`all`**
: Enables all OSPF IFSM logs.

**`events`**
: Enables only OSPF IFSM event logs.

**`status`**
: Enables only OSPF IFSM status logs.

**`timers`**
: Enables only OSPF IFSM timer logs.

**Configuration mode**

```
protocols {
        ospf {
                process pid {
                        log {
                                ifsm {
                                        all
                                        events
                                        status
                                        timers
                                }
                        }
                }
        }
}
```

Use the `set` form of this command to enable Open Shortest Path First (OSPF) interface finite-state machine (IFSM) logs.

Use the `delete` form of this command to remove OSPF IFSM logs.

Use the `show` form of this command to view OSPF IFSM logs.

# protocols ospf log lsa

Enables OSPF LSA logs.

**Syntax:**
`set protocols ospf` [ **`process`** *pid* ] **`log lsa`** { **`all`** | **`flooding`** | **`generate`** | **`install`** | **`maxage`** | **`refresh`** }

**Syntax:**
`delete protocols ospf` [ **`process`** *pid* ] **`log lsa`** { **`all`** | **`flooding`** | **`generate`** | **`install`** | **`maxage`** | **`refresh`** }

**Syntax:**
`show protocols ospf` [ **`process`** *pid* ] **`log lsa`** { **`all`** | **`flooding`** | **`generate`** | **`install`** | **`maxage`** | **`refresh`** }

None

`process` ***pid***
: Specifies the PID of an IPv4 OSPF process.

**`all`**
: Enables all OSPF LSA logs.

**`flooding`**
: Enables only OSPF LSA flooding logs.

**`generate`**
: Enables only OSPF LSA generation logs.

**`install`**
: Enables only OSPF LSA installation logs.

**maxage**

Enables only OSPF LSA maximum age logs.

**refresh**

Enables only OSPF LSA refresh logs.

**Configuration mode**

```
protocols {
        ospf {
                process pid {
                        log {
                                lsa {
                                        all
                                        flooding
                                        generate
                                        install
                                        zmaxage
                                        refresh
                                }
                        }
                }
        }
}
```

Use the `set` form of this command to enable Open Shortest Path First (OSPF) LSA logs.

Use the `delete` form of this command to remove OSPF LSA logs.

Use the `show` form of this command to view OSPF LSA logs.

## protocols ospf log nfsm

Enables OSPF NFSM logs.

**Syntax:**

set protocols ospf [ **process** *pid* ] **log nfsm** { **all** | **events** | **status** | **timers** }

**Syntax:**

delete protocols ospf [ **process** *pid* ] **log nfsm** { **all** | **events** | **status** | **timers** }

**Syntax:**

show protocols ospf [ **process** *pid* ] **log nfsm** { **all** | **events** | **status** | **timers** }

None

**process** ***pid***

Specifies the PID of an IPv4 OSPF process.

**all**

Enables all OSPF NFSM logs.

**events**

Enables only OSPF NFSM event logs.

**status**

Enables only OSPF NFSM status logs.

**timers**

Enables only OSPF NFSM timers logs.

**Configuration mode**

```
protocols {
        ospf {
                process pid {
                        log {
                                nfsm {
                                        all
                                        events
                                        status
                                        timers
                                }
                        }
                }
        }
}
```

Use the `set` form of this command to enable Open Shortest Path First (OSPF) neighbor FSM (NFSM) logs.

Use the `delete` form of this command to remove OSPF NFSM logs.

Use the `show` form of this command to view OSPF NFSM logs.

## protocols ospf log nsm

Enables OSPF NSM logs.

**Syntax:**

set protocols ospf [ **process** *pid* ] **log nsm** { **all** | **interface** | **redistribute** }

**Syntax:**

delete protocols ospf [ **process** *pid* ] **log nsm** { **all** | **interface** | **redistribute** }

**Syntax:**

show protocols ospf [ **process** *pid* ] **log nsm** { **all** | **interface** | **redistribute** }

None

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

**all**
Enables all OSPF NSM logs.

**interface**
Enables only OSPF interface logs.

**redistribute**
Enables only OSPF redistribute logs.

**Configuration mode**

```
protocols {
        ospf {
                process pid {
                        log {
                                nsm {
                                        all
                                        interface
                                        redistribute
                                }
                        }
                }
        }
}
```

Use the `set` form of this command to enable OSPF NSM log.

Use the `delete` form of this command to remove an OSPF NSM log.

Use the `show` form of this command to remove an OSPF NSM log.

# protocols ospf log packet

Enables OSPF packet logs.

**Syntax:**

set protocols ospf [ **process** *pid* ] **log packet** { **all** | **dd** | **detail** | **hello** | **ls-ack** | **ls-request** | **ls-update** | **recv** | **send** }

**Syntax:**

delete protocols ospf [ **process** *pid* ] **log packet** { **all** | **dd** | **detail** | **hello** | **ls-ack** | **ls-request** | **ls-update** | **recv** | **send** }

**Syntax:**

show protocols ospf [ **process** *pid* ] **log packet** { **all** | **dd** | **detail** | **hello** | **ls-ack** | **ls-request** | **ls-update** | **recv** | **send** }

None

**process** ***pid***
: Specifies the PID of an IPv4 OSPF process.

**all**
: Enables all OSPF packet logs.

**dd**
: Enables only OSPF packet DD logs.

**detail**
: Enables only OSPF packet detail logs.

**hello**
: Enables only OSPF packet hello logs.

**ls-ack**
: Enables only OSPF packet ls-ack logs.

**ls-request**
: Enables only OSPF packet ls-request logs.

**ls-update**
: Enables only OSPF packet ls-update logs.

**recv**
: Enables only OSPF packet receive logs.

**send**
: Enables only OSPF packet send logs.

### Configuration mode

```
protocols {
       ospf {
               process pid {
                       log {
                               packet {
                                       all
                                       dd
                                       detail
                                       hello
                                       ls-ack
                                       ls-request
                                       ls-update
                                       recv
                                       send
                               }
                       }
               }
       }
}
```

Use the `set` form of this command to enable Open Shortest Path First (OSPF) packet logs.

Use the `delete` form of this command to remove OSPF packet logs.

Use the `show` form of this command to view OSPF packet logs.

# protocols ospf log rib

Enables OSPF RIB logs.

**Syntax:**

```
set protocols ospf log rib [ process pid ] log rib { all | interface | redistribute }
```

**Syntax:**

```
delete protocols ospf log rib [ process pid ] log rib { all | interface | redistribute }
```

**Syntax:**

```
show protocols ospf log rib [ process pid ] log rib { all | interface | redistribute }
```

None

`process` ***pid***
: Specifies the PID of an IPv4 OSPF process.

**`all`**
: Enables all OSPF RIB logs.

**`interface`**
: Enables only OSPF RIB interface logs.

**`redistribute`**
: Enables only OSPF RIB redistribute logs.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                       log {
                               rib {
                                       all
                                       interface
                                       redistribute
                               }
                       }
               }
       }
}
```

Use the `set` form of this command to enable OSPF RIB logs.

Use the `delete` form of this command to remove OSPF RIB logs.

Use the `show` form of this command to view OSPF RIB logs.

# protocols ospf log route

Enables OSPF route logs.

**Syntax:**

```
set protocols ospf [ process pid ] log route { all | ase | ia | install | spf }
```

**Syntax:**

delete protocols ospf [ **process** *pid* ] **log route** { **all** | **ase** | **ia** | **install** | **spf** }

**Syntax:**
show protocols ospf [ **process** *pid* ] **log route** { **all** | **ase** | **ia** | **install** | **spf** }

None

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

**all**
Enables all OSPF route all logs.

**ase**
Enables only OSPF route ASE logs.

**ia**
Enables only OSPF route IA logs.

**install**
Enables only OSPF route install logs.

**SPF**
Enables only OSPF route SPF logs.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                       log {
                               route {
                                       all
                                       ase
                                       ia
                                       install
                                       spf
                               }
                       }
               }
       }
}
```

Use the `set` form of this command to enable OSPF route logs.

Use the `delete` form of this command to remove OSPF route logs.

Use the `show` form of this command to view OSPF route logs.

# protocols ospf neighbor <ipv4>

Creates an OSPF neighbor and sets or modifies its characteristics.

**Syntax:**
set protocols ospf [ **process** *pid* ] **neighbor** *ipv4* { **poll-interval** *interval* | **priority** *priority* }

**Syntax:**
delete protocols ospf [ **process** *pid* ] **neighbor** *ipv4* [ **poll-interval** | **priority** ]

**Syntax:**
show protocols ospf [ **process** *pid* ] **neighbor** *ipv4* [ **poll-interval** | **priority** ]

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

***ipv4***
The IPv4 address of the OSPF neighbor that is being created.

**poll-interval** ***interval***

Sets the interval, in seconds, at which the specified neighbor is polled to determine whether it can still be reached. The interval ranges from 1 through 65535. The default interval is 120.

**priority** ***priority***

Sets the priority of the specified neighbor. The number for the priority ranges from 0 through 255, where the lower the number, the higher the priority. The default number is 1.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                       neighbor ipv4 {
                               poll-interval interval
                               priority priority
                       }
               }
       }
}
```

Use this command to create an OSPF neighbor and set or modify its characteristics.

Use the `set` form of this command to create an OSPF neighbor or modify its characteristics.

Use the `delete` form of this command to remove an OSPF neighbor or reset neighbor characteristics to default values.

Use the `show` form of this command to display characteristics of an OSPF neighbor.

# protocols ospf overflow database external limit <number-of-as-external-lsas> recovery <recovery-time>

Sets the maximum number of AS-External-LSAs that are allowed in the OSPF database.

**Syntax:**

set protocols ospf [ **process** *pid* ] **overflow database external limit** *number-of-as-external-lsas* **recovery** *recovery-time*

**Syntax:**

delete protocols ospf [ **process** *pid* ] **overflow database external limit** *number-of-as-external-lsas* **recovery** [ *recovery-time* ]

**Syntax:**

show protocols ospf [ **process** *pid* ] **overflow database external limit** *number-of-as-external-lsas* **recovery**

**process** ***pid***

Specifies the PID of an IPv4 OSPF process.

***number-of-as-external-lsas***

The maximum number of AS-External-LSAs that are allowed in the OSPF database. The number ranges from 0 through 2147483647.

***recovery-time***

The number of seconds that the router waits before exiting the overflow state. The number of seconds ranges from 0 through 65535. A number of 0 indicates that there is no recovery and the router stays in the overflow state until an administrator intervenes.

**Configuration mode**

```
protocols {
      ospf {
              process pid {
                      overflow {
                              database {
                                      external {
                                              limit number-of-as-external-lsas {
                                                      recovery recovery-time
                                              }
                                      }
                              }
                      }
              }
      }
}
```

Use this command to limit the number of external LSAs in the database when the router is in a wait state. It can be used to protect the router from excessive memory usage.

Use the `set` form of this command to specify the maximum number of AS-External-LSAs that are allowed in the OSPF database.

Use the `delete` form of this command to remove the limit on the number of AS-External-LSAs that are allowed in the OSPF database.

Use the `show` form of this command to display the current recovery time and maximum number of AS-External-LSAs that are allowed in the database.

# protocols ospf overflow database hard limit <number-of-lsas>

Sets a maximum number of LSAs that are allowed in the OSPF database and shuts down any instance that exceeds the limit.

**Syntax:**
`set protocols ospf` [ **`process`** *pid* ] **`overflow database hard limit`** *number-of-lsas*

**Syntax:**
`delete protocols ospf` [ **`process`** *pid* ] **`overflow database hard limit`** [ *number-of-lsas* ]

**Syntax:**
`show protocols ospf` [ **`process`** *pid* ] **`overflow database hard limit`**

**`process`** ***pid***
Specifies the PID of an IPv4 OSPF process.

***number-of-lsas***
The maximum number of LSAs that are allowed in the database. The number ranges from 0 through 4294967294.

**Configuration mode**

```
protocols {
      ospf {
              process pid {
                      overflow {
                              database {
                                      hard {
                                              limit number-of-lsas
                                      }
                              }
                      }
              }
      }
}
```

Use this command to limit the number of link-state acknowledgments (LSAs) that are allowed in the OSPF database and shuts down any instance that exceeds the specified limit.

Use the `set` form of this command to set the maximum number of LSAs that are allowed in the OSPF database.

Use the `delete` form of this command to remove the maximum number of LSAs that are supported in the OSPF database.

Use the `show` form of this command to display the current maximum number of LSAs that are allowed in the OSPF database.

# protocols ospf overflow database soft limit

Sets a maximum number of LSAs that are allowed in the OSPF database and sends a warning if the limit is exceeded.

**Syntax:**
`set protocols ospf` [ **`process`** *pid* ] **`overflow database soft limit`** *number-of-lsas*

**Syntax:**
`delete protocols ospf` [ **`process`** *pid* ] **`overflow database soft limit`** [ *number-of-lsas* ]

**Syntax:**
`show protocols ospf` [ **`process`** *pid* ] **`overflow database soft limit`**

**`process`** ***pid***
Specifies the PID of an IPv4 OSPF process.

***number-of-lsas***
The maximum number of LSAs that are allowed in the database. The number ranges from 0 through 4294967294.

**Configuration mode**

