ADTRAN Operating System (AOS)

Configuring OSPFv2 in AOS

Configuration Guide

61200870L1-29.3D

June 2020

To the Holder of this Document

This document is intended for the use of ADTRAN customers only for the purposes of the agreement under which the document is submitted, and no part of it may be used, reproduced, modified or transmitted in any form or means without the prior written permission of ADTRAN.

The contents of this document are current as of the date of publication and are subject to change without notice.

Trademark Information

"ADTRAN" and the ADTRAN logo are registered trademarks of ADTRAN, Inc. Brand names and product names included in this document are trademarks, registered trademarks, or trade names of their respective holders.

Disclaimer of Liability

The information or statements given in this document concerning the suitability, capacity, or performance of the mentioned hardware or software products are given "as is", and any liability arising in connection with such hardware or software products shall be governed by ADTRAN's standard terms and conditions of sale unless otherwise set forth in a separately negotiated written agreement with ADTRAN that specifically applies to such hardware or software products.

To the fullest extent allowed by applicable law, in no event shall ADTRAN be liable for errors in this document for any damages, including but not limited to special, indirect, incidental or consequential, or any losses, such as but not limited to loss of profit, revenue, business interruption, business opportunity or data, that may arise from the use of this document or the information in it.

Revision History

Rev D	June 2020	Updated document format and corrected IP addresses in example illustration.

Table of Contents

1. Configuring OSPFv2 in AOS

This document explains the advantages of the Open Shortest Path First version 2 (OSPFv2) protocol over a distance-vector routing protocol such as Routing Information Protocol (RIP), describes the key components necessary to implement OSPFv2, discusses the types of networks in which OSPFv2 is implemented, and gives details on topics such as Hello packets, link-state advertisements (LSAs), tables and databases, router types, and areas. In addition, this guide provides configuration instructions for OSPFv2 globally and per interface on ADTRAN Operating System (AOS) products, an OSPFv2 configuration example, and troubleshooting information.

2. OSPFv2 Overview

OSPFv2 (version 1 was never implemented) is a powerful alternative to RIP, superior to RIP in many ways, but it makes more demands on router resources and requires more planning to implement. The next section of this configuration guide delineates the differences between OSPFv2 and RIP and explains both the benefits and negative aspects of OSPFv2.

Link-State vs. Distance-Vector Routing Protocols

Basic RIP configuration and setup are well known. RIP looks at Hop Count to determine its cost for the network. Every 30 seconds, RIP propagates its routing table out all interfaces configured for RIP. As a result, on larger networks RIP is very *chatty*. Although RIP has certain mechanisms to handle routing loops (split horizon and split horizon with poison reverse), it does not handle redundant paths gracefully. In some instances, RIP's convergence time causes the network to be down for an extended time while trying to re-converge the network.

Because of these issues, a link-state protocol such as OSPFv2 may be a better solution than RIP in large networks (links that consist of more than 15 hops) and in significantly redundant networks. This guide describes how OSPFv2 solves these RIP issues.

OSPFv2 was designed to address RIP's shortcoming of offering only a hop count metric. Unlike RIP, OSPFv2 is a link-state routing protocol, which means that changes in routing are based on the status and speeds of the physical links. In addition, changes are immediately propagated to every router on the network.

When an OSPFv2 router is first activated, it uses OSPFv2’s Hello protocol (for more information refer to *Hello Protocol on page 9*) to discover any connected neighboring routers. It then exchanges link-state information with these routers in the form of LSAs. Each router creates an LSA database that consists of every OSPF interface, the corresponding neighbor, and a metric representing the speed of the interface. Each router then uses LSAs to pass this information along to each neighboring router. Each router passes all received LSAs to all its other neighbors.

The link-state database is distinct from the routing table. From the link-state database information, each router calculates a path to every destination on the network, building a tree with itself at the root. This comprises its shortest path first (SPF) tree, which forms the basis of the routing table. The routers exchange LSAs every 30 minutes, unless there is a change in network topology. If an interface goes down, for example, the information is propagated across the network at once. If there is a redundant path, the convergence will last as long as it takes to recalculate the SPF tree and update the routing tables. This can happen in a few seconds or less, depending on the size of the network. Because of these calculations, routers running OSPFv2 require more CPU resources. Frequent flooding of LSAs may also cause problems on wide area network (WAN) links.

OSPFv2 compensates for increased CPU and memory demands by dividing the network into separate, hierarchical domains, called areas. Routers exchange LSAs only with other routers in their own areas. There is also a backbone area known as Area 0. All areas must be adjacent to Area 0. A border between two areas is defined on an area border router (ABR). ABRs have at least one interface in Area 0 and one interface in another non-backbone area. The best designed OSPFv2 networks contain contiguous networks to each area that can be summarized on the backbone through variable length subnet masks (VLSMs). This makes it possible to summarize multiple networks in one routing table entry.

Because routers within the same area share the same information, they have identical topological databases. When two neighboring routers are synchronized, the routers are said to be adjacent. When network topology changes, adjacent routers send LSAs to provide information on a router's adjacencies or to inform others when a router's state changes. By comparing established adjacencies to link-states, failed routers can be detected quickly and the network's topology can be altered appropriately.

Where RIP is unable to consider the speed of interfaces when determining the best path through the network, OSPFv2 addresses this by considering a cost that is derived by the speed of the interface. However, since the formula for determining the cost is not standardized, the default settings will vary.

The OSPFv2 protocol was developed by the OSPFv2 working group of the Internet Engineering Task Force. It is expressly designed for the Transmission Control Protocol (TCP)/IP Internet environment and includes explicit support for Classless Inter-Domain Routing (CIDR) and the tagging of externally derived routing information. OSPFv2 also provides for the authentication of routing updates, and utilizes IP multicast when sending/receiving the updates.

The following table summarizes the differences between link-state and distance-vector routing protocols.

Table 1. Distance-Vector versus Link-State

Distance-Vector	Link-State
Sends its entire routing table at periodic intervals out of all interfaces (typically, this is based in seconds). It also sends triggered updates to reflect changes in the network.	Sends incremental updates when a change is detected. OSPFv2 sends summary information every 30 minutes, regardless of whether incremental updates have been sent in that same time.
Typically involves updates sent using a broadcast address to everyone on the link.	Typically involves updates sent to those routers participating in the routing protocol domain via a multicast address.
Uses a metric based on distance of the remote network from the router.	Is capable of using a complex metric, referred to by OSPFv2 as cost.
Has knowledge of the network based on information learned from its neighbor.	Has knowledge of the network based on information learned from every router in the area.

As mentioned previously, a neighbor in OSPFv2 is a router that shares the same network link (this is the same physical segment), and a router running OSPFv2 discovers its neighbors and adjacencies by sending Hello packets every 10 seconds. The Hello packet has a source address of the router and a multicast destination address set to ALLSPFRouters: 224.0.0.5. The multicast address 224.0.0.6 is used for communication to a designated router (DR) or backup designated router (BDR). The type of network media used determines how the Hello protocol functions and how OSPFv2 builds its neighbor. The following section describes OSPFv2 network topologies.

OSPFv2 Network Topologies

OSPFv2 can be used in the following network topologies:

- Broadcast multi-access
- Point-to-point
- Point-to-multipoint
- Non-broadcast multi-access (NBMA)
- Virtual links

Broadcast Multi-access

This network topology is used in local area networks (LANs) such as Ethernet, token ring, or fiber distributed data interface (FDDI). OSPFv2 sends out multi-access traffic. A DR and a BDR are elected. (These two types of routers will be discussed later in this document.) Each pair of routers on a broadcast network is assumed to be able to communicate directly. Networks supporting many (more than two) attached routers use a single physical message to all the attached routers (broadcast). Neighboring routers are discovered dynamically on these networks using OSPFv2's Hello protocol. The Hello protocol itself takes advantage of the broadcast capability. The OSPFv2 protocol makes further use of multicast capabilities, if they exist.

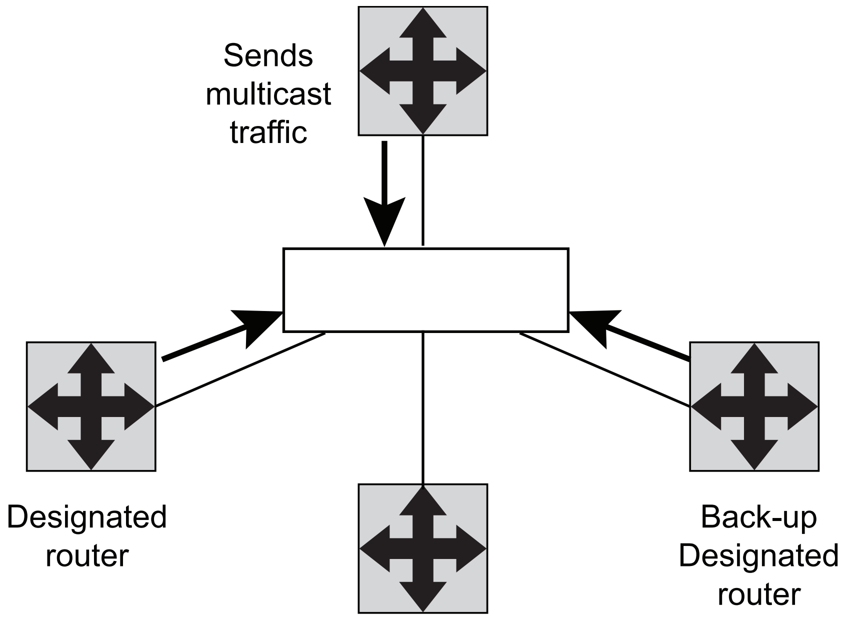

Figure 1. Broadcast Multi-access Network Topology

Point-to-Point

This topology is used when only one other router is directly connected to the transmitting or receiving interface. A typical example of this is a serial line. OSPFv2 has no need of a DR or BDR in this scenario. Network traffic uses the multicast address for OSPFv2 AllSPFRouters 224.0.0.5.

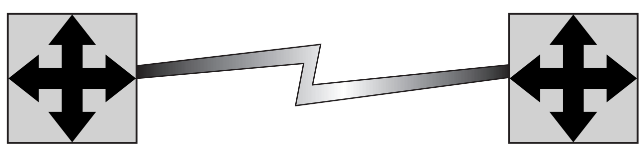

Figure 2. Point-to-Point Network Topology

Point-to-Multipoint

Point-to-multipoint topology features a single interface that connects to many destinations. The underlying network treats the network as a series of point-to-point circuits and replicates LSA packets for each circuit. The addressing of network traffic, again, is multicast. There is no DR or BDR election. This technology uses one IP subnet. According to the RFC, there are only two types of non point-to-point networks:

point-to-multipoint and NBMA. (Cisco has a non-broadcast implementation of point-to-multipoint. The RFC only cites broadcast point-to-multipoint networks.)