```
protocols {
      ospf {
              process pid {
                      overflow {
                              database {
                                      soft {
                                              limit number-of-lsas
                                      }
                              }
                      }
              }
      }
}
```

Use the `set` form of this command to set the maximum number of LSAs that are allowed in the OSPF database and send a warning if the limit is exceeded.

Use the `delete` form of this command to remove the maximum number of LSAs that are supported in the OSPF database.

Use the `show` form of this command to display the current maximum number of LSAs that are allowed in the OSPF database.

# protocols ospf parameters

Sets global OSPF parameters.

**Syntax:**
`set protocols ospf` [ **`process`** *pid* ] **`parameters`** { **`opaque-lsa`** | **`rfc1583-compatibility`** | **`router-id`** *ipv4* }

**Syntax:**

`delete protocols ospf [ `**`process`**` `*`pid`*` ] `**`parameters`**` [ `**`opaque-lsa`**` | `**`rfc1583-compatibility`**` | `**`router-id`**` ]`

**Syntax:**
`show protocols ospf [ `**`process`**` `*`pid`*` ] `**`parameters`**

By default, support for opaque LSAs is disabled. By default, RFC 1583 support is disabled.

If no router ID is explicitly configured, the OSPF process calculates an ID for the router by using the following algorithm.

1. Use the IP address of the loopback interface.
2. Use the highest IP address of the address on router interfaces.
3. If no interfaces are defined, use 0.0.0.0.

**`process`** ***pid***
Specifies the PID of an IPv4 OSPF process.

**`opaque-lsa`**
Enables support for opaque LSAs, as described in RFC 2370.

**`rfc1583-compatibility`**
Indicates whether the handling of AS external routes complies with RFC 1583.

OSPF RFCs that are subsequent to RFC 1583 enhance the way external route calculations are performed. This option controls the preference rules that are used in external route calculations when choosing among multiple AS-external-LSAs that advertise the same destination.

When set to “enabled,” the preference rules remain those specified by RFC 1583. When set to “disabled,” the preference rules are those stated in later RFCs, which prevent routing loops when AS-external-LSAs for the same destination have been originated from different areas.

To minimize the chance of routing loops, all OSPF routers in an OSPF routing domain should have `rfc1583-compatibility` set identically. When there are routers present that have not been updated with the functionality that is specified in later RFCs, all routers should have `rfc1583-compatibility` enabled. Otherwise, all routers should have `rfc1583-compatibility` disabled, preventing all routing loops. The default setting is disabled.

**`router id`** ***ipv4***
Sets an explicit router identifier, overriding the router ID that is calculated by the OSPF process. The format of the identifier is an IPv4 address.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                       parameters {
                               abr-type type
                               opaque-lsa
                               rfc1583-compatibility
                               router-id ipv4
                       }
               }
       }
}
```

Use this command to set global OSPF parameters.

**Note:** Modifying the router ID causes the router to restart.

Use the `set` form of this command to set global OSPF parameters.

Use the `delete` form of this command to restore the default settings of global OSPF parameters.

Use the `show` form of this command to display the (current) global OSPF parameters.

## protocols ospf passive-interface

Suppresses OSPF routing updates on an interface.

**Syntax:**
set protocols ospf [ **process** *pid* ] **passive-interface** *interface*

**Syntax:**
delete protocols ospf [ **process** *pid* ] **passive-interface** [ *interface* ]

**Syntax:**
show protocols ospf [ **process** *pid* ] **passive-interface**

Routing updates are not suppressed.

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

***interface***
The identifier of an interface. Supported interface types are:

- Data plane
- Loopback

For more information about these interface types, refer to Supported Interface Types *(page 90)*.

You can suppress routing updates on multiple interfaces by creating multiple **passive-interface** configuration nodes.

**default**
Suppresses routing updates on all interfaces.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                       passive-interface interface

               }
       }
}
```

Use this command to suppress OSPF routing updates on an interface. OSPF traffic can be received on the interface, but the traffic is not sent on the interface.

Use the `set` form of this command to suppress OSPF routing updates on an interface.

Use the `delete` form of this command to remove the suppression of OSPF routing updates on an interface.

Use the `show` form of this command to display the suppression configuration of OSPF routing updates.

## protocols ospf passive-interface-exclude

Excludes the suppression of OSPF routing updates on an interface.

**Syntax:**
set protocols ospf [ **process** *pid* ] **passive-interface-exclude** *interface*

**Syntax:**
delete protocols ospf [ **process** *pid* ] **passive-interface-exclude** *interface*

**Syntax:**

`show protocols ospf [` **`process`** *pid* `]` **`passive-interface-exclude`**

Routing updates are not suppressed and  passive-interface is configured.

**`process`** ***pid***
: Specifies the PID of an IPv4 OSPF process.

***interface***
: The identifier of an interface. Supported interface types are:

- Data plane
- Loopback

For more information about these interface types, refer to Supported Interface Types *(page  90)*.

You can exclude the suppression of routing updates on multiple interfaces by creating multiple **passive-interface-exclude** configuration nodes.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                       passive-interface-exclude interface
               }
       }
}
```

Use this command to exclude the suppression of OSPF routing updates on an interface. OSPF traffic can be neither received on the interface nor sent through the interface.

Use the `set` form of this command to exclude the suppression of OSPF routing updates on an interface.

Use the `delete` form of this command to restore the default behavior, that is, routing updates are not suppressed.

Use the `show` form of this command to display the suppression configuration of OSPF routing updates.

The following example shows how to exclude the suppression of OSPF routing update suppression on the 10.11.121.134 interface.

```
vyatta@Rn# set protocols ospf passive-interface-exclude 10.11.121.134
vyatta@R6# show protocols ospf
ospf {
      passive-interface-exclude 10.11.121.134
}
```

# protocols ospf redistribute bgp

Sets the parameters for redistribution of BGP routes into OSPF.

**Syntax:**

`set protocols ospf [` **`process`** *pid* `]` **`redistribute bgp`** `{` **`metric`** *metric* `|` **`metric-type`** *type* `|` **`route-map`** *map-name* `}`

**Syntax:**

`delete protocols ospf [` **`process`** *pid* `]` **`redistribute bgp`** `[` **`metric`** `|` **`metric-type`** `|` **`route-map`** `]`

**Syntax:**

`show protocols ospf [` **`process`** *pid* `]` **`redistribute bgp`** `[` **`metric`** `|` **`metric-type`** `|` **`route-map`** `]`

BGP routes that are redistributed into OSPF are assigned a routing metric of 1 and a metric type of 2. By default, no route map is applied to redistributed BGP routes.

`process` ***pid***

Specifies the PID of an IPv4 OSPF process.

`metric` ***metric***

Applies the specified metric to BGP routes that are redistributed into OSPF. The metric ranges from 1 through 16. The default metric is 1.

`metric-type` ***type***

The external route metric type, that specifies how cost is calculated, to be associated with the Type 5 default link-state advertisement (LSA). Supported values are:

1: Internal cost is added to external cost. (E1 routes use the redistributed cost plus the cost to the autonomous system boundary router (ASBR).

2: External cost only. (E2 routes use only the redistributed cost.)

The default is 2.

`route-map` ***map-name***

Redistributes routes that satisfy the specified route map.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                     redistribute {
                            bgp {
                                   metric metric
                                   metric-type type
                                   route-map map-name
                            }
                     }
               }
       }
}
```

Use this command to set the parameters for redistribution of BGP routes into OSPF.

> **Note:** When extended ACLs are used directly or in a route map, the route must be specified in the source field of the ACL and the net mask of the route must be specified in the destination field of the ACL.
>
> The inverse mask of the source field is applied to the prefix prior to the comparison with the source. The inverse mask for the destination field is applied to the net mask of the prefix prior to the comparison with the net-mask as specified in the destination field.
>
> Effectively, the prefix must match the source with the application of inverse mask of the source field. And the net mask for the prefix must match the destination with the application of the inverse mask of the destination field.

Use the `set` form of this command to set the parameters for redistribution of BGP routes.

Use the `delete` form of this command to remove the parameters for redistribution of BGP routes.

Use the `show` form of this command to display the parameters for redistribution of BGP routes.

# protocols ospf redistribute connected

Sets the parameters for redistribution of connected routes into OSPF.

**Syntax:**

`set protocols ospf` [ **`process`** *pid* ] **`redistribute connected`** { **`metric`** *metric* | **`metric-type`** *type* | **`route-map`** *map-name* }

**Syntax:**

`delete protocols ospf` [ **`process`** *pid* ] **`redistribute connected`** [ **`metric`** | **`metric-type`** | **`route-map`** ]

**Syntax:**
`show protocols ospf` [ **`process`** *pid* ] **`redistribute connected`** [ **`metric`** | **`metric-type`** | **`route-map`** ]

Connected routes that are redistributed into OSPF are assigned a routing metric of 1 and a metric type of 2. By default, no route map is applied to redistributed connected routes.

`process` ***pid***
Specifies the PID of an IPv4 OSPF process.

`metric` ***metric***
Applies the specified metric to connected routes that are redistributed into OSPF. The metric ranges from 1 through 16. The default metric is 1.

`metric-type` ***type***
The metric type of an external route, which specifies how cost is calculated, to associate with the Type 5 default link-state advertisement (LSA). The metric type is either of the following:

1: Adds internal cost to external cost. (E1 routes use the redistributed cost plus the cost to the ASBR.)

2: Uses only external cost. (E2 routes use only the redistributed cost.)

The default type is 2.

`route-map` ***map-name***
Redistributes routes that satisfy the specified route map.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                     redistribute {
                            connected {
                                   metric metric
                                   metric-type type
                                   route-map map-name
                            }
                     }
               }
       }
}
```

Use this command to set the parameters for redistribution of connected routes into OSPF.

> **Note:** When extended ACLs are used directly or in a route map, the route must be specified in the source field of the ACL and the net mask of the route must be specified in the destination field of the ACL.
>
> The inverse mask of the source field is applied to the prefix prior to the comparison with the source. The inverse mask for the destination field is applied to the net mask of the prefix prior to the comparison with the net-mask as specified in the destination field.
>
> Effectively, the prefix must match the source with the application of inverse mask of the source field. And the net mask for the prefix must match the destination with the application of the inverse mask of the destination field.

Use the `set` form of this command to set the parameters for redistribution of connected routes.

Use the `delete` form of this command to remove the parameters for redistribution of connected routes.

Use the `show` form of this command to display the parameters for redistribution of connected routes.

## protocols ospf redistribute kernel

Sets the parameters for redistribution of kernel routes into OSPF.

**Syntax:**
set protocols ospf [ **process** *pid* ] **redistribute kernel** { **metric** *metric* | **metric-type** *type* | **route-map** *map-name* }

**Syntax:**
delete protocols ospf [ **process** *pid* ] **redistribute kernel** [ **metric** | **metric-type** | **route-map** ]

**Syntax:**
show protocols ospf [ **process** *pid* ] **redistribute kernel** [ **metric** | **metric-type** | **route-map** ]

Kernel routes that are redistributed into OSPF are assigned a routing metric of 1 and a metric type of 2. By default, no route map is applied to redistributed kernel routes.

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

**metric** ***metric***
Applies the specified metric to kernel routes that are redistributed into OSPF. The metric ranges from 1 through 16. The default metric is 1.

**metric-type** ***type***
The metric type of an external route, which specifies how cost is calculated, to associate with the Type 5 default link-state advertisement (LSA). The metric type is either of the following:

1: Adds internal cost to external cost. (E1 routes use the redistributed cost plus the cost to the ASBR.)

2: Uses only external cost. (E2 routes use only the redistributed cost.)

The default type is 2.

**route-map** ***map-name***
Redistributes routes that satisfy the specified route map.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                     redistribute {
                            kernel {
                                   metric metric
                                   metric-type type
                                   route-map map-name
                            }
                     }
               }
       }
}
```

Use this command to set the parameters for redistribution of kernel routes into OSPF.

> **Note:** When extended ACLs are used directly or in a route map, the route must be specified in the source field of the ACL and the net mask of the route must be specified in the destination field of the ACL.
>
> The inverse mask of the source field is applied to the prefix prior to the comparison with the source. The inverse mask for the destination field is applied to the net mask of the prefix prior to the comparison with the net-mask as specified in the destination field.
>
> Effectively, the prefix must match the source with the application of inverse mask of the source field. And the net mask for the prefix must match the destination with the application of the inverse mask of the destination field.

Use the `set` form of this command to set the parameters for redistribution of kernel routes.

Use the `delete` form of this command to remove the parameters for redistribution of kernel routes.

Use the `show` form of this command to display the parameters for redistribution of kernel routes.

## protocols ospf redistribute <protocol> <options>

Sets the parameters for redistribution of routes based on the protocol value into OSPF. Protocol extends to bgp, connected, static, rip, and kernel. Options extends to metric, metric-type and route_map

**Syntax:**

`set protocols ospf` [ **`process`** *pid* ] **`redistribute`** *protocol* { **`metric`** *metric* | **`metric-type`** *type* | **`route-map`** *map-name* | **`tag`** *value* }

**Syntax:**

`delete protocols ospf` [ **`process`** *pid* ] **`redistribute`** *protocol* [ **`metric`** | **`metric-type`** | **`route-map`** | **`tag`** ]

**Syntax:**

`show protocols ospf` [ **`process`** *pid* ] **`redistribute`** *protocol* [ **`metric`** | **`metric-type`** | **`route-map`** | **`tag`** ]

**Syntax:**

All IGP routes redistributed into OSPF are assigned a default metric of 20 and a metric type of 2. All EGP routes redistributed into OSPF are assigned a default metric of 1 and a metric type of 2. By default, a route map is not applied to redistributed routes.

`process` ***pid***

Specifies the PID of an IPv4 OSPF process.

`protocol` ***`protocol`***

The values are defined as follows:

- bgp: Sets the parameters for redistribution of connected routes into OSPF.
- kernel: Sets the parameters for redistribution of kernel routes into OSPF.
- rip: Sets the parameters for redistribution of RIP routes into OSPF.
- static: Sets the parameters for redistribution of static routes into OSPF.
- connected: Sets the parameters for redistribution of connected routes into OSPF.

`metric` ***`metric`***

Applies the specified metric to routes being redistributed into OSPF.

The range is 1 to 16777214.

`metric-type` ***`type`***

The external route metric type, that specifies how cost is calculated, to be associated with the Type 5 default link#state advertisement (LSA). Supported values are:

1: Internal cost is added to external cost. (E1 routes use the redistributed cost plus the cost to the autonomous system boundary router (ASBR).)

2: External cost only. (E2 routes use only the redistributed cost.)

The default is 2.

`route-map` ***`map-name`***

Redistributes routes satisfying the specified route map.

`tag` ***value***

The range is t to 2147483647.

**Configuration mode.**

```
protocols {
       ospf {
               process pid {
                     redistribute {
                            protocol {
                                   metric metric
                                   metric-type type
                                   route-map map-name
                            }
                     }
               }
       }
}
```

Use this command to define the parameters for redistribution of routes into OSPF.

> **Note:** When extended ACLs are used directly or in a route map, the route must be specified in the source field of the ACL and the net mask of the route must be specified in the destination field of the ACL. The inverse mask of the source field is applied to the prefix prior to the comparison with the source. The inverse mask for the destination field is applied to the net mask of the prefix prior to the comparison with the net-mask as specified in the destination field. Effectively, the prefix must match the source with the application of inverse mask of the source field. And the net mask for the prefix must match the destination with the application of the inverse mask of the destination field.

Use the `set` form of this command to set route redistribution parameters.

Use the `delete` form of this command to remove route redistribution parameters.

Use the `show` form of this command to display route redistribution configuration.

## protocols ospf redistribute rip

Sets the parameters for redistribution of RIP routes into OSPF.

**Syntax:**

`set protocols ospf` [ **`process`** *pid* ] **`redistribute rip`** { **`metric`** *metric* | **`metric-type`** *type* | **`route-map`** *map-name* }

**Syntax:**

`delete protocols ospf` [ **`process`** *pid* ] **`redistribute rip`** [ **`metric`** | **`metric-type`** | **`route-map`** ]

**Syntax:**

`show protocols ospf` [ **`process`** *pid* ] **`redistribute rip`** [ **`metric`** | **`metric-type`** | **`route-map`** ]

RIP routes that are redistributed into OSPF are assigned a routing metric of 1 and a metric type of 2. By default, no route map is applied to redistributed RIP routes.

`process` ***pid***

Specifies the PID of an IPv4 OSPF process.

`metric` ***metric***

Applies the specified metric to RIP routes that are redistributed into OSPF. The metric ranges from 1 through 16. The default metric is 1.

`metric-type` ***type***

The metric type of an external route, which specifies how cost is calculated, to associate with the Type 5 default link-state advertisement (LSA). The metric type is either of the following:

1: Adds internal cost to external cost. (E1 routes use the redistributed cost plus the cost to the ASBR.)

2: Uses only external cost. (E2 routes use only the redistributed cost.)

The default type is 2.

`route-map` ***map-name***

Redistributes routes that satisfy the specified route map.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
```

```
                        redistribute {
                                rip {
                                        metric metric
                                        metric-type type
                                        route-map map-name
                                }
                        }
                }
        }
}
```

Use this command to set the parameters for redistribution of Routing Information Protocol (RIP) routes into OSPF.

> **Note:** When extended ACLs are used directly or in a route map, the route must be specified in the source field of the ACL and the net mask of the route must be specified in the destination field of the ACL.
>
> The inverse mask of the source field is applied to the prefix prior to the comparison with the source. The inverse mask for the destination field is applied to the net mask of the prefix prior to the comparison with the net-mask as specified in the destination field.
>
> Effectively, the prefix must match the source with the application of inverse mask of the source field. And the net mask for the prefix must match the destination with the application of the inverse mask of the destination field.

Use the `set` form of this command to set the parameters for redistribution of RIP routes.

Use the `delete` form of this command to remove the parameters for redistribution of RIP routes.

Use the `show` form of this command to display the parameters for redistribution of RIP routes.

# protocols ospf redistribute static

Sets the parameters for redistribution of static routes into OSPF.

**Syntax:**

set protocols ospf [ **process** *pid* ] **redistribute static** { **metric** *metric* | **metric-type** *type* | **route-map** *map-name* }

**Syntax:**

delete protocols ospf [ **process** *pid* ] **redistribute static** [ **metric** | **metric-type** | **route-map** ]

**Syntax:**

show protocols ospf [ **process** *pid* ] **redistribute static** [ **metric** | **metric-type** | **route-map** ]

Static routes that are redistributed into OSPF are assigned a routing metric of 1 and a metric type of 2. By default, no route map is applied to redistributed static routes.

**process** ***pid***

Specifies the PID of an IPv4 OSPF process.

**metric** ***metric***

Applies the specified metric to static routes that are redistributed into OSPF. The metric ranges from 1 through 16. The default metric is 1.

**metric-type** ***type***

The metric type of an external route, which specifies how cost is calculated, to associate with the Type 5 default link-state advertisement (LSA). The metric type is either of the following:

1: Adds internal cost to external cost. (E1 routes use the redistributed cost plus the cost to the ASBR.)

2: Uses only external cost. (E2 routes use only the redistributed cost.)

The default type is 2.

**route-map** ***map-name***

Redistributes routes that satisfy the specified route map.

### Configuration mode

```
protocols {
       ospf {
               process pid {
                     redistribute {
                            static {
                                   metric metric
                                   metric-type type
                                   route-map map-name
                            }
                     }
               }
       }
}
```

Use this command to set the parameters for redistribution of static routes into OSPF.

Use the `set` form of this command to set the parameters for redistribution of static routes.

Use the `delete` form of this command to remove the parameters for redistribution of static routes.

Use the `show` form of this command to display the parameters for redistribution of static routes.

## protocols ospf refresh timers <value>

Sets the time of the OSPF link-state refresh timer.

**Syntax:**

```
set protocols ospf [ process pid ] refresh timers value
```

**Syntax:**

```
delete protocols ospf [ process pid ] refresh timers [ value ]
```

**Syntax:**

```
show protocols ospf [ process pid ] refresh timers
```

By default, the refresh timer expires every 30 minutes (1,800 seconds).

`process` ***pid***

Specifies the PID of an IPv4 OSPF process.

***value***

The time, in seconds, of the OSPF link-state refresh timer. The time ranges from 10 through 1800.

### Configuration mode

```
protocols {
       ospf {
               process pid {
                       refresh {
                              timers value
                       }
               }
       }
}
```

Use this command to set the time of the OSPF link-state refresh timer.

A link-state refresh is a mechanism for validating a link-state acknowledgment (LSA) and resetting its age before it reaches the maximum age. When the link-state refresh timer expires, the router floods a new link-state update to all its neighbors who reset the age of the LSA.

Use the `set` form of this command to set the time of the refresh timer.

Use the `delete` form of this command to restore the default time, which 1800 seconds, of the refresh timer.

Use the `show` form of this command to display the current time of the refresh timer.

## protocols ospf summary-address <address> <action>

Sets the advertise and tag parameters for the summary network and mask.

**Syntax:**

`set protocols ospf` [ **`process`** *pid* ] **`summary-address`** *address* [ **`advertise`** [ **`tag`** *address* ] | **`not-advertise`** ]

**Syntax:**

`delete protocols ospf` [ **`process`** *pid* ] **`summary-address`** *address* [ **`advertise`** **`tag`** *address* | **`not-advertise`** ]

**Syntax:**

`show protocols ospf` [ **`process`** *pid* ] **`summary-address`** *address*

**`process`** ***pid***

Specifies the PID of an IPv4 OSPF process.

***address***

The IPv4 address of the summary network and its mask.

**`advertise`**

The summary address to be advertised.

**`not-advertise`**

The dead-neighbor polling interval. The summary networks within range are not advertised.

**`tag`** ***address***

The neighbor priority. The priority number ranges from 0 through 4294967295. There is no default value for tag.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                     summary-address address {
                            advertise {
                                   tag address
                            }
                            not-advertise
                       }
               }
       }
}
```

Use the `set` form of this command to set the advertise and tag parameters for the summary network and mask.

Use the `delete` form of this command to delete the advertise and tag parameters for the summary network and mask.

Use the `show` form of this command to display the advertise and tag parameters for the summary network and mask.

## show ip ospf

Displays high-level information about OSPF configuration.

**Syntax:**

`show ip ospf` [ **`process`** *pid* ]

**`process`** ***pid***

Specifies the PID of an IPv4 OSPF process.

### Operational mode

Use this command to display high-level information about OSPF configuration.

When used with

The following example shows how to display high-level information about OSPF configuration.

```
vyatta@vyatta:~$ show ip ospf
OSPF Routing Process, Router ID: 10.100.10.1
 Supports only single TOS (TOS0) routes
 This implementation conforms to RFC2328
 RFC1583Compatibility flag is disabled
 OpaqueCapability flag is disabled
 Initial SPF scheduling delay 200 millisec(s)
 Minimum hold time between consecutive SPFs 1000 millisec(s)
 Maximum hold time between consecutive SPFs 10000 millisec(s)
 Hold time multiplier is currently 1
 SPF algorithm last executed 1w2d01h ago
 SPF timer is inactive
 Refresh timer 10 secs
 Number of external LSA 1. Checksum Sum 0x000083e4
 Number of opaque AS LSA 0. Checksum Sum 0x00000000
 Number of areas attached to this router: 1

 Area ID: 10.1.0.0
   Shortcutting mode: Default, S-bit consensus: no
   Number of interfaces in this area: Total: 1, Active: 1
   Number of fully adjacent neighbors in this area: 2
   Area has no authentication
   Number of full virtual adjacencies going through this area: 0
   SPF algorithm executed 3 times
   Number of LSA 4
   Number of router LSA 3. Checksum Sum 0x0000ccad
   Number of network LSA 1. Checksum Sum 0x00000df2
   Number of summary LSA 0. Checksum Sum 0x00000000
   Number of ASBR summary LSA 0. Checksum Sum 0x00000000
   Number of NSSA LSA 0. Checksum Sum 0x00000000
   Number of opaque link LSA 0. Checksum Sum 0x00000000
   Number of opaque area LSA 0. Checksum Sum 0x00000000
vyatta@vyatta:~$
```

The following example shows how to display high-level configuration information for an OSPF process.

```
vyatta@vyatta:~$ show ip ospf process 5
Routing Process "ospf 5" with ID 20.2.2.2
** Allocated Router ID 20.2.2.2 overlaps with one or more ospf processes - please reconfigure
 **
Process uptime is 30 minutes
Process bound to routing-instance default
Conforms to RFC2328, and RFC1583 Compatibility flag is disabled
Supports only single TOS(TOS0) routes
Supports opaque LSA
Does not support Restarting
This router is an ASBR (injecting external routing information)
SPF schedule delay initial 0 secs 500 msecs
SPF schedule delay min 0 secs 500 msecs
SPF schedule delay max 50 secs 0 msecs
Refresh timer 10 secs
Number of incoming current DD exchange neighbors 0/64
Number of outgoing current DD exchange neighbors 0/64
```

```
Initial LSA throttle delay 0 secs 0 msecs
Minimum hold time for LSA throttle 5 secs 0 msecs
Maximum wait time for LSA throttle 5 secs 0 msecs
Minimum LSA arrival 1 secs 0 msecs
Number of external LSA 5. Checksum 0x0220D2
Number of opaque AS LSA 0. Checksum 0x000000
Number of non-default external LSA 5
External LSA database is unlimited.
Number of LSA originated 8
Number of LSA received 5
Number of areas attached to this router: 1
    Area 0 (BACKBONE)
        Number of interfaces in this area is 1(1)
        Number of fully adjacent neighbors in this area is 1
        Area has no authentication
        SPF algorithm last executed 00:28:35.258 ago
        SPF algorithm executed 3 times
        Number of LSA 7. Checksum 0x03be7d
```

# show ip ospf border-routers

Displays information about OSPF border routers.

**Syntax:**
show ip ospf [ **process** *pid* ] **border-routers**

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

**Operational mode**

Use this command to display information about OSPF border routers.

The following example shows how to display information about OSPF border routers.