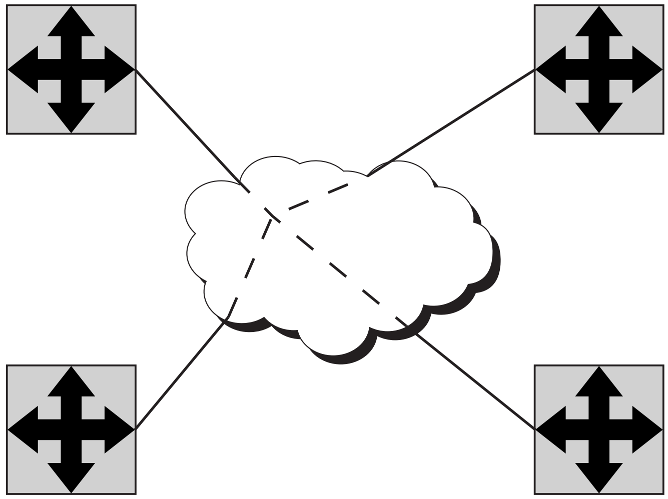

Figure 3. Point-to-Multipoint Network Topology

Non-broadcast Multi-access

Non-broadcast multi-access (NBMA) networks, which are not configurable on AOS devices, include multipoint or access networks such as Frame Relay, X.25, and Asynchronous Traffic Mode (ATM). Broadcast messages are turned off by default. Neighbor tables must be created manually. Networks supporting many (more than two) routers but having no broadcast capability are considered NBMA networks. Neighboring routers are maintained on these networks using OSPFv2's Hello protocol. However, due to the lack of broadcast capability, some configuration information may be necessary to aid in the discovery of neighbors. (A separate neighbor command is implemented.) On non-broadcast networks, OSPFv2 protocol packets that are normally multicast need to be sent to each neighboring router, in turn. Frame Relay is an example of a non-broadcast network.

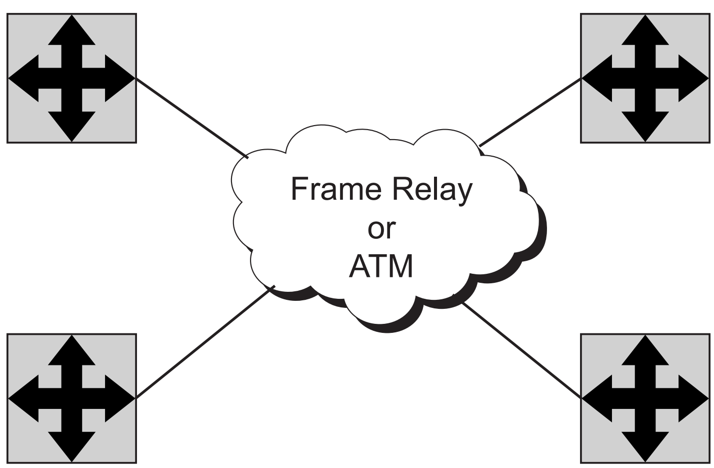

Figure 4. Non-broadcast Multi-access Network Topology

Non-broadcast networks are referred to as NBMA networks or point-to-multipoint networks, depending on OSPFv2's mode of operation over the network. The first mode, called non-broadcast multi-access or NBMA, simulates the operation of OSPFv2 on a broadcast network. The second mode, called point-to-multipoint, treats the non-broadcast network as a collection of point-to-point links.

Virtual Link

A virtual link topology is a virtual connection that tunnels through the network. The OSPFv2 network traffic is sent in unicast datagrams across these links. There are three scenarios for the connection:

- Two backbone areas that need to be combined into one large area 0.0.0.0 (possibly, two independent OSPFv2 networks are combined).
- This is a virtual connection to a remote area that does not have any connections to the backbone (area 0.0.0.0).
- The area is critical to the company and an extra link has been configured for redundancy.

Hello Protocol

A link-state routing protocol develops a relationship with a neighbor router: one that is on the same physical network. The two routers must also have the same subnet mask and the same hello timers. The routing protocol develops and maintains a relationship by sending a simple message, known as the Hello protocol, across the medium. This Hello protocol is used by OSPFv2 routers to advertise themselves on the network when they first join and to form relationships with other OSPFv2 routers. This relationship (the neighbor relationship) is maintained as long as the simple message (Hello protocol) is received. If it is not received before the dead timer, the neighbor is determined to be unavailable.

Table 2. Hello Protocol Timer

Network Type	Hello Timer	Dead Timer
Frame Relay - NBMA	30 seconds	120 seconds
Frame Relay - Point-to-Point	10 seconds	40 seconds
Ethernet - Broadcast	10 seconds	40 seconds

Streamlined communications result because, after the topological databases are synchronized, incremental updates are sent to the neighbors as soon as a change is perceived, as well as every 30 minutes. Adjacencies created between neighbors control the distribution of the router protocol packets. Adjacencies and neighbor relationships result in a much faster convergence of the network than can be achieved by RIP version 1.

OSPFv2 Link-State Advertisement

LSAs are link-state protocol packets that contain information about neighbors and path costs. LSAs are used by the receiving router to maintain their routing tables. The following figure shows LSAs in action.

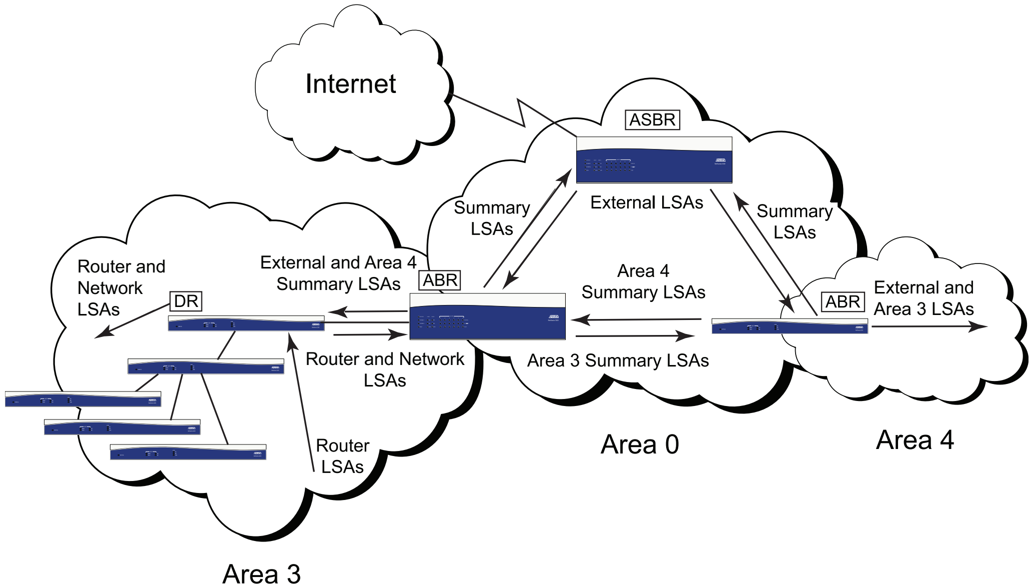

Figure 5. OSPFv2 Link-State Advertisement

The table below describes LSA types.

Table 3. LSA Types

LSA Type	Routing Table Entry	Description
Type 1 - Router	O	This is generated by each router, listing all the links to which it is connected, their status, and their cost. It is propagated within the area.
Type 2 - Network	O	This is generated by the DR on a multi-access LAN to the area.
Type 3 - Summary	IA	This LSA includes the networks or subnets within an area that may have been summarized and that are sent into the backbone and between ABRs.
Type 4 - ASBR-Summary	IA	This LSA is sent to the autonomous system boundary router (ASBR) from the ABR to allow routers to calculate the routing path for external and NSSA LSAs that are generated from the router that the ASBR-Summary describes. (As per RFC2328 section 12.4.3.1 these are not allowed in stub areas or totally stubby areas.)

Table 3. LSA Types (Continued)

LSA Type	Routing Table Entry	Description
Type 5 - External	E1 or E2	The routes in this LSA are external to the autonomous system. They can be configured to have one or two as the metric type. E1 will include the internal cost to the ASBR. E2 does not compute the internal cost; it simply reports the external cost to the remote destination. (E1 externals take precedence over E2 externals.)
Type 6 - Group Membership	Cisco Proprietary	Flooded by a multicast OSPF (MOSPF) router to distribute group membership location information throughout the routing domain. MOSPF is a Cisco implementation.
Type 7 - Not So Stubby Area	Cisco Proprietary	Originated by the ASBR in a not so stubby area (NSSA). These LSAs are similar to Type 5 LSAs except that they are only flooded within the NSSA. At the ABR, selected Type 7 LSAs are translated into Type 5 LSAs and flooded into the backbone.

For LSAs to be propagated into an area, the following criteria must be met:

- The LSA was not received through the interface.
- The interface is in a state of exchange or full adjacency.
- The interface is not connected to a stub area (applies to Type 4 and Type 5 LSAs).

OSPFv2 Tables and Databases

The following sections describe the link-state database, OSPFv2 route table, and neighbor database used in OSPFv2.

Topology Database/Link-State Database

The link-state map or database is created and updated by incremental link-state advertisements. This topology database outlines all neighbors and adjacencies for the network. The router will use Dijkstra's algorithm (SPF) to compute a new map, placing itself as the root. This map will determine which routes are added to the route table.

Cost helps determine the metric for which routes should be added to the routing table as the preferred route. OSPFv2 cost is heavily weighted by bandwidth, and cost is calculated by dividing the reference bandwidth (in kbps) by the configured bandwidth of the interface (for example, 10^5 kbps/56 kbps = 1785). The table below shows the default cost for various interfaces:

Table 4. OSPFv2 Cost

Link Type	Default Cost
56-kbps serial link	1785
T1 (1.544 Mbps)	64
10 Mbps Ethernet	10

> **NOTE**
>
> *OSPFv2 allows for load balancing across equal-cost paths.*

OSPFv2 Route Table

After a neighbor is discovered in OSPFv2, an adjacency is formed. Comparing link-state databases forms adjacencies. Once these databases are created, the SPF algorithm is applied, and a tree of shortest paths is created. The router calculating the shortest path is considered the root of the tree. While the tree calculates the entire cost to a particular destination, only the next hop to the destination is used in the forwarding process.

The following five packets are used to build the route table for the first time:

- **Hello protocol** - Finds neighbors and determines the DR and BDR as well as maintains the transmitting router in the topology database of those that hear the message.
- **Database descriptor packets (DDP)** - Used to send LSA header information to neighbors to synchronize topology databases.
- **Link-state request (LSR)** - Request for more detailed information.
- **Link-state update (LSU)** - LSA packet issued in response to the request for database information in the link-state request packet.
- **Link-state acknowledgment** - Acknowledges the LSU.

Neighbor Database

The neighbor database is created to display all of the states of the OSPFv2 neighbors. This database displays who is the active DR or BDR for a broadcast OSPFv2 network.

Finding a neighbor:

- **The down state -**When first powered on, a router transmits Hello packets.
- **The init state -** The first router waits for a reply (usually four times the length of the hello timer). If on a broadcast network, the router learns the DR and BDR. Another router on that segment adds the router ID to its list and sends out a multicast 224.0.0.5.
- **The two-way state -** The first router sees its own router ID in the list of neighbors, and a neighbor relationship is established. The OSPFv2 status is changed to a two-way state.

Discovering routes:

- **The exstart state -** The router with the highest ID takes priority and becomes the initiator (master/slave) to start the exchange state.
- **The exchange state -** Both routers send out database description packets, changing the state to exchange state. (Networks are referred to as links.) Most of these have been received from other routers (via LSAs). The source of the link information is referred to as the router ID. Each link has an interface ID for the outgoing interface, a link ID, and a metric to state the cost of the path. A DDP is sent. The DDP does not contain all the necessary information. It just contains a summary (enough for the receiving router to determine whether more information is required or whether it already contains that entry in its database.) If the M-bit is set, more information is available and should be requested.
- **The loading state -** Once a router has compared the DDP information with that in its topology table, it can be determined if more information is needed. If so, an LSR is sent out for more information. The LSR prompts the master router to send an LSU packet.
- **The full state** - When these LSRs are received and the databases are updated and synchronized with the DR or BDR, the neighbors are fully adjacent.

The following figure illustrates packet exchanges between neighboring routers.

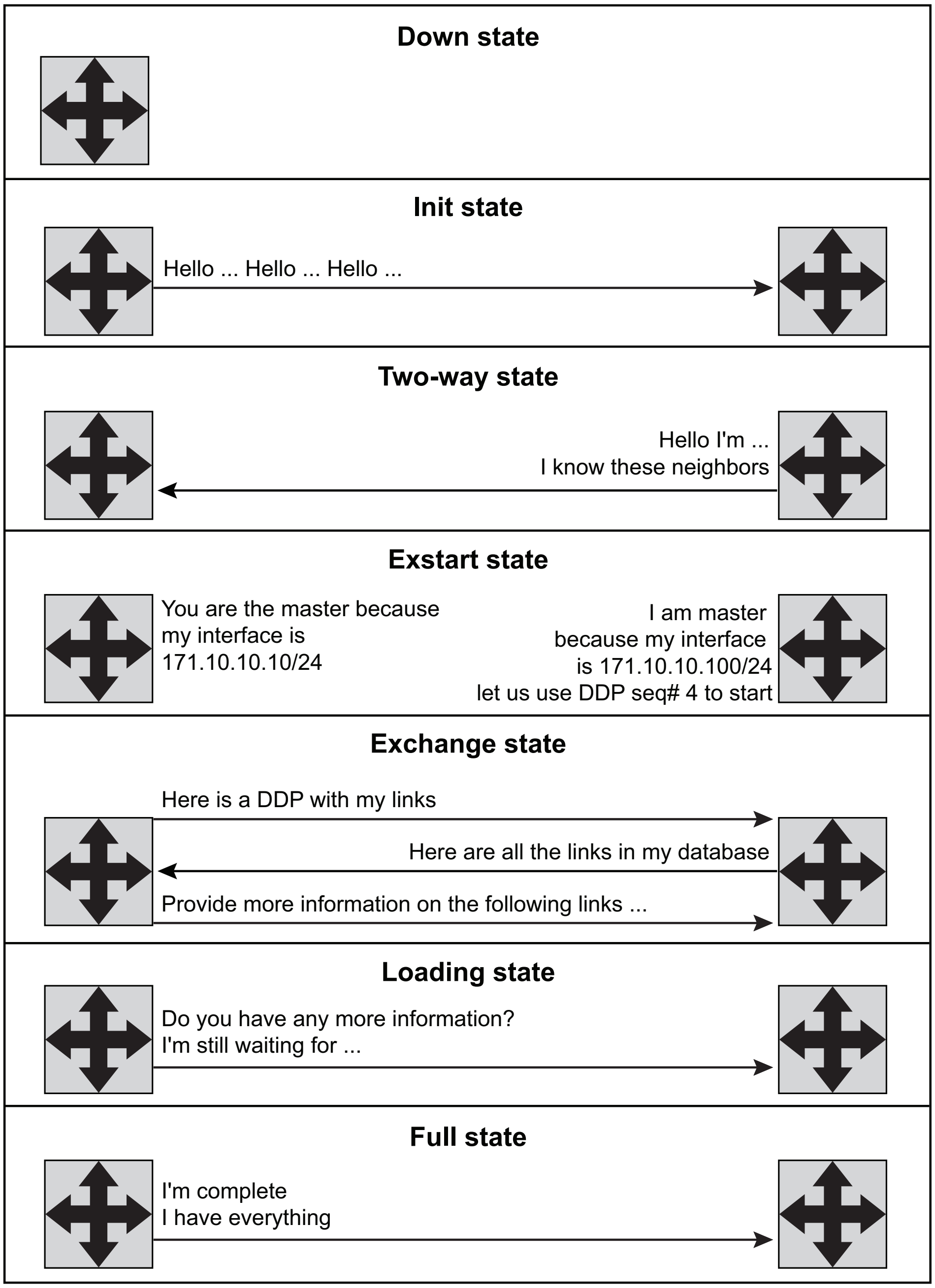

Figure 6. OSPFv2 Neighbor Packet Exchange

OSPFv2 Router Types

Internal Router - A router with all directly connected interfaces belonging to the same area. These routers run a single copy of the OSPFv2 routing algorithm.

Area Border Routers - A router that attaches to multiple areas. ABRs run multiple copies of the basic routing algorithm, one copy for each attached area. ABRs condense the topological information of their attached areas for distribution into the backbone. The backbone in turn distributes the information to other areas.

Backbone Routers - A router that has an interface in the backbone area. This includes ABRs as well as routers with all interfaces connected to the backbone.

Autonomous System Boundary Routers - A router that exchanges routing information with routers attached to other autonomous systems. The ASBR will advertise external routes throughout the autonomous system. Every router in the autonomous system knows the paths to each ASBR. These routers can be either internal or ABR and may or may not participate in the backbone.

Designated Router - The DR on a broadcast network or NBMA network is responsible for maintaining the topology table for the segment. The DR originates a network LSA on behalf of the network. This LSA lists the sets of routers (including the DR and BDR) currently attached to the network. The link-state ID for this LSA is the IP interface address of the DR. The IP network number can then be obtained by the network's subnet/network mask.

This router is elected by the use of a Hello packet and is determined by the highest priority or the router ID. (Set the priority manually using the command **ip ospf priority [0-255]**. A zero value for priority is not a DR or BDR.) All other routers need only peer with the DR, which informs them of any changes on the segment. DRs help relieve the CPUs of other routers. The router that should be chosen for the DR is a centralized router. It should have plenty of processing capabilities. A loopback address is often used to indicate who is the DR for the network.

Backup Designated Router - The BDR is the router on a broadcast or NBMA network that has the second highest priority. This router will take over the responsibilities for the DR if the DR fails. Like the DR, this router is adjacent to all routers on the network. If there is no BDR on a network when a new DR must be established, new adjacencies have to be formed and convergence time increases. This is because part of the adjacency forming process is the synchronizing of link-state databases, which can take a long time. During this time, the network is not available for transit data traffic. The BDR obviates the need to form these adjacencies, since they already exist. This means that the disruption in transit traffic lasts only as long as it takes to flood the new LSAs (which announce the BDR as the DR). Until the BDR is the active DR, it plays a passive role in the flooding process.

OSPFv2 Area Types

Autonomous System - OSPFv2 is an interior routing protocol. All instances of OSPFv2 for an organization are contained within this autonomous systems boundary. Each autonomous system is designated by its own unique identifier.

Area 0.0.0.0 Backbone Area - The backbone area must contain all area border routers. The backbone is responsible for distributing routing information between non-backbone areas. The backbone must be contiguous.

General Areas (Ordinary Area) - General areas are used to segment the network. For large networks, areas are created to decrease the number of LSAs propagated through the network. These advertisements can be limited to an individual area and can be summarized by an ABR to other areas.

Networks with multiple OSPF areas require an IP address scheme. Since the network of an area may be summarized by an ABR, it is essential to verify that the different networks within an area can be summarized easily.

An area must connect to the backbone through an ABR. This area is seen as an entity unto itself. Every router knows about every network in the area, and each router has the same topological database. LSA Types 1 through 5 are supported in this area. However, the routing tables are unique from the perspective of the router and its position in the area.

Stub Areas - This is an area that will not accept ASBR-summary or external routes (Types 4 and 5 LSAs). A router within the area can see outside the autonomous system only via the configuration of a default route. Every router within a stub area can see every network within the area and the networks (summarized or not) within other areas. A stub area is typically used in a hub-and-spoke network design.

Some considerations for a stub area are:

- No external routes are allowed (Type 4 and 5 LSAs)
- No virtual links are allowed
- No redistribution is allowed
- No ASBRs are allowed
- May not be part of Area 0
- Must be configured on all routers in that area.

Not-So-Stubby Areas - NSSAs are used primarily to connect to service providers or when redistribution is required. In most respects, it is the same as the stub area. External routes are not propagated into or out of the area. It does not allow Type 4 or Type 5 LSAs. Although the NSSA is used to connect to a service provider or for redistribution (which requires external routes for OSPFv2), it utilizes another type of LSA known as Type 7.

Totally Stubby Areas - This area does not accept summary LSAs from the other areas or the external summary LSAs from outside the autonomous system. The LSAs blocked are Types 3, 4, and 5. The most common application for this type of area is for remote sites that have few networks and limited connectivity with the rest of the network.

Importance of Addressing Schemes

IP addressing schemes play a vital role in the network design for OSPFv2. As mentioned earlier, the highest router ID and the highest priority determine the DR and the BDR for an area.

A loopback interface is often used in a router to denote a fixed high IP address. If you want a router to be the DR always, setting a high IP address on a loopback interface is a more reliable method of implementation than a standard interface. The loopback interface is not subject to link-state changes, whereas a normal interface might be.

Route Summarization - When dealing with a multi-area network in OSPFv2, routes can be summarized.