```
vyatta@vyatta:~$ show ip ospf border-routers

OSPF process 0 VRF (default) internal Routing Table
Codes: i - Intra-area route, I - Inter-area route
i 10.0.100.3 [10] via 10.1.1.3, dp0s5, ASBR, Area 10.0.0.1
```

# show ip ospf database

Displays OSPF database information.

**Syntax:**
show ip ospf [ **process** *pid* ] **database** [ **max-age** | **self-originate** | { **asbr-summary** | **external** | **network** | **nssa-external** | **opaque-area** | **opaque-as** | **opaque-link** | **router** | **summary** } [ **adv-router** *ipv4* | *ipv4* [ **adv-router** *ipv4* | **self-originate** ] ] ]

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

**max-age**
Display OSPF max-age database.

**self-originate**
Display OSPF self-originate database.

**asbr-summary**

Display OSPF Autonomous System Border Router (ASBR) summary database.

**`external`**

Display OSPF external database.

**`network`**

Display OSPF network database.

**`nssa-external`**

Display OSPF NSSA external database.

**`opaque-area`**

Display OSPF opaque-area database.

**`opaque-as`**

Display OSPF opaque-as database.

**`opaque-link`**

Display OSPF opaque-link database.

**`router`**

Display OSPF router database.

**`summary`**

Display summary of OSPF database.

**`adv-router`** ***ipv4***

Display the OSPF database for a given address of the advertised router specified.

***ipv4***

Display the OSPF database for a given address.

**`self-originate`**

Display the self-originate OSPF database for a given address.

### Operational mode

Use this command to display OSPF database information.

The following example shows how to display general information about the OSPF database.

```
vyatta@vyatta:~$ show ip ospf database

       OSPF Router with ID (10.100.10.1)

                Router Link States (Area 10.1.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
10.1.0.33       10.1.0.33        123 0x800003e5 0x791f 1
10.1.0.58       10.1.0.58        123 0x80000562 0x4e7e 1
10.100.10.1     10.100.10.1      117 0x800001b6 0xfe13 1

                Net Link States (Area 10.1.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
10.1.0.58       10.1.0.58        123 0x800003df 0x0bf3

                AS External Link States

Link ID         ADV Router      Age  Seq#       CkSum  Route
76.0.0.0        10.1.0.58       1850 0x800000b3 0x83e4 E2 76.0.0.0/8 [0x0]

vyatta@vyatta:~$
```

## show ip ospf interface

Displays information about OSPF configuration and status for an interface.

**Syntax:**

`show ip ospf` [ **`process`** *pid* ] **`interface`** [ *interface* ]

If no interfaces are specified, then information for all interfaces is displayed.

`process` ***pid***

Specifies the PID of an IPv4 OSPF process.

***interface***

An interface for which to display information.

**Operational mode**

Use this command to display information about OSPF configuration and status for an interface.

The following example shows how to display information about OSPF configuration and status for all interfaces.

```
vyatta@vyatta:~$ show ip ospf interface
dp0p192p1 is up, line protocol is up
  Internet Address 19.1.1.1/24, Area 0.0.0.0, MTU 1500
  Process ID 0, Router ID 50.0.100.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State Backup, Priority 1
  Designated Router (ID) 50.0.100.2, Interface Address 19.1.1.2
  Backup Designated Router (ID) 50.0.100.1, Interface Address 19.1.1.1
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:00
  Neighbor Count is 1, Adjacent neighbor count is 1
  Crypt Sequence Number is 27
  Hello received 577 sent 578, DD received 3 sent 4
  LS-Req received 0 sent 1, LS-Upd received 11 sent 12
  LS-Ack received 11 sent 10, Discarded 0
dp0p224p1 is down, line protocol is down
  Internet Address 22.2.2.2/24, Area 0.0.0.1 [NSSA], MTU 1500
  Process ID 0, Router ID 50.0.100.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State Down, Priority 1
  No designated router on this network
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
vyatta@vyatta:~$
```

# show ip ospf neighbor

Displays information about OSPF neighbors for an address or interface.

**Syntax:**

`show ip ospf` [ **`process`** *pid* ] **`neighbor`** [ *interface* | *ipv4* | **`detail`** | **`address`** *ipv4* ]

If no interfaces are specified, then information about all neighbors is displayed.

`process` ***pid***

Specifies the PID of an IPv4 OSPF process.

***interface***

An interface for which to display neighbor information.

***ipv4***

An address for which to display neighbor information.

**`detail`**

Displays detailed neighbor information for all neighbors.

`address` ***ipv4***

Displays neighbor information for the specified address.

**Operational mode**

Use this command to display information about OSPF neighbors for an address or interface.

The following example shows how to display information about OSPF neighbors for all neighbors.

```
vyatta@vyatta:~$ show ip ospf neighbor

OSPF process 0 VRF(default):
Neighbor      ID  Pri  State    Dead Time  Address    Interface
22.22.22.22   1   Full/Backup  00:00:07  193.1.1.2   dp0p192p1
3.3.3.3       1   Full/Backup  00:00:31  194.1.1.2   dp0p224p1
```

## show ip ospf opaque-area

Displays information about opaque-area LSAs for OSPF (originator is reachable).

**Syntax:**

show ip ospf [ **process** *pid* ] **opaque-area**

process ***pid***

Specifies the PID of an IPv4 OSPF process.

**Operational mode**

Use this command to display information about opaque-area LSAs for OSPF.

The following example shows how to display information about opaque-area LSAs for OSPF.

```
vyatta@vyatta:~$ show ip ospf process 5 opaque-area

                Area-Local Opaque-LSA (Area 0.0.0.0)

  Advertising Router: 10.0.0.3
  Opaque Type: 1
  Opaque ID: 1
  Length: 28

                Area-Local Opaque-LSA (Area 0.0.0.0)

  Advertising Router: 20.2.2.2
  Opaque Type: 1
  Opaque ID: 1
  Length: 28

                Area-Local Opaque-LSA (Area 0.0.0.0)

  Advertising Router: 10.0.0.3
  Opaque Type: 1
  Opaque ID: 6
  Length: 108

                Area-Local Opaque-LSA (Area 0.0.0.0)

  Advertising Router: 20.2.2.2
  Opaque Type: 1
  Opaque ID: 9
  Length: 108
```

# show ip ospf opaque-link

Displays information about opaque-link LSAs for OSPF (originator is reachable).

**Syntax:**

show ip ospf [ **process** *pid* ] **opaque-link**

**process *pid***

Specifies the PID of an IPv4 OSPF process.

**Operational mode**

Use this command to display information about opaque-link LSAs for OSPF.

The following example shows how to display information about opaque-link LSAs for OSPF.

```
vyatta@vyatta:~$ show ip ospf opaque-link
```

# show ip ospf opaque-as

Displays information about opaque-as LSAs for OSPF (originator is reachable).

**Syntax:**

show ip ospf [ **process** *pid* ] **opaque-as**

**process *pid***

Specifies the PID of an IPv4 OSPF process.

**Operational mode**

Use this command to display information about opaque-as LSAs for OSPF.

The following example shows how to display information about opaque-as LSAs for OSPF.

```
vyatta@vyatta:~$ show ip ospf opaque-as
```

# show ip ospf route

Displays information about OSPF routes.

**Syntax:**

show ip ospf [ **process** *pid* ] **route**

**process *pid***

Specifies the PID of an IPv4 OSPF process.

**Operational mode**

Use this command to display information about OSPF routes.

The following example shows how to display information about OSPF routes.

```
vyatta@vyatta:~$ show ip ospf route
```

```
OSPF process 0:
Codes: C - connected, D - Discard, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2

E2 10.1.1.0/24 [10/20] via 10.1.1.3, dp0s5
C  10.1.1.0/24 [10] is directly connected, dp0s4, Area 0.0.0.0
C  10.1.1.0/24 [10] is directly connected, dp0s5, Area 10.0.0.1
O  10.1.1.1/32 [10] via 10.1.1.3, dp0s5, Area 10.0.0.1
E2 10.1.1.0/24 [10/20] via 10.1.1.3, dp0s5
```

# show ip ospf routing-instance

Displays the mapping between routing instance and OSPF process and identifies which processes are running in a given routing instance.

**Syntax:**

show ip ospf [ **process** *pid* ] **routing instance** { **all** | **default** | *instance-name* }

**process *pid***

Specifies the PID of an IPv4 OSPF process.

**Operational mode**

Use this comment to identify the routing instance that a given process runs in.

The following example displays the mapping between each routing instance and OSPF process.

```
vyatta@vyatta:~$ show ip ospf routing-instance all
OSPFv2 processes bound to routing-instance GREEN:
None
OSPFv2 processes bound to routing-instance BLUE:
OSPF process id:100
OSPFv2 processes bound to routing-instance default:
OSPF process id:0
vyatta@vyatta:~$
```

# show ip route ospf

Displays all IP OSPF routes.

**Syntax:**

show ip route ospf

**Operational mode**

Use this command to display all IP OSPF routes.

The following example shows how to display all IP OSPF routes.

```
vyatta@vyatta:~$ show ip route ospf

IP Route Table for VRF "default"
O E2 *> 10.1.1.0/24 [110/20] via 10.1.1.3, dp0s5, 00:04:29
O    *> 10.1.1.1/32 [110/10] via 10.1.1.3, dp0s5, 00:04:30
O E2 *> 10.1.1.0/24 [110/20] via 10.1.1.3, dp0s5, 00:04:29
```

# show monitoring protocols ospf

Displays the debugging flags for the OSPF protocol.

**Syntax:**

`show monitoring protocols ospf [ **process** *pid* ]`

`process` ***pid***

Specifies the PID of an IPv4 OSPF process.

**Operational mode**

Use this command to see how debugging is set for OSPF.

# OSPF Area Commands

## protocols ospf area <area-id>

Defines an OSPF area.

**Syntax:**
`set protocols ospf [ process pid ] area area-id`

**Syntax:**
`delete protocols ospf [ process pid ] area area-id`

**Syntax:**
`show protocols ospf [ process pid ] area area-id`

`process` ***pid***
Specifies the PID of an IPv4 OSPF process.

***area-id***
The identifier of the OSPF area that is being created, expressed either as an IP address or a decimal value.

**Configuration mode**

```
protocols {
 ospf {
  area area-id
 }
}
```

Use this command to create an area within an OSPF Autonomous System (AS).

Use the `set` form of this command to create an OSPF area.

Use the `delete` form of this command to remove an OSPF area.

Use the `show` form of this command to display an OSPF area.

## protocols ospf area <area-id> area-type normal

Designates an OSPF area as a normal area.

**Syntax:**
`set protocols ospf [ process pid ] area area-id area-type normal`

**Syntax:**
`delete protocols ospf[ process pid ] area area-id area-type`

**Syntax:**
`show protocols ospf [ process pid ] area area-id area-type`

`process` ***pid***
Specifies the PID of an IPv4 OSPF process.

***area-id***
The identifier of an OSPF area, expressed as an IP address or a decimal value.

**Configuration mode**

```
protocols {
        ospf {
                process pid {
                        area area-id {
                                area-type {
                                        normal
                                }
                        }
                }
        }
}
```

Use this command to designate an OSPF area as a normal area.

A normal area is an area that is neither a stub area nor a not-so-stubby area. All external routes are advertised into normal areas.

Use the `set` form of this command to set the OSPF area type as normal.

Use the `delete` form of this command to remove area type configuration.

Use the `show` form of this command to display an area type configuration.

# protocols ospf area <area-id> area-type nssa <action>

Designates an OSPF area as an NSSA.

**Syntax:**

```
set protocols ospf [ process pid ] area area-id area-type nssa { default-cost cost | [ default-information-originate [ metric 0-16777214 | metric-type { 1 | 2 } ] | no-summary | translate { always | candidate } }
```

**Syntax:**

```
delete protocols ospf [ process pid ] area area-id area-type nssa [ default-cost cost | default-information-originate [ metric 0-16777214 | metric-type { 1 | 2 } ] | no-summary | translate { always | candidate } ]
```

**Syntax:**

```
show protocols ospf [ process pid ] area area-id area-type nssa
```

By default, summary routes are generated into the area, and only Type 7 LSAs from the candidate NSSA ABR are translated.

**`process`** ***pid***

Specifies the PID of an IPv4 OSPF process.

***area-id***

The ID of the OSPF area being configured, expressed as an IP address or a decimal value.

**`default-cost`** ***cost***

Sets the administrative cost, or metric, that applies to the default route in the specified OSPF area. The cost ranges from 0 through 6777215.

**`default-information-originate`**

Distributes the default route into an NSSA area.

**`metric`**: The OSPF default metric.

**`metric-type`**: Specifies the OSPF metric type on how cost is calculated for default routes. The metric type is either of the following:

1: Adds internal cost to external cost.

2: Uses only external cost.

The default type is 2.

**`no-summary`**

Prevents route summaries from being generated into the area.

**`translate`**

Directs the NSSA ABR when to translate Type 7 LSAs into Type 5 AS-External-LSAs. The direction is either of the following:

`always`: Always translates Type 7 LSAs into Type 5 AS-External-LSAs.

`candidate`: Translates only Type 7 LSAs from the candidate NSSA ABR.

**Configuration mode**