There are two types of route summarization:

- **Inter-area Summarization** - This is performed by the ABR and creates Type 3 and 4 LSAs.
- **External Summarization** - This is performed at the ASBR and creates Type 5 LSAs. This summarization route can represent an exterior routing protocol, another interior gateway protocol, or static routes. These routes are often redistributed within an OSPFv2 process.

Both types of route summarization require contiguous IP addressing, and this must be considered when setting up an OSPFv2 network.

OSPFv2 and VRF

OSPFv2 supports per virtual routing and forwarding (VRF) operation, meaning the VRF is specified when creating the OSPFv2 process. All interfaces enabled with this OSPFv2 process must already be a member of that VRF. Routes resolved by a specific OSPFv2 process are placed in the route table belonging to that VRF. Changing the VRF settings at an interface removes all OSPFv2 configuration from the interface. Deleting a

VRF removes all OSPFv2 configurations associated with that VRF globally across the router. *Figure 7* below illustrates the relationships between interfaces, OSPFv2 processes, and VRFs.

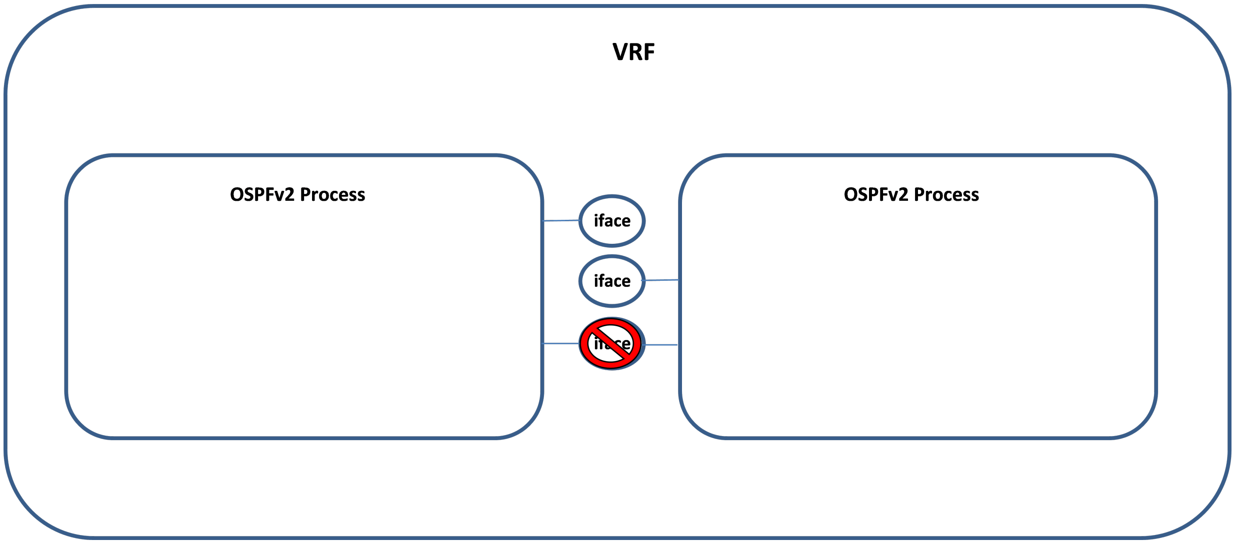

Figure 7. The Relationship Between Interface, OSPFv2 Processes, and VRFs

OSPFv2 Redistribution

Redistribution is supported between OSPFv2 and other sources or targets that are in the same VRF instance. You can redistribute into OSPFv2 or redistribute from OSPFv2. Redistribute commands are used to redistribute routes from another source into OSPFv2.

3. Hardware and Software Requirements and Limitations

OSPFv2 is available on AOS devices using firmware AOS 3.1 or later. For a complete listing of products that support OSPFv2, refer to the *AOS Feature Matrix*, available online at https://supportcommunity.adtran.com.

In AOS firmware R11.3.0 or later, OSPFv2 supports OSPFv2 processes that are not in the default VRF instance. In addition, multiple OSPFv2 processes can be created, whether they are in the same or different VRF instances.

The maximum number of routers and prefixes allowed in a single area is hardware dependent. However, regardless of the hardware used, allocating more than 20 routers or 1000 prefixes to a single area may cause performance degradation.

4. Configuring OSPFv2 in AOS

OSPFv2 is configured in AOS using the CLI. The OSPFv2 commands are separated into three main categories: global settings, area settings, and interface settings. The global OSPFv2 settings are used to specify the OSPFv2 process number, the VRF on which OSPFv2 is operating, and other general OSPFv2 global settings. The area OSPFv2 settings are used to configure settings specific to the area, such as maximum paths, summary prefixes, and redistribution information. The interface OSPFv2 settings include enabling OSPFv2 on the interface, specifying packet intervals, route costs, and area options. The following sections describe the commands used in OSPFv2 configuration.

Accessing the CLI

To access the CLI on your AOS unit, follow these steps:

1. Boot up the unit.
2. Telnet to the unit (**telnet** <*ip address*>), for example:

```
telnet 10.10.10.1.
```

> **NOTE**
>
> *If during the unit's setup process you have changed the default IP address (10.10.10.1), use the configured IP address.*

3. Enter your user name and password at the prompt.

> **NOTE**
>
> *The AOS default user name is **admin** and the default password is **password**. If your product no longer has the default user name and password, contact your system administrator for the appropriate user name and password.*

4. Enable your unit by entering **enable** at the prompt as follows:

```
>enable
```

5. If configured, enter your Enable mode password at the prompt.
6. Enter the unit's Global Configuration mode as follows:

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

Configuring OSPFv2 Global Settings

To configure the global parameters for OSPFv2 in AOS, use the commands described in this section.

1. To enable OSPFv2, create an OSPFv2 process, and enter the router's OSPFv2 Configuration mode, enter the **router ospf [**<*process id*>**] [vrf** <*name*>**]** command from the Global Configuration mode. This command creates a new OSPFv2 process, or allows you to edit a previously configured OSPFv2 process. By default, OSPFv2 is disabled. The <*process id*> parameter creates a process ID for the OSPFv2 process. These IDs are globally unique across the AOS device. The ID value must be unique among all OSPFv2 processes on the device, and it does not correlate or interact with any OSPFv2 processes. Valid ID range is **1** to **65535**. By default, no process ID exists. The optional **vrf** <*name*> parameter specifies a nondefault VRF on which to enable OSPFv2. If the VRF is not specified, OSPFv2 is enabled and configured on the default (unnamed) VRF. Using the **no** form of this command removes the OSPFv2 process and all of its settings at both the global and interface level. To enable OSPFv2 and create an OSPFv2 process on the default VRF instance, enter the command as follows:

```
(config)#router ospf 5
(config-ospf)#
```

2. Specify the cost of the default summary route that is injected into the specified OSPFv2 area (when the area is a stub or NSSA) using the **area** <*area id* | *ipv4 address*> **default-cost** <*cost*> command from the OSPFv2 Configuration mode. This command cannot be applied to the backbone area, since the backbone area cannot be a stub or NSSA. The <*area id* | *ipv4 address*> parameter specifies the ID of the area that receives the default summary route. The area ID can be specified as an integer or as an IPv4 address. Valid integer area ID range is **1** to **4294967295**. Enter IPv4 addresses in dotted decimal notation (for example, **10.10.10.1**). The <*cost*> parameter is the advertised cost of the default summary route. Valid cost range is **0** to **16777214**. By default, the cost is set to **0**. Using the **no** form of this command

returns the cost of the summary route to the default value. To specify the cost of the default summary route, enter the command from the OSPFv2 Configuration mode as follows:

```
(config-ospf)#area 10 default-cost 100
(config-ospf)#
```

3. Control the route summarization between OSPFv2 areas (inter-area prefixes of Type 3 LSAs) using the **area** <*area id* | *ipv4 address*> **range** <*ipv4 address*> <*subnet mask*> **[advertise | not-advertise]** command from the OSPFv2 Configuration mode. By default, route summarization is disabled. Using the **no** form of this command removes the specified summarization, and returns to advertisement of individual summary prefixes between areas. The <*area id* | *ipv4 address*> parameter specifies the ID of the area of which the router is a member and that contains the prefixes being summarized. The summaries are advertised into other areas of which this router is also a member. The area ID can be specified as an integer or as an IPv4 address. Valid integer area ID range is **1** to **4294967295**. Enter IPv4 addresses in dotted decimal notation (for example, **10.10.10.1**). The <*ipv4 address*> parameter is the IPv4 address to be advertised. The <*subnet mask*> parameter specifies the subnet mask that corresponds to a range of IPv4 addresses (network) or a specific host. Subnet masks can be expressed in dotted decimal notation (for example, **255.255.255.0**). The optional **advertise** and **not-advertise** parameters specify whether the summary will or will not be advertised to other areas. The summary is advertised by default, unless otherwise specified.

   To configure the route summarization between OSPFv2 areas, enter the command as follows:

```
(config-ospf)#area 11.0.0.0 range 11.0.0.0 255.0.0.0 advertise
(config-ospf)#
```

4. Specify the area type as a stub or total stub using the **area** <*area id* | *ipv4 address*> **stub [no-summary]** command from the OSPFv2 Configuration mode. When an area is specified as a stub, the area border router withholds inter-area-router-LSAs and autonomous system-external-LSAs, and instead injects a default summary route. When an area is a total stub, the area border router also withholds inter-area-prefix-LSAs. By default, the area type is normal, and is not specified as a stub. The backbone area cannot be used as a stub. Using the **no** form of this command returns the area to the default area type. The <*area id* | *ipv4 address*> parameter specifies the ID of the area that you are defining. The area ID can be specified as an integer or an IPv4 address. Valid integer area ID range is **1** to **4294967295**. Enter IPv4 addresses in dotted decimal notation (for example, **10.10.10.1**). The optional **no-summary** parameter specifies the area is a total stub. To specify the area as a total stub, enter the command as follows:

```
(config-ospf)#area 10 stub no-summary
(config-ospf)#
```

NOTE

All routers connected to an area must have the same configuration for the area type.

5. Specify the reference value used to calculate the OSPFv2 cost of an interface using the **auto-cost reference-bandwidth** <*bandwidth*> command from the OSPFv2 Configuration mode. In OSPFv2, the cost of an interface is the reference value divided by the interface's bandwidth. Certain conditions require a change to the system's reference value, such as an increase in interface speeds. The default reference of 100 Mbps can become ineffective since 1 is the lowest possible cost value. This command allows the reference value to be customized for your specific network design. All OSPFv2 participants in the network should use the same reference value. The <*bandwidth*> parameter is the reference bandwidth used to calculate the OSPFv2 cost on the AOS device's OSPFv2 interfaces. Valid range is **1** to **4294967** Mbps. By default, the reference value is set to **100** Mbps. Using the **no** form of this command returns the reference to the default value. To change the default value, enter the command as follows:

```
(config-ospf)#auto-cost reference-bandwidth 400
```

```
(config-ospf)#
```

> **NOTE**
>
> *All routers in the OSPFv2 autonomous system need to have the same reference-bandwidth configured otherwise routing loops and non-optimal routing paths can occur.*

6. Specify the administrative distance for OSPFv2 routes using the **distance [ospf external | ospf inter-area | ospf intra-area]** <*distance*> command from the OSPFv2 Configuration mode. The distance can be set once for all OSPFv2 route types or individually for each route type. The optional **ospf external**, **ospf inter-area**, and **ospf intra-area** parameters allow you to specify the administrative distance for routes injected into the route table from the SPF algorithm (external, inter-area, or intra-area). The <*distance*> parameter specifies the administrative distance to be used by the route type (whether all routes or routes in a specified area). Valid distance range is **0** to **255**. By default, all routes have an administrative distance of **110**. Using the **no** form of this command returns the administrative distance to the default value.

   To change the administrative distance for all OSPFv2 routes, enter the command as follows:

   ```
   (config-ospf)#distance 150
   (config-ospf)#
   ```

7. Specify the maximum number of equal cost routes to a given prefix that OSPFv2 can enter into the route table using the **maximum-paths** <*value*> command from the OSPFv2 Configuration mode. The <*value*> parameter specifies the maximum number of equal cost routes. Valid range is **1** to **6**, with a default value of **4**. Using the **no** form of this command returns the maximum paths to the default value. To change the maximum number of equal cost routes, enter the command as follows:

   ```
   (config-ospf)#maximum-paths 5
   (config-ospf)#
   ```

8. Enable routing on an IP stack and specify OSPFv2 area IDs for the interface on which OSPFv2 will run using the **network** <*ipv4 address*> <*wildcard mask*> **area** <*area id* **|** *ipv4 address*> command from the OSPFv2 Configuration mode. In order for OSPFv2 to operate on an interface, the primary address for the interface must be included in the **network area** command. Assigning an interface to an OSPFv2 area is done using the **network area** command. There is no limit to the number of **network area** commands used on a router. If the address ranges defined for different areas overlap, the first area in the **network area** command list is used and all other overlapping portions are disregarded. Try to avoid overlapping to avoid complications. If the network overlaps with another OSPF process the first one matched in the running config is the OSPF process and network statement chosen. The <*ipv4 address*> parameter specifies a valid IPv4 address. Enter IPv4 addresses in dotted decimal notation (for example, **10.10.10.1**). The <*wildcard mask*> parameter specifies the wildcard mask that corresponds to a range of IPv4 addresses. Wildcard masks are expressed in dotted decimal notation (for example, **0.0.0.255**). The **area** <*area id* **|** *ipv4 address*> parameter specifies the ID of the area that you are defining. The area ID can be specified as an integer or an IPv4 address. Valid integer area ID range is **1** to **4294967295**. Enter IPv4 addresses in dotted decimal notation (for example, **10.10.10.1**). To define an OSPFv2 area, enter the command as follows:

   ```
   (config-ospf)#network 192.22.72.101 0.0.0.255 area 0
   (config-ospf)#
   ```

9. Specify that the algorithm defined in RFC 1583 is used to determine AS external routes using the **rfc1583compatibility** command from the OSPFv2 Configuration mode. Use the **no** form of this command to specify that the algorithm defined in RFC 2328 is used to determine AS external routes. With the introduction of RFC 2328, the algorithm used to determine the cost of AS external routes was changed to help minimize the occurrence of routing loops. This command should be set identically on all routers in the OSPFv2 network. Otherwise, the summary route from area boundary routers (ABRs) using RFC 1583 will

be preferentially chosen because the summary routes they advertise will have a lower cost. By default, RFC 1583 compatibility is enabled. To specify that the algorithm defined in FRC 2328 is used to determine AS external routes, enter the command as follows:

```
(config-ospf)#no rfc1583compatibility
(config-ospf)#
```

10. Specify the value to be used by the OSPFv2 process as the router ID using the **router-id** <*ipv4 address*> command from the OSPFv2 Configuration mode. An OSPFv2 router ID is selected using an algorithm. The default router ID value is chosen by first looking at the configured router ID (if there is one), then the highest value IPv4 address assigned to a loopback interface in the same VRF, and then the highest value IPv4 address assigned to a non-loopback interface. The <*ipv4 address*> parameter of this command is a 32-bit value (represented in an IPv4 address format), that is used by this specific OSPFv2 instance as the router ID. The value must be unique for the OSPFv2 domain to which the OSPFv2 process belongs. Though the value uses an IPv4 address in general format, it does not actually use the IPv4 address. IPv4 addresses should be entered in dotted decimal notation (for example, **10.10.10.1**). Valid IPv4 address range for this command is **0.0.0.1** to **255.255.255.255**. Using the **no** form of this command returns the router ID for this OSPFv2 process to the default (which is determined as explained previously). To specify a router ID for this OSPFv2 process, enter the command as follows:

```
(config-ospf)#router-id 10.10.10.2
(config-ospf)#
```

11. Control route summarization and route advertisement that is redistributed into this OSPFv2 process (external prefixes of Type 5 LSAs) using the **summary-address** <*ipv4 address*> <*subnet mask*> **[not-advertise] [tag** <*value*>**]** command from the OSPFv2 Configuration mode. By default, the cost applied to the summary route is that of the lowest cost route in the set it summarizes. The command can be entered multiple times to summarize different addresses. If at least one route being summarized would bypass the route map, then the resulting summary will be created even though the route map could deny other routes which would have also created the same summary. However, if all of the routes being summarized are blocked, then no summary is generated. In addition, if multiple summaries exist where one summary subsumes another, the prefix with the shortest length is used. The <*ipv4 address*> specifies the IP address of the route being redistributed into the OSPFv2 area of which the router is a member, and that contains the prefixes being summarized. Enter IPv4 addresses in dotted decimal notation (for example, **10.10.10.1**). The <*subnet mask*> parameter specifies the subnet mask that corresponds to a range of IPv4 addresses (network) or a specific host. Subnet masks can be expressed in dotted decimal notation (for example, **255.255.255.0**).The optional **not-advertise** parameter specifies that the summary will not be advertised to other OSPFv2 areas. The optional **tag** <*value*> parameter specifies the route tag for the summarized route. Route tags can be used to implement routing policies using route maps. For example, they can be used to control route redistribution to prevent routing loops when configuring multipoint route redistribution (i.e. routes are redistributed from protocol A into protocol B, then back into protocol A). Route tags only apply to external (Type 5) LSAs. Use the **no** form of this command to remove the specified route summary, and return to advertising the individual prefixes that are being redistributed. To create a route summary, enter the command as follows:

```
(config-ospf)#summary-address 11.0.0.0 255.0.0.0 not-advertise
(config-ospf)#
```

12. Specify the OSPFv2 shortest path first (SPF) calculation and hold interval timers using the **timers spf** <*delay*> **[**<*hold*>**]** command from the OSPFv2 Configuration mode. The <*delay*> parameter specifies the time (in seconds) between receipt of OSPFv2 topology changes and the beginning of SPF calculations. Valid range is **0** to **65535** seconds, with a default value of **5** seconds. The optional <*hold*> parameter specifies the time (in seconds) between consecutive SPF calculations. Valid range is **0** to **65535** seconds, with a default value of **10** seconds. Using the **no** form of this command returns the SPF timers to the default value. To change the SPF timers, enter the command as follows:

```
(config-ospf)#timers spf 3 7
(config-ospf)#
```

Configuring OSPFv2 Redistribution Settings

Redistribute commands are used to redistribute routes from another source protocol into the OSPFv2 process. Redistributed routes can be tagged using route tags and filtered using route maps (for more information on using route maps, refer to . To configure the OSPFv2 redistribution settings, use the commands described in this section.

1. Specify that a default external route is injected into this OSPFv2 process using the **default-information originate [always] [metric** *<value>***] [metric-type** *<type>***] [tag** *<value>***]** command from the OSPFv2 Configuration mode. A router that injects an external default route becomes an OSPFv2 ASBR because it sources routes from outside the OSPFv2 AS, even if the ASBR itself lacks a default route. The optional **always** parameter allows the ASBR to inject a default route into the OSPFv2 AS. The optional **metric** *<value>* parameter assigns an OSPFv2 metric value to the route being injected into OSPFv2. If the metric is not specified, then the **default-metric** command is used for the default route metric (refer to *Step 2* below). If the **default-metric** command is not configured, then the metric of **10** is used. If the metric is specified as **0** using this command, it means that the metric is set to an unconfigured value, and the **default-metric** command setting is used. Valid metric range is **0** to **16777214**, with a default value of **10**. The optional **metric-type** *<type>* parameter specifies the external metric type for the route being injected into OSPFv2 by this command. Metric type **1** specifies that when external routes are assigned a metric they begin with the metric value specified by this command, and add the cost of the OSPFv2 path as they are advertised throughout the AS. Metric type **2** is not affected by the OSPFv2 path cost, and retains the original metric values. By default, metric type **2** is used. The optional **tag** *<value>* parameter specifies the route tag for the default external route being injected. Route tags can be used to implement routing policies using route maps. For example, they can be used to control route redistribution to prevent routing loops when configuring multipoint route redistribution (i.e., routes are redistributed from protocol A into protocol B, then back into protocol A). Route tags only apply to external (Type 5) LSAs. These parameters can be entered in any order. By default, no external route is injected into OSPFv2. Using the **no** form of this command removes the default route. To specify an external route is injected into OSPFv2, using the default metric of **10** and the default metric type of **2**, enter the command as follows:

```
(config-ospf)#default-information originate always
(config-ospf)#
```

2. Specify the metric value used for redistributed routes when the value is not otherwise specified by entering the **default-metric** *<value>* command from the OSPFv2 Configuration mode. The metric for redistributed routes can be specified using the **redistribute** command (refer to *Configuring OSPFv2 Interface Settings on page 23*), or using a route map. When the value is not set in one of the locations, the default metric is used. This setting does not affect the metric of the default route injected using the **default-information originate** command, unless the metric is not specified in the **default-information originate** command. The *<value>* parameter specifies the OSPFv2 metric value assigned to the route being injected into OSPFv2. Valid range is **0** to **6777214**. By default, the metric is set to **20**. Using the **no** form of this command returns the default metric to the default value. To change the default metric value, enter the command as follows:

```
(config-ospf)#default-metric 30
(config-ospf)#
```