```
protocols {
        ospf {
                process pid {
                        area area-id {
                                area-type {
                                        nssa {
                                                default-cost cost
                                                default-information-originate
                                                metric 0-16777214
                                                metric-type 1-2
                                                no-summary
                                                translate {
                                                        always
                                                        candidate
                                                }
                                        }

                                }
                        }
                }
        }
}
```

Use this command to designate an OSPF area as a not-so-stubby area.

Type 5 AS-External-LSAs are not allowed in stubby areas, but Type 7 LSAs may be translated into Type 5 LSAs by the not-so-stubby (NSSA) area border router and may traverse the NSSA in this manner. Interarea routes are not allowed.

Use the `set` form of this command to set the OSPF area type to not-so-stubby.

Use the `delete` form of this command to designate an OSPF are.

Use the `show` form of this command to display an area type configuration.

# protocols ospf area <area-id> area-type stub <action>

Designates an OSPF area as a stub area.

**Syntax:**

`set protocols ospf [` **`process`** *`pid`* `]` **`area`** *`area-id`* **`area-type stub`** `{` **`default-cost`** *`cost`* `|` **`no-summary`** `}`

**Syntax:**

`delete protocols ospf [` **`process`** *`pid`* `]` **`area`** *`area-id`* **`area-type stub`** `[` **`default-cost`** `|` **`no-summary`** `]`

**Syntax:**

`show protocols ospf [` **`process`** *`pid`* `]` **`area`** *`area-id`* **`area-type stub`** `[` **`default-cost`** `]`

By default, summary routes are generated into the area.

**`process`** ***pid***

Specifies the PID of an IPv4 OSPF process.

***area-id***

The ID of the OSPF area being configured, expressed as an IP address or a decimal value.

**default-cost** ***cost***

Sets the administrative cost, or metric, that applies to the default route in the specified OSPF area. The cost ranges from 0 through 6777215.

**no-summary**

Prevents route summaries from being generated into the area.

### Configuration mode

```
protocols {
       ospf {
               process pid {
                      area area-id {
                              area-type {
                                      stub {
                                             default-cost cost
                                             no-summary
                                      }

                              }
                      }
               }
       }
}
```

Use this command to designate an OSPF area as a stub area. No Type 5 AS-External-LSAs are allowed into a stub area.

Use the `set` form of this command to set the OSPF area type to stub.

Use the `delete` form of this command to remove area type configuration.

Use the `show` form of this command to display an area type configuration.

# protocols ospf area <area-id> authentication <type>

Sets the authentication type of an OSPF area.

### Syntax:

set protocols ospf [ **process** *pid* ] **area** *area-id* **authentication** { **plaintext** | **md5** }

### Syntax:

delete protocols ospf [ **process** *pid* ] **area** *area-id* **authentication** [ **plaintext** | **md5** ]

### Syntax:

show protocols ospf [ **process** *pid* ] **area** *area-id* **authentication**

The authentication is plain text.

**process** ***pid***

Specifies the PID of an IPv4 OSPF process.

***area-id***

The ID of the OSPF area being configured, expressed as an IP address or a decimal value.

**plaintext**

Specifies that passwords are sent through the network in plain text.

**md5**

Specifies the MD5 authentication key to use as input to the MD5 hashing algorithm. This key must be the same on both the sending and receiving systems.

A hash value is sent through the network, computed from the password in the OSPF packet and the password, by using the Message Digest algorithm.

### Configuration mode

```
protocols {
       ospf {
               process pid {
                      area area-id {
                              authentication
                              plaintext
                              md5
                      }
               }
       }
}
```

Use this command to set the authentication type of an OSPF area.

In plain-text authentication, passwords are sent through the network in plain text. In MD5 authentication, the system uses the Message Digest 5 (MD5) algorithm to compute a hash value from the contents of the OSPF packet and the password. The hash value and the MD5 key are included in the transmitted packet, and the receiving system (configured with the same password) calculates its own hash function, which must match.

Use the `set` form of this command to set the authentication type of an OSPF area.

Use the `delete` form of this command to remove the authentication type of an OSPF area.

Use the `show` form of this command to display the authentication type of an OSPF area.

## protocols ospf area <area-id> network <ipv4net>

Sets the network address of an OSPF area.

**Syntax:**
`set protocols ospf` [ **`process`** *pid* ] **`area`** *area-id* **`network`** *ipv4net*

**Syntax:**
`delete protocols ospf` [ **`process`** *pid* ] **`area`** *area-id* **`network`** [ *ipv4net* ]

**Syntax:**
`show protocols ospf` [ **`process`** *pid* ] **`area`** *area-id* **`network`**

**`process`** ***pid***
Specifies the PID of an IPv4 OSPF process.

***area-id***
The ID of the OSPF area being configured, expressed as an IP address or a decimal value.

***ipv4net***
Multinode. A network to use for the specified OSPF area. The format is *ip-address/prefix*.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                      area area-id {
                              network ipv4net
                      }
               }
       }
}
```

Use the `set` form of this command to set the network address of an OSPF area.

Use the `delete` form of this command to delete the network address of an OSPF area.

Use the `show` form of this command to display the network address of an OSPF area.

# protocols ospf area <area-id> range <ipv4net> not-advertise

Directs the router to summarize routes that match a prefix range.

**Syntax:**
`set protocols ospf` [ **process** *pid* ] **area** *area-id* **range** *ipv4net* **not-advertise**

**Syntax:**
`delete protocols ospf` [ **process** *pid* ] **area** *area-id* **range** *ipv4net* [ **not-advertise** ]

**Syntax:**
`show protocols ospf` [ **process** *pid* ] **area** *area-id* **range** *ipv4net*

By default, routes are advertised and routes are not substituted.

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

***area-id***
The ID of the OSPF area being configured, expressed as an IP address or a decimal value.

***ipv4net***
A range to summarize, expressed as an IPv4 network in the format *ip-address/prefix*.

**not-advertise**
Directs the router not to advertise routes in the specified range.

**Configuration mode**

```
protocols {
        ospf {
                process pid {
                        area area-id {
                                area-type {
                                        range {
                                                not-advertise
                                        }

                                }
                        }
                }
        }
}
```

Use this command to direct the router to summarize routes that match a prefix range. This command may be used only with an Area Border Router (ABR).

Use the `set` form of this command to set the area range.

Use the `delete` form of this command to remove area range configuration.

Use the `show` form of this command to display area range configuration.

# protocols ospf area <area-id> shortcut <mode>

Sets the OSPF shortcut mode for an ABR.

**Syntax:**
`set protocols ospf` [ **process** *pid* ] **area** *area-id* **shortcut** { **default** | **disable** | **enable** }

**Syntax:**
`delete protocols ospf` [ **process** *pid* ] **area** *area-id* **shortcut** [ **default** | **disable** | **enable** ]

**Syntax:**
show protocols ospf [ **process** *pid* ] **area** *area-id* **shortcut**

The shortcut mode is `default`.

**process** ***pid***
Specifies the PID of an IPv4 OSPF process.

***area-id***
The ID of the OSPF area being configured, expressed as an IP address or a decimal value.

***mode***
A shortcut mode. The mode is one of the following:

`default`: If the ABR has an active backbone connection, the specified area is not used for shortcutting and the ABR does not set the shortcut bit (S-bit) in the router-LSA that is originated for the area. If the ABR does not have a backbone connection, the area is always used for shortcutting and the ABR sets the S-bit in the router-LSA for that area.

`disable`: The ABR does not use the specified area for shortcutting and does not set the S-bit in the router-LSA that is originated for the area.

`enable`: If the ABR has an active backbone connection, the ABR sets the S-bit in the router-LSA and the specified area is used for shortcutting provided that all other ABRs seen through this area also report the S-bit. If the ABR does not have a backbone connection, the ABR unconditionally uses the area for shortcutting and sets the S-bit in the router-LSA that is originated for the area.

**Configuration mode**

```
protocols {
       ospf {
               process pid {
                      area area-id {
                              shortcut {
                                     default
                                     disable
                                     enable
                      }
               }
       }
}
```

Use this command to set the OSPF shortcut mode for an area border router (ABR), according to the standard that is described in *draft-ietf-ospf-shortcut-abr-02.txt*. This command may be used only with an ABR.

Use the `set` form of this command to set the OSPF shortcut mode for an ABR.

Use the `delete` form of this command to remove the OSPF shortcut mode for an ABR.

Use the `show` form of this command to display the OSPF shortcut mode for an ABR.

## protocols ospf area <area-id> virtual-link <ipv4> authentication <method>

Sets the authentication characteristics for a virtual link.

**Syntax:**
set protocols ospf [ **process** *pid* ] **area** *area-id* **virtual-link** *ipv4* **authentication** { **md5** | **md5-key-id** *1-255* | **plaintext** | **plaintext-password** *password* }

**Syntax:**
delete protocols ospf [ **process** *pid* ] **area** *area-id* **virtual-link** *ipv4* **authentication** [ **md5** | **md5-key-id** *1-255* | **plaintext** | **plaintext-password** *password* ]

**Syntax:**

show protocols ospf [ **process** *pid* ] **area** *area-id* **virtual-link** *ipv4* **authentication**

**process** ***pid***

Specifies the PID of an IPv4 OSPF process.

***area-id***

The ID of the OSPF area being configured, expressed as an IP address or a decimal value.

***ipv4***

The area identifier of a virtual link, expressed as an IPv4 address.

**md5**

Specifies the MD5 authentication key to use as input to the MD5 hashing algorithm. This key must be the same on both the sending and receiving systems.

A hash value is sent through the network, computed from the password in the OSPF packet and the password, by using the Message Digest algorithm.

**md5-key-ID** ***1-255***

Sets the MD5 authentication key identifier. This identifier must be the same on both the sending and receiving systems. The identifier ranges from 1 through 255.

**plaintext**

Specifies the plain-text authentication method.

**plaintext-password** ***password***

Sets the password to use in plain-text authentication. This password must be eight or fewer characters and be the same on both the sending and receiving systems.

**Configuration mode**

```
protocols {
 ospf {
  area area-id {
   virtual-link ipv4 {
    authentication {
     md5 {
      md5
      md5-key 1-255
      plaintext
      plaintext-password password
     }
    }
   }
  }
 }
}
```

Use this command to set the authentication characteristics for a virtual link.

In plain-text authentication, passwords are sent through the network in plain text. In MD5 authentication, the system uses the Message Digest 5 (MD5) algorithm to compute a hash value from the contents of the OSPF packet and the password. The hash value and the MD5 key are included in the transmitted packet, and the receiving system (configured with the same password) calculates its own hash function, which must match.

Use the set form of this command to specify the authentication characteristics for a link.

Use the delete form of this command to remove the authentication characteristics for a link.

Use the show form of this command to display the authentication characteristics for a link.

# protocols ospf area <area-id> virtual-link <ipv4> dead-interval <interval>

Sets the dead interval for a virtual link.

**Syntax:**

`set protocols ospf [` **`process`** *pid* `]` **`area`** *area-id* **`virtual-link`** *ipv4* **`dead-interval`** *interval*

**Syntax:**
`delete protocols ospf [` **`process`** *pid* `]` **`area`** *area-id* **`virtual-link`** *ipv4* **`dead-interval`**

**Syntax:**
`show protocols ospf [` **`process`** *pid* `]` **`area`** *area-id* **`virtual-link`** *ipv4* **`dead-interval`**

The dead interval is 4 times the hello interval.

**`process`** ***pid***
Specifies the PID of an IPv4 OSPF process.

***area-id***
The ID of the OSPF area being configured, expressed as an IP address or a decimal value.

***ipv4***
The area ID of the virtual link, expressed as an IPv4 address.

***interval***
The time, in seconds, that the virtual link should wait to detect hello packets from neighboring routers before declaring the neighbor down. The interval ranges from 1 through 65535. The default interval is 4 times the hello interval.

**Configuration mode**

```
protocols {
 ospf {
  area area-id {
   virtual-link ipv4 {
    dead-interval interval
   }
  }
 }
}
```

Use this command to specify the interval during which a virtual link should expect a hello packet from its neighbor.

If the dead interval passes before the interface receives a hello packet from a neighbor, the status of the neighbor is changed to out-of-service and all associated state is cleared.

The dead interval must be the same for all routers that are to establish two-way communication within a network. If two routers do not agree on these parameters, they do not establish adjacencies and disregard communication from each other.

Use the `set` form of this command to set the dead interval for a link.

Use the `delete` form of this command to restore the default dead interval, which is 4 times the hello interval, for a link.

Use the `show` form of this command to display the dead interval for a link.

# protocols ospf area <area-id> virtual-link <ipv4> hello-interval <interval>

Sets the interval between OSPF hello packets on a virtual link.

**Syntax:**
`set protocols ospf [` **`process`** *pid* `]` **`area`** *area-id* **`virtual-link`** *ipv4* **`hello-interval`** *interval*

**Syntax:**
`delete protocols ospf [` **`process`** *pid* `]` **`area`** *area-id* **`virtual-link`** *ipv4* **`hello-interval`**

**Syntax:**

show protocols ospf [ **process** *pid* ] **area** *area-id* **virtual-link** *ipv4* **hello-interval**

Hello packets are sent every 10 seconds.