3. To redistribute routes from a specified source into the OSPFv2 process, enter the **redistribute [bgp | connected | ospf** *<process id>* **[no-include-connected] | rip | static] [metric** *<value>***] [metric-type** *<type>***] [route-map** *<name>***] [tag** *<value>***] subnets** command from the OSPFv2 Configuration mode. A router that performs redistribution becomes an OSPFv2 ASBR, because it is sourcing routes from outside the OSPFv2 AS. The specified source from which routes are redistributed must be in the same VRF as the OSPFv2 instance being configured. This command can be entered multiple times (at most, once for each source). Reentering the command with the same source replaces any existing command with that source. The **bgp**, **connected**, **rip**, and **static** parameters define the source from which routes are redistributed. The **ospf** *<process id>* parameter specifies the OSPFv2 process from which to redistribute routes. The optional **no-include-connected** parameter specifies that prefixes of the interface running this

source protocol are not automatically included in the route redistribution. When this parameter is not used, all connected routes on interfaces participating in the OSPFv2 process are also redistributed. The optional **metric** <*value*> parameter assigns an OSPFv2 metric value to the route being redistributed into OSPFv2. If the metric is not specified, then the **default-metric** command setting is used for the default route metric (refer to *Step 2 on page 21*). If the **default-metric** command is not configured, then the metric of **20** is used. If the metric is specified as **0** using this command, it means that the metric is set to an unconfigured value, and the **default-metric** command setting is used. Valid metric range is **0** to **16777214**, with a default value set by the **default-metric** command. The optional **metric-type** <*type*> parameter specifies the external metric type for the route being redistributed into OSPFv2 by this command. Metric type **1** specifies that when external routes are assigned a metric they begin with the metric value specified by this command, and add the cost of the OSPFv2 path as they are advertised throughout the AS. Metric type **2** is not affected by the OSPFv2 path cost, and retains the original metric values. By default, metric type **2** is used. The optional **route-map** <*name*> parameter specifies a route map that is applied to routes being redistributed by this command. A route map can impose granular control on routes being redistributed. For more information on using route maps with OSPFv2 refer to *Using Route Maps with OSPFv2 on page 26*. The optional **tag** <*value*> parameter specifies the route tag for the redistributed routes. Route tags can be used to implement routing policies using route maps. For example, they can be used to control route redistribution to prevent routing loops when configuring multipoint route redistribution (i.e., routes are redistributed from protocol A into protocol B, then back into protocol A). Route tags only apply to external (Type 5) LSAs. The **subnets** parameter specifies subnet redistribution when redistributing routes into OSPFv2. These parameters can be entered in any order.

When using redistribution, keep in mind the following:

- OSPFv2 settings, such as tag, metric and, metric types, that are specified in a route map entry override those settings in the **redistribution** command for the routes that match the route map entry.
- When using the **summary-address** command (refer to *Configuring OSPFv2 Interface Settings on page 23*), if at least one route being summarized would be allowed by the route map, then the resulting summary will be created even though the route-map could deny other routes that would have also created the same summary. However, if all of the routes being summarized are blocked, then no summary is generated.
- By default, no route map is used.

By default, no routes are redistributed into the OSPFv2 process. Using the **no** form of this command removes the redistribution from the specified source. To redistribute routes from another source, enter the command as follows:

```
(config-ospf)#redistribute bgp
(config-ospf)#
```

4. Apply a standard IPv4 access control list (ACL) to all routes that are redistributed into the OSPFv2 network using the **distribute-list** <*ipv4 acl name*> **out** command. This command allows you to filter what routes gets redistributed into the OSPFv2 network. For example, if the OSPFv2 process is configured to redistribute RIP routes (using the **redistribute rip** command), and RIP has a route to 10.22.8.0/24, OSPFv2 would normally generate a Type 5 external LSA to distribute the routes through the AS. However, if an ACL is applied that blocks the 10.22.8.0/24 prefix, the external LSA will not be generated and the routes will not be distributed through the AS. The <*ipv4 acl name*> parameter specifies an IPv4 ACL name. This is a standard IPv4 ACL against which the redistributed routes are matched. By default, no distribute list filters exist for the OSPFv2 routing process. To apply an IPv4 ACL filter to routes redistributed into the OSPFv2 process, enter the command as follows:

```
(config-ospf)#distribute-list RIP out
(config-ospf)#
```

5. Apply a route map to filter the routes added to the route table by OSPFv2 using the **distribute-list route-map** <*name*> **in**. This commands allows you to filter what OSPFv2 is allowed to add to the route table. For example, if OSPF calculations determine that a route to 10.22.8.0/24 should be

added to the route table, the route map matching rules are applied to the prefix to determine whether the route should be added to the route table. This command does not affect the distribution of LSAs or the LS database (every router in the same area should still have the same database for that area). The <*name*> parameter specifies a route map name. By default, no distribute list filters exist for the OSPFv2 routing process. To apply a route map to filter the routes added to the OSPFv2 route table, enter the command as follows:

```
(config-ospf)#distribute-list route-map OSPF-IN in
(config-ospf)#
```

Configuring OSPFv2 Interface Settings

After configuring the global and redistribution settings, you must configure the OSPFv2 interface. When configuring the interface, you will enable OSPFv2 on the interface, associate the interface with an OSPFv2 process, specify an OSPFv2 cost for the interface, and configure other OSPFv2 settings that are specific to the interface. Each OSPFv2 interface configuration requires a process ID, which allows you to match the OSPFv2 settings you are configuring with an OSPFv2 process that is defined at the global level (refer to *Configuring OSPFv2 Global Settings on page 17*). OSPFv2 interface configuration commands are described below.

1. Add an interface to an OSPFv2 process using the **ip ospf** <*process id*> **area** <*area id* | *ipv4 address*> command from the interface's configuration mode. This command places the interface in the specified area. The <*process id*> parameter specifies the OSPFv2 routing process this interface is to join. The process ID is locally significant to the device. This command can also be used to create the OSPFv2 process instance (rather than with the global configuration command), but the value must be unique among all OSPFv2 processes on the device. Valid process ID range is **1** to **65535**. By default, a process ID is not defined. The **area** <*area id* | *ipv4 address*> parameter specifies the ID of the area that you are defining. The area ID can be specified as an integer or an IPv4 address. Valid integer area ID range is **1** to **4294967295**. Enter IPv4 addresses in dotted decimal notation (for example, **10.10.10.1**). By default, interfaces are not configured to be part of an OSPFv2 process. Using the **no** form of this command removes the interface from the OSPFv2 process. The following rules apply to this command:
   - An IP address must be assigned to the interface using the **ip address** command.
   - The interface and the specified OSPFv2 process (if defined in the global configuration) must be in the same VRF or the command will fail.
   - The interface must not already be covered by a network statement of another OSPFv2 process in the same VRF.

   > **NOTE**
   >
   > *This command overrides any network command in the OSPF process and forces this interface to be part of that OSPF process*

   To add an interface to the OSPFv2 process, enter the interface's configuration mode and enter the command as follows:

   ```
   (config-eth 0/1)#ip ospf 5 area 10
   (config-eth 0/1)#
   ```

2. Enable and specify the type of OSPFv2 authentication used on an interface using the **ip ospf** <*process id*> **authentication [message-digest | null]** command. By default, no authentication is used. The <*process id*> parameter specifies the OSPFv2 routing process to which this interface belongs. Valid process ID range is **1** to **65535**. The optional **message-digest** parameter specifies the **message-digest** as the authentication type. The **null** parameter specifies that no OSPFv2 authentication is used on the

interface. To configure authentication for OSPFv2 messages on the interface, enter the command from the interface's configuration mode:

```
(config-eth 0/1)#ip ospf 5 authentication message-digest
(config-eth 0/1)#
```

3. Assign a simple-text authentication password to be used by other routers using OSPFv2 simple-text password authentication using the **ip ospf** *<process id>* **authentication-key** *<password>* command. By default, no authentication is used. The *<process id>* parameter specifies the OSPFv2 routing process to which this interface belongs. Valid process ID range is **1** to **65535**. The **authentication-key** *<password>* parameter specifies the simple-text password. To configure simple-text authentication on the interface, enter the command from the interface's configuration mode:

```
(config-eth 0/1)#ip ospf 5 authentication-key password
(config-eth 0/1)#
```

4. Specify the OSPFv2 cost of the interface using the **ip ospf** *<process id>* **cost** *<cost>* command from the interface's configuration mode. This command specifies a value that represents the cost of sending a packet over the interface. By default, the cost of the interface is automatically computed. The automatic cost computation formula is the reference bandwidth divided by the interface bandwidth. The reference bandwidth is set using the **auto-cost reference-bandwidth** command (refer to *Step 5 on page 18*) and defaults to 100 Mbps. The *<process id>* parameter specifies the OSPFv2 routing process to which this interface belongs. Valid process ID range is **1** to **65535**. The *<cost>* parameter is the OSPFv2 cost of the interface. This value overrides any computed cost value. Valid range is **1** to **65535**. Use the **no** form of this command to return the cost to the default value. To change the OSPFv2 cost of the interface, enter the command as follows:

```
(config-eth 0/1)#ip ospf 5 cost 10
(config-eth 0/1)#
```

5. Specify the maximum interval allowed between OSPFv2 Hello packets on this interface by entering the **ip ospf** *<process id>* **dead-interval** *<value>* command from the interface's configuration mode. If the maximum interval is exceeded, the device determines that the neighbor is down. This value must be the same across all interfaces on a link. The *<process id>* parameter specifies the OSPFv2 routing process to which this interface belongs. Valid process ID range is **1** to **65535**. If the process ID has not already been created, entering this command will not create the process. The *<value>* parameter is the maximum number of seconds allowed between OSPFv2 Hello packets. It is recommended that this value be 4 times the Hello packet interval value. Valid range is **1** to **65535** seconds, with a default value of **40** seconds. Using the **no** form of this command returns to the dead interval to the default value. To change the dead interval on the interface, enter the command as follows:

```
(config-eth 0/1)#ip ospf 5 dead-interval 100
(config-eth 0/1)#
```

6. Specify the interval between OSPFv2 Hello packets sent on this interface by entering the **ip ospf** *<process id>* **hello-interval** *<value>* command from the interface's configuration mode. This value must be the same across all interfaces on the link. The *<process id>* parameter specifies the OSPFv2 routing process to which this interface belongs. Valid process ID range is **1** to **65535**. The *<value>* parameter is the number of seconds allowed between OSPFv2 Hello packets. Valid range is **1** to **65535** seconds, with a default value of **10** seconds. Using the **no** form of this command returns to the Hello packet interval to the default value. To change the Hello packet interval on the interface, enter the command as follows:

```
(config-eth 0/1)#ip ospf 5 hello-interval 20
(config-eth 0/1)#
```