**process *pid***
Specifies the PID of an IPv4 OSPF process.

***area-id***
The ID of the OSPF area being configured, expressed as an IP address or a decimal value.

***ipv4***
The area ID of the virtual link, expressed as an IPv4 address.

***interval***
The interval, in seconds, between hello packets. This interval must be the same for all nodes on the network. The interval ranges from 1 through 65535. The default interval is 10 seconds.

**Configuration mode**

```
protocols {
 ospf {
  area area-id {
   virtual-link ipv4 {
    hello-interval interval
   }
  }
 }
}
```

Use this command to set the interval at which OSPF hello packets are sent for a virtual link.

A hello packet is an OSPF packet that detects and maintains relationships with neighbors on the same network (directly connected routers). The greater the interval between hello packets, the less router traffic occurs, but the longer it takes for topological changes to be detected.

The hello interval must be the same for all routers that are to establish two-way communication within a network. If two routers do not agree on these parameters, they do not establish adjacencies and disregard communication from each other.

Use the `set` form of this command to set the interval between OSPF hello packets on a virtual link.

Use the `delete` form of this command to restore the default interval, which is 10 seconds, between OSPF hello packets on a virtual link.

Use the `show` form of this command to display the interval between OSPF hello packets on a virtual link.

# protocols ospf area <area-id> virtual-link <ipv4> retransmit-interval <interval>

Sets the retransmission interval for a virtual link.

**Syntax:**
set protocols ospf [ **process** *pid* ] **area** *area-id* **virtual-link** *ipv4* **retransmit-interval** *interval*

**Syntax:**
delete protocols ospf [ **process** *pid* ] **area** *area-id* **virtual-link** *ipv4* **retransmit-interval**

**Syntax:**
show protocols ospf [ **process** *pid* ] **area** *area-id* **virtual-link** *ipv4* **retransmit-interval**

Unacknowledged LSAs are retransmitted at five-second intervals.

**process *pid***
Specifies the PID of an IPv4 OSPF process.

***area-id***
The ID of the OSPF area being configured, expressed as an IP address or a decimal value.

***ipv4***

The area ID of the virtual link, expressed as an IPv4 address.

***interval***

The interval, in seconds, between the retransmission of unacknowledged LSAs. This interval must be the same for all nodes on the network. The interval ranges from 5 through 65535. The default interval is 5 seconds.

**Configuration mode**

```
protocols {
 ospf {
  area area-id {
   virtual-link ipv4 {
    retransmit-interval interval
   }
  }
 }
}
```

Use this command to set the retransmission interval for a virtual link. This interval is the number of seconds before retransmitting an unacknowledged link-state advertisement (LSA).

When an OSPF router sends an LSA to a neighbor, the neighbor acknowledges receipt with a link-state acknowledgement (LS Ack) packet. If the local router fails to receive the expected LS Ack packet, it retransmits the LSA at the interval specified by this command. This interval must be the same for all nodes on the network.

Use the `set` form of this command to set the retransmission interval for a virtual link.

Use the `delete` form of this command to restore the default retransmission interval, which is 5 seconds, for a virtual link.

Use the `show` form of this command to display the retransmission interval for a virtual link.

# protocols ospf area <area-id> virtual-link <ipv4> transmit-delay <delay>

Sets the transmission delay for a virtual link in an OSPF area.

**Syntax:**

```
set protocols ospf [ process pid ] area area-id virtual-link ipv4 transmit-delay delay
```

**Syntax:**

```
delete protocols ospf [ process pid ] area area-id virtual-link ipv4 transmit-delay
```

**Syntax:**

```
show protocols ospf [ process pid ] area area-id virtual-link ipv4 transmit-delay
```

Link-state transmissions occur at one-second intervals.

`process` ***pid***

Specifies the PID of an IPv4 OSPF process.

***area-id***

The ID of the OSPF area being configured, expressed as an IP address or a decimal value.

***ipv4***

The area ID of the virtual link, expressed as an IPv4 address.

***delay***

The delay, in seconds, between link-state transmissions. This delay must be the same for all nodes on the network. The delay ranges from 1 through 65535. The default delay is 1 second.

**Configuration mode**

```
protocols {
 ospf {
  area area-id {
   virtual-link ipv4 {
    transmit-delay delay
   }
  }
 }
}
```

Use this command to set the transmission delay for a virtual link in an OSPF area. This delay is the estimated time that is required to send a link-state update (LSU) packet.

This timer accommodates transmission and propagation delays on the network, particularly on low-speed networks in which delays may be significant. The router increments the age of link-state advertisements (LSAs) in LSU packets to account for these delays.

The transmission delay includes both the transmission time and propagation delay across the network. The delay is added to the age of the LSA packet before the LSA is transmitted. The LSA age helps the network sequence LSAs, so that it can determine which of competing LSAs is the most recent and trustworthy.

LSAs are numbered in sequence, but the sequence numbers are finite, and so cannot be used as the sole determinant of the most recent LSA. Instead, OSPF also tracks the age of LSAs. Each time the LSA is forwarded to another router, its current age is incremented by the transmission delay. The age of the LSA packet, together with its sequence number, helps the receiving router to determine which version of a received LSA is more recent and, therefore, to be used.

Use the `set` form of this command to set the transmission delay for a virtual link in an OSPF area.

Use the `delete` form of this command to restore the default transmission delay, which is 1 second, for a virtual link in an OSPF area.

Use the `show` form of this command to display the transmission delay for a virtual link in an OSPF area.

# OSPF Interface Commands

## interfaces <interface> <interface-name> ip ospf authentication <text>

Sets the authentication method to use for OSPF on an interface.

**Syntax:**
`set interfaces` *interface interface-name* **`ip ospf authentication`** { **`md5`** | **`md5-key-id`** *key-id* **`md5-key`** *md5-key* | **`plaintext`** | **`plaintext-password`** *password* }

**Syntax:**
`delete interfaces` *interface interface-name* **`ip ospf authentication`** [ **`md5`** | **`md5-key-id`** *key-id* **`md5-key`** *md5-key* | **`plaintext`** | **`plaintext-password`** *password* ]

**Syntax:**
`show interfaces` *interface interface-name* **`ip ospf authentication`**

***interface***
: A type of interface. For detailed keywords and arguments that can be specified as interface types, refer to Supported Interface Types *(page 90)*.

***interface-name***
: The name of an interface.

**`md5 key-id`** ***key-id***
: Sets a key that identifies the MD5 key. This key must be the same on both the sending and receiving systems. The key ranges from 1 through 255.

**`md5-key`** ***md5-key***
: Sets a password-like MD5 key of up to 16 alphanumeric characters to use as input to the MD5 hashing algorithm. The longer the key, the stronger the security. This key must be the same on both the sending and receiving systems.

**`plaintext-password`** ***password***
: A password to use in plain-text authentication. This password must be eight or fewer characters and be the same on both the sending and receiving systems.

**Configuration mode**

```
interfaces {
 interfaces interface-name {
  ip {
   ospf {
    authentication {
     md5 {
      key-id key-id {
       md5-key md5-key
      }
      plaintext
      plaintext-password password
     }
    }
   }
  }
 }
}
```

Use this command to set the authentication method to use for OSPF on an interface. This authentication is independent of the authentication that is configured for the OSPF area.

In plain-text authentication, passwords are sent through the network in plain text. In MD5 authentication, the system uses the Message Digest 5 (MD5) algorithm to compute a hash value from the contents of the OSPF packet and the password. The hash value and the MD5 key are included in the transmitted packet, and the receiving system (configured with the same password) calculates its own hash function, which must match.

The authentication parameters must be the same for all routers that are to establish two-way communication within a network. If two routers do not agree on these parameters, they do not consider adjacencies and disregard communication from each other.

Use the `set` form of this command to set the authentication method to use for OSPF on an interface.

Use the `delete` form of this command to remove the authentication method for OSPF from an interface.

Use the `show` form of this command to display the authentication method to use for OSPF on an interface.

## interfaces <interface> <interface-name> ip ospf bandwidth <bandwidth>

Sets the bandwidth of an interface for calculating OSPF cost.

**Syntax:**
`set interfaces` *interface interface-name* **`ip ospf bandwidth`** *bandwidth*

**Syntax:**
`delete interfaces` *interface interface-name* **`ip ospf bandwidth`** [ *bandwidth* ]

**Syntax:**
`show interfaces` *interface interface-name* **`ip ospf bandwidth`**

***interface***
A type of interface. For detailed keywords and arguments that can be specified as interface types, refer to Supported Interface Types *(page 90)*.

***interface-name***
The name of an interface.

***bandwidth***
The bandwidth of the specified data plane interface in kilobits per second. The bandwidth ranges from 1 through 10000000.

**Configuration mode**

```
interfaces {
 interfaces interface-name {
  ip {
   ospf {
    bandwidth bandwidth
   }
  }
 }
}
```

Use this command to set the bandwidth of an interface for calculating OSPF cost.

Use the `set` form of this command to set the bandwidth of an interface for calculating OSPF cost.

Use the `delete` form of this command to remove the bandwidth of an interface for calculating OSPF cost.

Use the `show` form of this command to display the bandwidth of an interface for calculating OSPF cost.

## interfaces <interface> <interface-name> ip ospf cost <cost>

Sets the routing cost of OSPF on an interface.

**Syntax:**

`set interfaces` *interface interface-name* **`ip ospf cost`** *cost*

**Syntax:**
`delete interfaces` *interface interface-name* **`ip ospf cost`**

**Syntax:**
`show interfaces` *interface interface-name* **`ip ospf cost`**

For details on the default of OSPF cost, refer to the usage guidelines that follow.

***interface***
A type of interface. For detailed keywords and arguments that can be specified as interface types, refer to Supported Interface Types *(page 90)*.

***interface-name***
The name of an interface.

***cost***
The link-state metric (OSPF cost) to be advertised in the link-state advertisement (LSA) as the cost of sending packets over the interface. The cost ranges from 1 through 65535.

**Configuration mode**

```
interfaces {
 interfaces interface-name {
  ip {
   ospf {
    cost cost
   }
  }
 }
}
```

Use this command to manually override the default cost of OSPF that is computed by the system on an interface. Only one cost can be set for each interface.

By default, the metric that is associated with a link is computed as follows:

```
Cost = 108 / bandwidth
```

The cost of reaching any destination is the sum of the costs of the individual hops. Costs are always rounded to the nearest integer. Costs lower than 1 are rounded up to 1.

The following table shows the OSPF costs for some common media types.

**Table 3: OSPF Costs for Common Media Types**

| Media Type | OSPF Cost |
|---|---|
| 56 Kbps | 1785 |
| 64 Kbps | 1562 |
| 128 Kbps | 781 |
| 256 Kbps | 390 |
| 512 Kbps | 195 |
| 768 Kbps | 130 |
| T1 (1.544 Mbps) | 64 |

| Media Type | OSPF Cost |
|---|---|
| E1 (2.048 Mbps) | 48 |
| 4 Mbps Token Ring | 6 |
| 10 Mbps Ethernet | 10 |
| 16 Mbps Token Ring | 6 |
| T3 (44.736 Mbps) | 2 |
| 100+ Mbps | 1 |

The values in the table show how OSPF fails to distinguish between interfaces that are faster than 100 Mbps, for example, between Fast Ethernet (100 Mbps) and Gigabit Ethernet (1000 Mbps) interfaces. If you want to distinguish interfaces equal to or greater than 100 Mbps, you must manually configure the cost of the interface by using this command.

Use the `set` form of this command to set the cost of OSPF on an interface.

Use the `delete` form of this command to restore the default cost of OSPF on an interface.

Use the `show` form of this command to display the cost of OSPF on an interface.

## interfaces <interface> <interface-name> ip ospf dead-interval <interval>

Sets the OSPF dead interval for an interface.

**Syntax:**
`set interfaces` *interface interface-name* **`ip ospf dead-interval`** *interval*

**Syntax:**
`delete interfaces` *interface interface-name* **`ip ospf dead-interval`**

**Syntax:**
`show interfaces` *interface interface-name* **`ip ospf dead-interval`**

The dead interval is 4 times the hello interval.

***interface***
A type of interface. For detailed keywords and arguments that can be specified as interface types, refer to Supported Interface Types *(page 90)*.

***interface-name***
The name of an interface.

***interval***
The time, in seconds, that the specified interface waits to detect hello packets from neighboring routers before declaring a neighbor down. The interval ranges from 1 through 65535. The default interval is 4 times the hello interval.

**Configuration mode**

```
interfaces {
 interfaces interface-name {
  ip {
   ospf {
    dead-interval interval
   }
   }
  }
 }
```

Use this command to specify the interval during which an interface should expect a hello packet from its neighbor.

If the dead interval passes before the interface receives a hello packet from a neighbor, the status of the neighbor is changed to out-of-service and all associated state is cleared.

The dead interval must be the same for all routers that are to establish two-way communication within a network. If two routers do not agree on these parameters, they do not establish adjacencies and disregard communication from each other.

Use the `set` form of this command to set the OSPF dead interval for an interface.

Use the `delete` form of this command to restore the default OSPF dead interval, which is 4 times the hello interval, for an interface.

Use the `show` form of this command to display the OSPF dead interval for an interface.

# interfaces <interface> <interface-name> ip ospf hello-interval <interval>

Sets the interval between OSPF hello packets on an interface.

**Syntax:**
`set interfaces` *interface interface-name* **`ip ospf hello-interval`** *interval*

**Syntax:**
`delete interfaces` *interface interface-name* **`ip ospf hello-interval`**

**Syntax:**
`show interfaces` *interface interface-name* **`ip ospf hello-interval`**

Hello packets are sent every 10 seconds.

***interface***
A type of interface. For detailed keywords and arguments that can be specified as interface types, refer to Supported Interface Types *(page 90)*.

***interface-name***
The name of an interface.

***interval***
The interval, in seconds, between hello packets. This interval must be the same for all nodes on the network.The interval ranges from 1 through 65535. The default interval is 10 seconds.

**Configuration mode**

```
interfaces {
 interfaces interface-name {
  ip {
   ospf {
    hello-interval interval
   }
  }
 }
}
```

Use this command to set the interval at which OSPF hello packets are sent on an interface.

A hello packet is an OSPF packet that detects and maintains relationships with neighbors on the same network (directly connected routers). The greater the interval between hello packets, the less router traffic occurs, but the longer it takes for topological changes to be detected.

The hello interval must be the same for all routers that are to establish two-way communication within a network. If two routers do not agree on these parameters, they do not establish adjacencies and disregard communication for each other.

Use the `set` form of this command to set the interval between OSPF hello packets on an interface.

Use the `delete` form of this command to restore the default interval, which is 10 seconds, between OSPF hello packets on an interface.

Use the `show` form of this command to display the interval between OSPF hello packets on an interface.

# interfaces <interface> <interface-name> ip ospf mtu-ignore

Disables detection of MTU mismatches on an interface.

**Syntax:**
`set interfaces` *interface interface-name* **`ip ospf mtu-ignore`**

**Syntax:**
`delete interfaces` *interface interface-name* **`ip ospf mtu-ignore`**

**Syntax:**
`show interfaces` *interface interface-name* **`ip ospf`**

Detection of MTU mismatches is enabled by default.

***interface***
A type of interface. For detailed keywords and arguments that can be specified as interface types, refer to Supported Interface Types *(page 90)*.

***interface-name***
The name of an interface.

**Configuration mode**

```
interfaces {
 interfaces interface-name {
  ip {
   ospf {
    mtu-ignore
   }
  }
 }
}
```

Use this command to disable detection of maximum transmission unit (MTU) mismatches on an OSPF interface.

OSPF sends the MTU of the interface in a database description packet. If the MTUs of OSPF neighbors do not match, they cannot form an OSPF adjacency and will be stuck in Exstart state. MTU mismatch detection detects MTU mismatches and indicates them in the form of a debug message.

Detection of MTU mismatches is an important troubleshooting feature. If mismatch detection is not enabled, MTU mismatches can be detected only by examining configuration for both interfaces.

Some network setups exist in which MTU mismatches are unavoidable and even part of the setup. Detection of MTU mismatches for only these cases should be disabled, so that normal OSPF adjacencies can be formed.

Use the `set` form of this command to disable detection of MTU mismatches on an interface.

Use the `delete` form of this command to re-enable detection of MTU mismatches on an interface.

Use the `show` form of this command to display whether detection of MTU mismatches on an interface is enabled or disabled.

## interfaces <interface> <interface-name> ip ospf network <type>

Sets the OSPF network type for an interface.

**Syntax:**
`set interfaces` *interface interface-name* **`ip ospf network`** [ **`broadcast`** | **`non-broadcast`** | **`point-to-multipoint`** | **`point-to-point`** ]

**Syntax:**
`delete interfaces` *interface interface-name* **`ip ospf network`**

**Syntax:**
`show interfaces` *interface interface-name* **`ip ospf network`**

Broadcast is supported.

***interface***
A type of interface. For detailed keywords and arguments that can be specified as interface types, refer to Supported Interface Types *(page 90)*.

***interface-name***
The name of an interface.

***type***
The network type for this interface. The type is one of the following:

`broadcast`: The interface supports broadcast mode, such as a LAN link.

`non-broadcast`: The interface does not support broadcast mode.

`point-to-point`: The interface supports point-to-point mode.

`point-to-multipoint`: The interface supports point-to-multipoint mode, such as a PPP interface or a point-to-point logical interface on Frame Relay.

The default type is broadcast.

**Configuration mode**

```
interfaces {
 interfaces interface-name {
  ip {
   ospf {
    network type
   }
  }
 }
}
```

Use this command to set the OSPF network type for the interface.

Use the `set` form of this command to set the OSPF network type for an interface.

Use the `delete` form of this command to remove the OSPF network type for an interface.

Use the `show` form of this command to display the OSPF network type for an interface.

## interfaces <interface> <interface-name> ip ospf priority <priority>

Sets the OSPF priority for an interface.

**Syntax:**

`set interfaces` *interface interface-name* **`ip ospf priority`** *priority*

**Syntax:**
`delete interfaces` *interface interface-name* **`ip ospf priority`**

**Syntax:**
`show interfaces` *interface interface-name* **`ip ospf priority`**

An OSPF interface has a priority of 1.

***interface***
A type of interface. For detailed keywords and arguments that can be specified as interface types, refer to Supported Interface Types *(page 90)*.

***interface-name***
The name of an interface.

***priority***
The OSPF priority for the specified interface. The priority ranges from 0 through 255, where a router with priority 0 can never become the designated router and cannot take part in the DR/BDR election. The default priority is 1.

**Configuration mode**

```
interfaces {
 interfaces interface-name {
  ip {
   ospf {
    priority priority
   }
  }
 }
}
```

Use this command to set the OSPF priority for an interface on the broadcast network to which the interface is connected. The priority determines which routers are selected as the designated router (DR) and backup designated router (BDR) of the area.

The DR and BDR reduce the amount of traffic on OSPF overhead on broadcast networks by reducing the number of adjacent routers to which a router must flood its topological information. In broadcast networks (such as Ethernet), each router establishes an adjacency with only the DR and BDR rather than with every router in its area. The DR and BDR then flood this information to all other routers on the network segment.

The priority ranges from 0 through 255. In general, the router with the highest priority is elected as the DR, and the router with the second-highest priority is elected as the BDR. The higher the number, the higher the priority.

A router with a priority of 0 is ineligible for election as a DR or BDR.

Use the `set` form of this command to set the OSPF priority for an interface.

Use the `delete` form of this command to restore the default OSPF priority, which is 1, for an interface.

Use the `show` form of this command to display the OSPF priority for an interface.

## interfaces <interface> <interface-name> ip ospf retransmit-interval <interval>

Sets the OSPF retransmission interval for an interface.

**Syntax:**
`set interfaces` *interface interface-name* **`ip ospf retransmit-interval`** *interval*

**Syntax:**
`delete interfaces` *interface interface-name* **`ip ospf retransmit-interval`**

**Syntax:**
`show interfaces` *interface interface-name* **`ip ospf retransmit-interval`**

Unacknowledged LSAs are retransmitted at five-second intervals.

***interface***
A type of interface. For detailed keywords and arguments that can be specified as interface types, refer to Supported Interface Types *(page 90)*.

***interface-name***
The name of an interface.

***interval***
The time, in seconds, to wait for an acknowledgment after which the system retransmits an LSA packet to its neighbors. The interval ranges from 5 to 65535. The default interval is 5 seconds.

**Configuration mode**

```
interfaces {
 interfaces interface-name {
  ip {
   ospf {
    retransmit-interval interval
   }
  }
 }
}
```

Use this command to specify how long an interface will wait for an acknowledgment of a link-state update before resending the update.

The LSU packet is part of the exchange of topology databases between routers. When the initial database description (DD) packet is sent, it contains only the headers of the link-state advertisements (LSAs). If the receiving router determines that it requires that piece of the OSPF topology, it sends a link-state request packet to request the complete LSA from the sending router.

After the update packet is sent, the sending router waits for an acknowledgement, either implicit or explicit, from the receiving router. In an explicit acknowledgement, the receiving router sends a link-state acknowledge (LS-Ack) packet to the router that sent the update. In an implicit acknowledgement, the router that sent the update receives an LSA from the receiving router that contains the update information.

If the retransmission interval passes with neither an explicit nor an implicit acknowledgement, the sending router retransmits the LSU packet.

Too high an interval slows network convergence. Too small an interval causes unnecessary retransmission.

Use the `set` form of this command to set the OSPF retransmission interval for an interface.

Use the `delete` form of this command to restore the default OSPF retransmission interval, which is 5 seconds, for an interface.

Use the `show` form of this command to display the OSPF retransmission level for an interface.

# interfaces <interface> <interface-name> ip ospf transmit-delay <delay>

Sets the OSPF transmission delay for an interface.

**Syntax:**
`set interfaces` *interface interface-name* **`ip ospf transmit-delay`** *delay*

**Syntax:**
`delete interfaces` *interface interface-name* **`ip ospf transmit-delay`**

**Syntax:**

`show interfaces` *interface interface-name* **`ip ospf transmit-delay`**

Link-state transmissions occur at one-second intervals.

***interface***

A type of interface. For detailed keywords and arguments that can be specified as interface types, refer to Supported Interface Types *(page 90)*.

***interface-name***

The name of an interface.

***delay***

The delay, in seconds, between link-state transmissions. This delay must be the same for all nodes on the network. The delay ranges from 1 through 65535. The default delay is 1 second.

**Configuration mode**

```
interfaces {
 interfaces interface-name {
  ip {
   ospf {
    transmit-delay delay
   }
  }
 }
}
```

Use this command to set the OSPF transmission delay for an interface. This delay is the estimated time required to send an LSU packet.

This timer accommodates transmission and propagation delays on the network, particularly on low-speed networks in which delays may be significant. The router increments the age of link-state advertisements (LSAs) in link-state update (LSU) packets to account for these delays.

The delay includes both the transmission time and the propagation delay across the network. The delay is added to the age of the LSA packet before the LSA is transmitted. The LSA age helps the network sequence LSAs, so that it can determine which of competing LSAs is the more recent and trustworthy.

LSAs are numbered in sequence, but the sequence numbers are finite, and so cannot be used as the sole determinant of the most recent LSA. Instead, OSPF also tracks the age of LSAs. Each time an LSA is forwarded to another router, its current age is incremented by the transmission delay. The age of the packet, together with its sequence number, help the receiving router to determine which version of a received LSA is more recent and, therefore, to be used.

Use the `set` form of this command to set the OSPF transmission delay for an interface.

Use the `delete` form of this command to restore the default OSPF transmission delay, which is one second, for an interface.

Use the `show` form of this command to display the OSPF transmission delay for an interface.

# Supported Interface Types

The following table shows the syntax and parameters of supported interface types. Depending on the command, some of these types may not apply.

| Interface Type | Syntax | Parameters |
|---|---|---|
| Bridge | `bridge` *brx* | *brx*: The name of a bridge group. The name ranges from br0 through br999. |

| Interface Type | Syntax | Parameters |
|---|---|---|
| Data plane | `dataplane` *interface-name* | *interface-name*: The name of a data plane interface. Following are the supported formats of the interface name:<br>• `dp`*x*`p`*y*`p`*z*—The name of a data plane interface, where<br>— `dp`*x* specifies the data plane identifier (ID). Currently, only dp0 is supported.<br>— `p`*y* specifies a physical or virtual PCI slot index (for example, p129).<br>— `p`*z* specifies a port index (for example, p1). For example, dp0p1p2, dp0p160p1, and dp0p192p1.<br>• `dp`*x*`em`*y* —The name of a data plane interface on a LAN-on-motherboard (LOM) device that does not have a PCI slot, where `em`*y* specifies an embedded network interface number (typically, a small number). For example, dp0em3.<br>• `dp`*x*`s`*y*—The name of a data plane interface in a system in which the BIOS identifies the network interface card to reside in a particular physical or virtual slot *y*, where *y* is typically a small number. For example, for the dp0s2 interface, the BIOS identifies slot 2 in the system to contain this interface.<br>• `dp`*x*`P`*n*`p`*y*`p`*z* —The name of a data plane interface on a device that is installed on a secondary PCI bus, where `P`*n* specifies the bus number. You can use this format to name data plane interfaces on large physical devices with multiple PCI buses. For these devices, it is possible to have network interface cards installed on different buses with these cards having the same slot ID. The value of *n* must be an integer greater than 0. For example, dp0P1p162p1 and dp0P2p162p1. |

| Interface Type | Syntax | Parameters |
|---|---|---|
| Data plane vif | `dataplane` *interface-name* `vif` *vif-id* [`vlan` *vlan-id* ] | *interface-name*: Refer to the preceding description.<br>*vif-id*: A virtual interface ID. The ID ranges from 1 through 4094.<br>*vlan-id*: The VLAN ID of a virtual interface. The ID ranges from 1 through 4094. |
| Loopback | `loopback lo`<br>or<br>`loopback lo`*n* | *n*: The name of a loopback interface, where *n* ranges from 1 through 99999. |
| OpenVPN | `openvpn` *vtunx* | *vtunx*: The identifier of an OpenVPN interface. The identifier ranges from vtun0 through vtun*x*, where *x* is a nonnegative integer. |
| Tunnel | `tunnel` *tunx*<br>or<br>`tunnel` *tunx* `parameters` | *tunx*: The identifier of a tunnel interface you are defining. The identifier ranges from tun0 through tun*x*, where *x* is a nonnegative integer. |
| Virtual tunnel | `vti` *vtix* | *vtix*: The identifier of a virtual tunnel interface you are defining. The identifier ranges from vti0 through vti*x*, where *x* is a nonnegative integer.<br>**Note:** Before you can configure a vti interface, you must configure a corresponding vpn.<br>**Note:** This interface does not support IPv6. |
| VRRP | *parent-interface* `vrrp vrrp-group` *group* | *parent-interface*: The type and identifier of a parent interface; for example, data plane dp0p1p2 or bridge br999.<br>*group*: A VRRP group identifier.<br>The name of a VRRP interface is not specified. The system internally constructs the interface name from the parent interface identifier plus the VRRP group number; for example, dp0p1p2v99. Note that VRRP interfaces support the same feature set as does the parent interface. |

# VRF Support

## VRF support for OSPF and OSPFv3

The AT&T Vyatta vRouter supports the implementation of OSPF and OSPFv3 on routing instances.