7. Specify the OSPFv2 message digest 5 (MD5) authentication (16 byte maximum) keys using the **ip ospf** *<process id>* **message-digest-key [1 | 2] md5** *<key>* command. By default, no authentication is used. The *<process id>* parameter specifies the OSPFv2 routing process to which this interface belongs. Valid process ID range is **1** to **65535**. The **1** and **2** parameters specifies the key ID number. The **md5** *<key>*

parameter specifies the OSPFv2 MD5 authentication key. To configure OSPFv2 MD5 authentication on the interface, enter the command from the interface's configuration mode:

```
(config-eth 0/1)#ip ospf 5 message-digest-key 1 md5 password
(config-eth 0/1)#
```

8. Specify the OSPFv2 network type for the interface by entering the **ip ospf** <*process id*> **network [broadcast | point-to-point]** command from the interface's configuration mode. The <*process id*> parameter specifies the OSPFv2 routing process to which this interface belongs. Valid process ID range is **1** to **65535**. The **broadcast** parameter sets the OSPFv2 network type to broadcast, and the **point-to-point** parameter sets the OSPFv2 network type to point-to-point. By default, Ethernet interfaces are set to broadcast, and PPP, Frame Relay, and loopback interfaces are set to point-to-point. Using the **no** form of this command returns the interface's network type to the default value. To change the interface's network type, enter the command as follows:

```
(config-eth 0/1)#ip ospf 5 network point-to-point
(config-eth 0/1)#
```

9. Specify the interface OSPFv2 priority by entering the **ip ospf** <*process id*> **priority** <*value*> command from the interface's configuration mode. Priority is used in the election of the DR and the BDR on multi-access networks. Interfaces connected to multi-access networks (such as Ethernet interfaces) perform an election for a DR and BDR. The router interface with the highest OSPFv2 priority on the link becomes the DR for that link. The interface with the next highest priority becomes the designated backup router. In the event there is a tie, the router interface with the highest router ID takes precedence. A value of **0** indicates the router is ineligible to become either the DR or BDR. The <*process id*> parameter specifies the OSPFv2 routing process to which this interface belongs. Valid process ID range is **1** to **65535**. The <*value*> parameter specifies the OSPFv2 priority of the interface. Valid range is **0** to **255**, with a default value of **1**. Using the **no** form of this command returns the interface's priority to the default value. To change the interface's priority, enter the command as follows:

```
(config-eth 0/1)#ip ospf 5 priority 6
(config-eth 0/1)#
```

10. Specify the interval between unacknowledged OSPFv2 LSAs sent on the interface by entering the **ip ospf** <*process id*> **retransmit-interval** <*value*> command from the interface's configuration mode. The <*process id*> parameter specifies the OSPFv2 routing process to which this interface belongs. Valid process ID range is **1** to **65535**. If the process ID has not already been created, entering this command will not create the process. The <*value*> parameter specifies the number of seconds between OSPFv2 LSAs sent on the interface. Valid range is **1** to **65535** seconds, with a default value of **5** seconds. Using the **no** form of this command returns the LSA interval to the default value. To change the interface's LSA interval, enter the command as follows:

```
(config-eth 0/1)#ip ospf 5 retransmit-interval 10
(config-eth 0/1)#
```

11. Specify the estimated time that is required to propagate an LSA on the interface by entering the **ip ospf** <*process id*> **transmit-delay** <*value*> command from the interface's configuration mode. The <*process id*> parameter specifies the OSPFv2 routing process to which this interface belongs. Valid process ID range is **1** to **65535**. The <*value*> parameter specifies the number of seconds required to send LSAs from the interface. Valid range is **1** to **65535** seconds, with a default value of **1** second. Using the **no** form of this command returns the transmit delay to the default value. To change the interface's transmit delay, enter the command as follows:

```
(config-eth 0/1)#ip ospf 5 transmit-delay 2
(config-eth 0/1)#
```

12. Disable an OSPFv2 process on the interface by entering the **ip ospf** <*process id*> **shutdown** command from the interface's configuration mode. When this command is used, the OSPFv2 commands remain in place, but logically it appears to the interface as though the OSPFv2 process was removed from the configuration. This command is beneficial when troubleshooting. The <*process id*> parameter specifies

the OSPFv2 routing process to shut down. Valid process ID range is **1** to **65535**. If the process ID has not already been created, entering this command will not create the process. Using the **no** form of this command reinstates the OSPFv2 process on the interface. To shutdown an OSPFv2 process on an interface, enter the command as follows:

```
(config-eth 0/1)#ip ospf 5 shutdown
(config-eth 0/1)#
```

Using Route Maps with OSPFv2

Route maps allow configuration of complex routing policies. In addition to filtering routes according to network address and prefix length, routes can be filtered according to their metric value, or route tag.

Route maps can also be configured to apply various attributes to the routes it filters. A route map applied to outbound data determines how the router advertises routes to a neighbor. The **outbound** route map can be configured to perform such tasks as:

- Define the routes that the router can advertise according to specified attributes or prefixes
- Apply attributes to matched routes

Route maps can also be applied to inbound routes to determine which routes are added to the route table by OSPFv2.

The route map itself is created first. Matching criteria and attributes are defined within the Route Map Configuration mode. Once a route map has been established, it can be assigned to one or more of the following:

- Redistributed routes of a single type (using the **redistribute** command)
- The route table (using the **distribute-list in** command)

Use the steps provided in this section to create a route map. For more information on creating and using route maps, refer to *Policy-based Routing* configuration guide, available online at https://supportcommunity.adtran.com.

Step 1: Creating a Route Map

Use the **route-map** *<name>* **[deny | permit]** *<number>* command to create a route map and enter the Route Map Configuration mode. The *<name>* parameter specifies a name for the route map. The **deny** parameter specifies not to process routes matching the specified route map attributes. The **permit** parameter specifies to processes routes matching the specified route map attributes. The *<number>* parameter specifies a sequence number for this route entry. Range is **1** to **4294967295**. To create a route map and enter the Route Map Configuration mode, enter the command as follows:

```
(config)#route-map OSPF-MAP permit 27
```

NOTE

*After creating a route map, route map attributes can be defined from the Route Map Configuration mode. Enter **?** at the **(config-route-map)#** prompt to explore the available options.*

Step 2: Defining Routes and Attributes to be Advertised and Filtered

Define the routes and attributes to be matched by the route map. OSPFv2 routes can be matched according to the following attributes:

- Network address and prefix length
- Metric

- Route tag

The match criteria for routes are specified using a **match** command in a route map entry. *Table 5* below provides a list of available filtering options for OSPFv2.

Table 5. Defining Routes to Be Filtered

Filtering According To	Command Syntax
Network address and/or prefix length	**match ip address prefix-list** <*name*>
ACL	**match ip address** <*ipv4 acl name*>
Metric	**match metric** <*value*>
Route Tag	**match tag** <*value*>

Detailed explanations of the above **match** commands begin with *A. Filtering OSPFv2 Routes Using Prefix Lists on page 27*.

If an attribute is to be applied to the route, then a **set** command must be entered in addition to the **match** command. Attributes are applied to the routes selected by the **match** command. The following attributes can be applied to redistributed routes:

- Metric
- Metric-type
- Route Tag

Detailed explanations of the **set** command attributes begin with *Step 3: Applying Attributes to Matched Routes on page 29*.

NOTE

*If the only action required is filtering of routes, then a **match** command can be used without a **set** command.*

A. Filtering OSPFv2 Routes Using Prefix Lists

One way to use route maps to filter routes is according to the network address and/or prefix length. First, a prefix list is created to define the routes that are to be filtered by the OSPFv2 process. An exact route can be specified or a range of prefix lengths for routes to variable length subnets. (refer to *C. Specifying a Route Tag for Matched Routes on page 30*).

1. Use the **ip prefix-list** <*name*> **seq** <*number*> **[deny | permit]** <*network ip/length*> **[ge | le]** <*value*> <*value*> command from the Global Configuration mode to specify a prefix to be matched or a range of mask lengths. The <*name*> parameter specifies the name of the list. Up to **80** characters are allowed in a name. The **seq** <*number*> parameter specifies the entry's unique sequence number that determines the processing order. Lower numbered entries are processed first. Range is **1** to **4294967294**. The **deny** <*network ip/length*> parameter denies access to entries matching the specified network IPv4 address and the corresponding network prefix length (for example, **10.10.10.0 /24**).The **permit** <*network ip/length*> parameter permits access to entries matching the specified network IPv4 address and the corresponding network prefix length (for example, **10.10.10.0 /24**). The **ge** <*value*> parameter specifies the lower end of

the range. Range is **0** to **32**. The **le** <value> parameter specifies the upper end of the range. Range is **0** to **32**.

> **NOTE**
>
> *If the network address is entered without specifying a range for prefix lengths, the router assumes that the route must be an exact match. For example, if the command* ***ip prefix-list TEST seq 5 permit 10.1.0.0/16*** *is entered, the OSPFv2 interface will only accept routes to the entire 10.1.0.0 /16 subnet. It will not accept routes to a network, such as 10.1.1.0 /24, which was subdivided from the /16 network.*

2. Routes to subnets within the larger network can be permitted or denied by specifying the permitted range of prefix lengths. For example, the filter could allow all routes to subnets in the 10.1.0.0 /16 network with a prefix length up to and including 24:

```
(config)#ip prefix-list OSPF-LIST seq 5 permit 10.1.0.0/16 ge 16 le 24
```

> **NOTE**
>
> *The* ***ge*** *keyword indicates that the length must be* ***greater than or equal to*** *that specified in order to match. The* ***le*** *keyword indicates that the length must be* ***less than or equal to*** *that specified in order to match. If* ***ge*** *is only specified, the router assumes 32 as the upper limit. If* ***le*** *is only specified, the router assumes the network address's length as the lower limit.*

A filter that exactly matches a prefix length can be created by entering the length for both the **ge** and **le** values. For example, the filter could allow any routes to a /24 subnet in the 10.1.0.0 /16 range, but not accept a route to the entire 10.1.0.0 /16 network:

```
(config)#ip prefix-list OSPF-LIST seq 5 permit 10.1.0.0/16 ge 24 le 24
```

3. After configuring the prefix list, use the **match ip address prefix-list** <*name*> command to configure a route map to route traffic based on the defined IPv4 prefix list route filter. The <*name*> parameter specifies the name of the prefix list. Enter the command from the Route Map Configuration mode as follows:

```
(config-route-map)#match ip address prefix-list OSPF-LIST
```

B. Filtering OSPFv2 Routes Using ACLs

Another way to use route maps to filter routes according to network address is by using standard IPv4 ACLs.

1. Create an ACL to define the routes that are to be filtered. For information on how to create a standard IPv4 ACL, refer to *IPv4 ACLs in AOS*, available online at https://supportcommunity.adtran.com.