If you configure OSPF or OSPFv3 on the AT&T Vyatta vRouter without specifying a routing instance, the router uses the default routing instance. When configuring OSPF or OSPFv3 for a particular routing instance, you must associate the instance with an OSPF process, and the process ID must be specified in the configuration command.

### OSPF configuration

All OSPF configuration commands are supported on routing instances.

In the following example, OSPF area 0 is defined for the default routing instance.

```
vyatta@R1# set protocols ospf area 0
```

In the following example, OSPF area 1 is defined for the RED routing instance, which is associated with OSPF process 1.

```
vyatta@R1# set routing routing-instance RED protocols ospf process 1 area 1
```

### OSPFv3 configuration

All OSPFv3 configuration commands are supported on routing instances.

> **Note:** In the default routing instance, OSPFv3 supports the default process (configuration without including a process ID) and non-default process (configuration with a process ID). In non-default routing instances, OSPFv3 supports only non-default processes (configuration with a process ID).

The following examples show the syntax for an individual configuration command. For an example of how to configure OSPFv3 processes on routing instances, refer to the section on configuring OSPFv3 on routing instances in AT&T Vyatta Network Operating System Basic Routing Configuration Guide.

In the following example, OSPFv3 access list 15 is specified for the default routing instance to filter networks in routing updates.

```
vyatta@R1# set protocols ospfv3 distribute-list 15
```

The following example shows how to apply the same configuration to the RED routing instance, which is associated with OSPFv3 process 10.

```
vyatta@R1# set routing routing-instance RED protocols ospfv3 process 10 distribute-list 15
```

## Command support for VRF routing instances

VRF allows an AT&T Vyatta vRouter to support multiple routing tables, one for each VRF routing instance. Some commands in this guide support VRF and can be applied to particular routing instances.

Use the guidelines in this section to determine correct syntax when adding VRF routing instances to commands. For more information about VRF, refer to AT&T Vyatta Network Operating System Basic Routing Configuration Guide. This guide includes an overview of VRF, VRF configuration examples, information about VRF-specific features, and a list of commands that support VRF routing instances.

### Adding a VRF routing instance to a Configuration mode command

For most Configuration mode commands, specify the VRF routing instance at the beginning of a command. Add the appropriate VRF keywords and variable to follow the initial action (**`set`**, **`show`**, or **`delete`**) and before the other keywords and variables in the command.

**Example: Configuration mode example: syslog**

The following command configures the syslog logging level for the specified syslog host. The command does not include a VRF routing instance, so the command applies to the default routing instance.

```
vyatta@R1# set system syslog host 10.10.10.1 facility all level debug
vyatta@R1# show system syslog
syslog {
    host 10.10.10.1 {
            facility all {
                    level debug
            }
    }
}
```

The following example shows the same command with the VRF routing instance (GREEN) added. Notice that **`routing routing-instance GREEN`** has been inserted between the basic action (**set** in the example) and the rest of the command. Most Configuration mode commands follow this convention.

```
vyatta@R1# set routing routing-instance GREEN system syslog host 10.10.10.1 facility all
 level debug
vyatta@R1# show routing
routing {
     routing-instance GREEN {
             system {
                     syslog {
                            host 11.12.13.2:514 {
                                     facility all {
                                             level debug
                                     }
                             }
                     }
             }
     }
}
```

**Example: Configuration mode example: SNMP**

Some features, such as SNMP, are not available on a per-routing instance basis but can be bound to a specific routing instance. For these features, the command syntax is an exception to the convention of specifying the routing instance at the beginning of Configuration mode commands.

The following example shows how to configure the SNMPv1 or SNMPv2c community and context for the RED and BLUE routing instances. The first two commands specify the RED routing instance as the context for community A and BLUE routing instance as the context for community B. The subsequent commands complete the configuration.

For more information about configuring SNMP, refer to AT&T Vyatta Network Operating System Remote Management Configuration Guide.

```
vyatta@R1# set service snmp community commA context RED
vyatta@R1# set service snmp community commB context BLUE
vyatta@R1# set service snmp view all oid 1
vyatta@R1# set service snmp community commA view all
```

```
vyatta@R1# set service snmp community commB view all
vyatta@R1# show service snmp community
 community commA {
        context RED
        view all
 }
 community commB {
        context BLUE
        view all
 }
[edit]
vyatta@vyatta#
```

### Adding a VRF routing instance to an Operational mode command

The syntax for adding a VRF routing instance to an Operational mode command varies according to the type of command parameters:

- If the command does not have optional parameters, specify the routing instance at the end of the command.
- If the command has optional parameters, specify the routing instance after the required parameters and before the optional parameters.

**Example: Operational mode examples without optional parameters**

The following command displays dynamic DNS information for the default routing instance.

```
vyatta@vyatta:~$ show dns dynamic status
```

The following command displays the same information for the specified routing instance (GREEN). The command does not have any optional parameters, so the routing instance is specified at the end of the command.

```
vyatta@vyatta:~$ show dns dynamic status routing-instance GREEN
```

**Example: Operational mode example with optional parameters**

The following command obtains multicast path information for the specified host (10.33.2.5). A routing instance is not specified, so the command applies to the default routing instance.

```
vyatta@vyatta:~$ mtrace 10.33.2.5 detail
```

The following command obtains multicast path information for the specified host (10.33.2.5) and routing instance (GREEN). Notice that the routing instance is specified before the optional **detail** keyword.

```
vyatta@vyatta:~$ mtrace 10.33.2.5 routing-instance GREEN detail
```

**Example: Operational mode example output: SNMP**

The following SNMP **show** commands display output for routing instances.

```
vyatta@vyatta:~$ show snmp routing-instance
Routing Instance SNMP Agent is Listening on for Incoming Requests:
Routing-Instance            RDID
```

```
-----------------           ----
RED                         5

vyatta@vyatta:~$ show snmp community-mapping
SNMPv1/v2c Community/Context Mapping:
Community                   Context
---------                   -------
commA                       'RED'
commB                       'BLUE'
deva                        'default'


vyatta@vyatta:~$ show snmp trap-target
SNMPv1/v2c Trap-targets:
Trap-target                   Port   Routing-Instance Community
-----------                   ----   ---------------- ---------
1.1.1.1                              'RED'           'test'


vyatta@vyatta:~$ show snmp v3 trap-target
SNMPv3 Trap-targets:
Trap-target                   Port   Protocol Auth Priv Type   EngineID              Routing-
Instance User
-----------                   ----   -------- ---- ---- ----   --------
 ---------------- ----
2.2.2.2                       '162'  'udp'    'md5      'infor                       'BLUE'
         'test'
```

# List of Acronyms

| Acronym | Description |
|---|---|
| ACL | access control list |
| ADSL | Asymmetric Digital Subscriber Line |
| AH | Authentication Header |
| AMI | Amazon Machine Image |
| API | Application Programming Interface |
| AS | autonomous system |
| ARP | Address Resolution Protocol |
| AWS | Amazon Web Services |
| BGP | Border Gateway Protocol |
| BIOS | Basic Input Output System |
| BPDU | Bridge Protocol Data Unit |
| CA | certificate authority |
| CCMP | AES in counter mode with CBC-MAC |
| CHAP | Challenge Handshake Authentication Protocol |
| CLI | command-line interface |
| DDNS | dynamic DNS |
| DHCP | Dynamic Host Configuration Protocol |
| DHCPv6 | Dynamic Host Configuration Protocol version 6 |
| DLCI | data-link connection identifier |
| DMI | desktop management interface |
| DMVPN | dynamic multipoint VPN |
| DMZ | demilitarized zone |
| DN | distinguished name |
| DNS | Domain Name System |

| Acronym | Description |
|---|---|
| DSCP | Differentiated Services Code Point |
| DSL | Digital Subscriber Line |
| eBGP | external BGP |
| EBS | Amazon Elastic Block Storage |
| EC2 | Amazon Elastic Compute Cloud |
| EGP | Exterior Gateway Protocol |
| ECMP | equal-cost multipath |
| ESP | Encapsulating Security Payload |
| FIB | Forwarding Information Base |
| FTP | File Transfer Protocol |
| GRE | Generic Routing Encapsulation |
| HDLC | High-Level Data Link Control |
| I/O | Input/Output |
| ICMP | Internet Control Message Protocol |
| IDS | Intrusion Detection System |
| IEEE | Institute of Electrical and Electronics Engineers |
| IGMP | Internet Group Management Protocol |
| IGP | Interior Gateway Protocol |
| IPS | Intrusion Protection System |
| IKE | Internet Key Exchange |
| IP | Internet Protocol |
| IPOA | IP over ATM |
| IPsec | IP Security |
| IPv4 | IP Version 4 |
| IPv6 | IP Version 6 |
| ISAKMP | Internet Security Association and Key Management Protocol |
| ISM | Internet Standard Multicast |

| Acronym | Description |
|---|---|
| ISP | Internet Service Provider |
| KVM | Kernel-Based Virtual Machine |
| L2TP | Layer 2 Tunneling Protocol |
| LACP | Link Aggregation Control Protocol |
| LAN | local area network |
| LDAP | Lightweight Directory Access Protocol |
| LLDP | Link Layer Discovery Protocol |
| MAC | medium access control |
| mGRE | multipoint GRE |
| MIB | Management Information Base |
| MLD | Multicast Listener Discovery |
| MLPPP | multilink PPP |
| MRRU | maximum received reconstructed unit |
| MTU | maximum transmission unit |
| NAT | Network Address Translation |
| NBMA | Non-Broadcast Multi-Access |
| ND | Neighbor Discovery |
| NHRP | Next Hop Resolution Protocol |
| NIC | network interface card |
| NTP | Network Time Protocol |
| OSPF | Open Shortest Path First |
| OSPFv2 | OSPF Version 2 |
| OSPFv3 | OSPF Version 3 |
| PAM | Pluggable Authentication Module |
| PAP | Password Authentication Protocol |
| PAT | Port Address Translation |
| PCI | peripheral component interconnect |

| Acronym | Description |
|---|---|
| PIM | Protocol Independent Multicast |
| PIM-DM | PIM Dense Mode |
| PIM-SM | PIM Sparse Mode |
| PKI | Public Key Infrastructure |
| PPP | Point-to-Point Protocol |
| PPPoA | PPP over ATM |
| PPPoE | PPP over Ethernet |
| PPTP | Point-to-Point Tunneling Protocol |
| PTMU | Path Maximum Transfer Unit |
| PVC | permanent virtual circuit |
| QoS | quality of service |
| RADIUS | Remote Authentication Dial-In User Service |
| RHEL | Red Hat Enterprise Linux |
| RIB | Routing Information Base |
| RIP | Routing Information Protocol |
| RIPng | RIP next generation |
| RP | Rendezvous Point |
| RPF | Reverse Path Forwarding |
| RSA | Rivest, Shamir, and Adleman |
| Rx | receive |
| S3 | Amazon Simple Storage Service |
| SLAAC | Stateless Address Auto-Configuration |
| SNMP | Simple Network Management Protocol |
| SMTP | Simple Mail Transfer Protocol |
| SONET | Synchronous Optical Network |
| SPT | Shortest Path Tree |
| SSH | Secure Shell |

| Acronym | Description |
|---|---|
| SSID | Service Set Identifier |
| SSM | Source-Specific Multicast |
| STP | Spanning Tree Protocol |
| TACACS+ | Terminal Access Controller Access Control System Plus |
| TBF | Token Bucket Filter |
| TCP | Transmission Control Protocol |
| TKIP | Temporal Key Integrity Protocol |
| ToS | Type of Service |
| TSS | TCP Maximum Segment Size |
| Tx | transmit |
| UDP | User Datagram Protocol |
| VHD | virtual hard disk |
| vif | virtual interface |
| VLAN | virtual LAN |
| VPC | Amazon virtual private cloud |
| VPN | virtual private network |
| VRRP | Virtual Router Redundancy Protocol |
| WAN | wide area network |
| WAP | wireless access point |
| WPA | Wired Protected Access |