2. After the ACL has been configured, it can be referenced in a route map entry or used to filter all redistributed routes using the **distribute-list** command (refer to *OSPFv2 Command Summary on page 36*). Use the **match ip address** <*ipv4 acl name*> command to configure a route map to match traffic based on an IPv4 ACL name defined with the **ip access-list** command. The <*ipv4 acl name*> parameter specifies the name of the IPv4 ACL to match. Enter the command from the Route Map Configuration mode as follows:

```
(config-route-map)#match ip address OSPF-ACL
```

C. Filtering OSPFv2 Routes Using Metric Values

Border gateway protocol routes being redistributed into OSPFv2 can be filtered based on the BGP multi-exit discriminator (MED) metric. When multiple paths into an AS exist, the MED metric conveys to a neighboring AS the preferred path into the AS. Additionally, OSPFv2 routes can be filtered based on the patch cost (metric) of the route. Use the **match metric** <*value*> command to configure a route map to match routes based on a specified metric value. The <*value*> parameter specifies the metric value to match. Valid range is **1** to **4294967295**. Enter the command from the Route Map Configuration mode as follows:

```
(config-route-map)#match metric 100
```

> **NOTE**
>
> *Routes can only be matched based on the Type 1 metric.*

D. Filtering OSPFv2 Routes Using Route Tags

OSPFv2 external (Type 5) LSAs can use route tags in their headers to identify redistributed routes. Routes that have non-zero route tags can be filtered based on these tags. Use the **match tag** <*value*> command to configure a route map to match routes based on a specified route tag. The <*value*> parameter specifies the route tag value to match. Valid range is **1** to **4294967295**. Enter the command from the Route Map Configuration mode as follows:

```
(config-route-map)#match tag 555
```

Step 3: Applying Attributes to Matched Routes

After deciding how to filter routes, you can use the **set** commands to apply attributes to the routes matched by the route map. These commands and their use are described in the following sections.

> **NOTE**
>
> *The **set** commands outlined in the section are not supported by the **distribute-list route-map** command. They may be used to apply attributes to routes redistributed using the **redistribute** command.*

A. Specifying a Metric for Matched Routes

OSPFv2 uses a combination of path type and path cost (metric) to determine the shortest path to a destination. OSPFv2 first examines the path type to determine the shortest path. If the types are the same, OSPFv2 chooses the path with the lowest metric. This preference for a lower metric can be used to influence neighbors to select a certain route. Since OSPF prefers routes with a lower metric, the connection to the neighbor that receives the route with the lowest metric is more likely to be selected.

When metrics are used, routes to a specific part of the network are typically classified according to their destination address. This classification is accomplished using one prefix list or several; depending on the network setup and the goal. Separate route maps are then configured for each neighbor to which the router connects. Refer to *A. Filtering OSPFv2 Routes Using Prefix Lists on page 27* for information on how to create a prefix list and associate it with a route map. A prefix list is associated with each route map entry. Again, depending on the network setup and the goal, the same prefix list can be associated with each route map entry or there can be a different prefix list associated with every route map entry.

Use the **set metric** <*value*> command to specify a metric value for the routes that have been matched by a route map. The <*value*> parameter specifies the metric value. Valid range is **0** to **4294967295**. Enter the command from the Route Map Configuration mode as follows:

```
(config-route-map)#set metric 10
```

B. Specifying a Metric Type for Matched Routes

Routes that are redistributed into OSPFv2 are assigned a metric type by the ASBR. Type 1 external metrics have a path cost that is equal to the sum of the internal cost of the route plus the external cost. Type 2 external metrics use only the cost to the destination and ignore the cost to reach the ASBR. Route maps can be used to specify the metric type (either Type 1 or Type 2) of routes redistributed into OSPFv2.

Use the **set metric-type [type-1 | type-2]** command to specify the metric type for routes that have been matched by a route map. The **type-1** parameter specifies the Type 1 metric. The **type-2** specifies the Type 2 metric. Enter the command from the Route Map Configuration mode as follows:

```
(config-route-map)#set metric-type type-2
```

C. Specifying a Route Tag for Matched Routes

Route tags can be used to implement routing policies using route maps. For example, they can be used to control route redistribution to prevent routing loops when configuring multipoint route redistribution (i.e. routes are redistributed from protocol A into protocol B, then back into protocol A).

Use the **set tag** <*value*> command to specify the route tag for routes that have been matched by a route map. The <*value*> parameter specifies the route tag value. Enter the command from the Route Map Configuration mode as follows:

```
(config-route-map)#set tag 555
```

5. OSPFv2 Configuration Examples

The example scenarios contained within this section are designed to enhance understanding of OSPFv2 configurations on AOS products. The examples describe some of the common real-world applications of OSPFv2. All configurations provided in this section use the command line interface (CLI).

> **NOTE**
>
> *The configuration parameters entered in these examples are sample configurations only. These applications should be configured in a manner consistent with the needs of your particular network. CLI prompts have been removed from the configuration examples to provide a method of copying and pasting configurations directly from this configuration guide into the CLI. These configurations should not be copied without first making the necessary adjustments to ensure they will function properly in your network.*

Example 1: Single Area OSPF with Default Route Propagation and Static Route Redistribution

For networks with fewer than 20 routers or 1000 prefixes, a single area OSPFv2 network design may simplify administration, optimize traffic, direct resource utilization, and provide a quickly converging network that is scalable for future growth. OSPFv2 specifications mandate that if the network contains multiple areas, all areas must connect to area 0 (the transit or backbone area); therefore, for scalabilty, it is best practice to always design an OSPFv2 implementation by configuring area 0 first, even when there is only a single area. The following example illustrates a scenario in which three AOS routers are configured with a single area OSPFv2 design. For simplicity, the OSPF **network area** commands are configured with a 0.0.0.0 wildcard mask (the mask contains wildcard bits where 0 is a match and 1 is a "do not care" bit). The wildcard does not have to match the subnet mask of the network participating in OSPF. This configuration practice ensures that OSPFv2 is activated on the proper interface.

In this example, Router A specifies that an external default route will be injected into the OSPFv2 process with the **default-information originate** command and redistributes two static routes (172.16.0.0/16 and 192.168.123.0/24) with the **redistribute static subnets** command. By default, the router with the highest priority or router ID will be selected as the designated router. In this example, Router A is a centralized router with the highest processing capabilities, so it will be manually configured with the highest priority (using the **ip ospf priority** interface command) to relieve the CPUs of other routers.

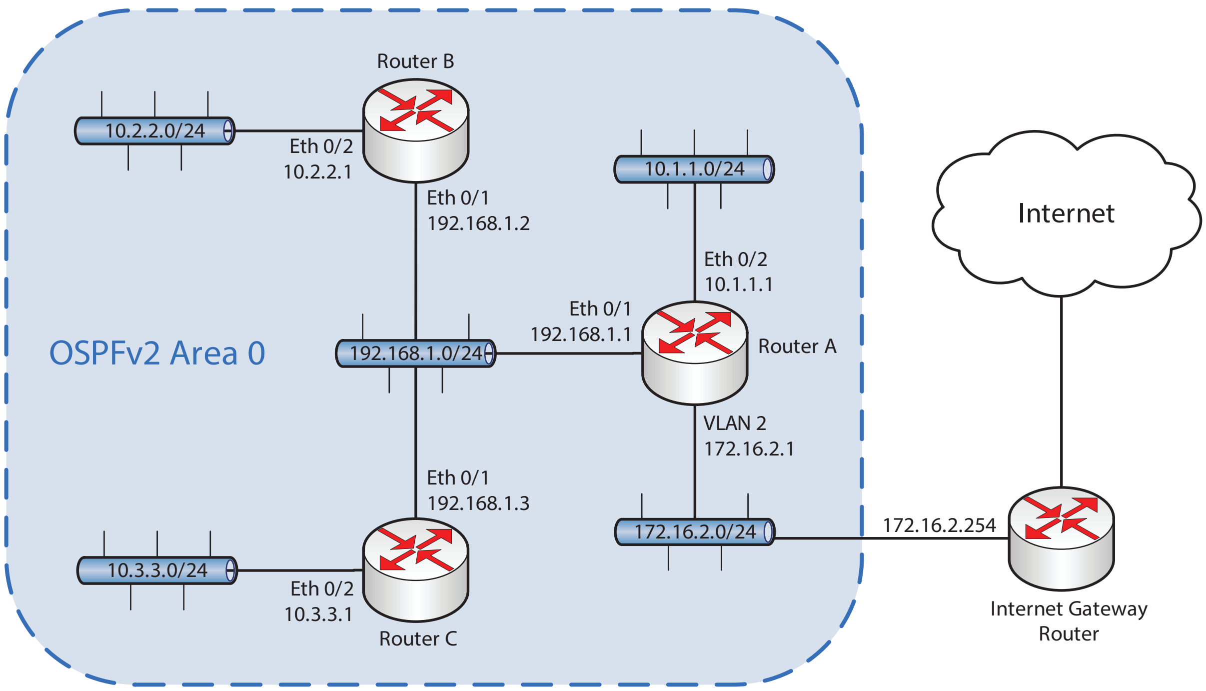

Figure 8. Single Area OSPFv2 Example

The following is the sample configuration for routers **A**, **B**, and **C**:

Router A

```
interface eth 0/1
  ip address 192.168.1.1 255.255.255.0
  ip ospf 1 area 0
  ip ospf 1 priority 200
  no shutdown
!
interface eth 0/2
  ip address 10.1.1.1 255.255.255.0
  ip ospf 1 area 0
  no shutdown
!
interface vlan 2
  ip address 172.16.2.1 255.255.255.0
  ip ospf 1 area 0
  no shutdown
!
router ospf 1
  default-information originate
  redistribute static subnets
```

```
  network 192.168.1.0 0.0.0.255 area 0
  network 10.1.1.0 0.0.0.255 area 0
  network 172.16.2.0 0.0.0.255 area 0
!
ip route 172.16.2.254 0.0.0.0 10.0.20.254
ip route 172.16.2.254 255.255.0.0 10.0.20.254
ip route 172.16.2.254 255.255.255.0 10.0.20.254
```

Router B

```
interface eth 0/1
  ip address 192.168.1.2 255.255.255.0
  ip ospf 1 area 0
  no shutdown
!
interface eth 0/2
  ip address 10.2.2.1 255.255.255.0
  ip ospf 1 area 0
  no shutdown
!
router ospf 1
  network 192.168.1.0 0.0.0.255 area 0
  network 10.2.2.0 0.0.0.255 area 0
```

Router C

```
interface eth 0/1
  ip address 192.168.1.3 255.255.255.0
  ip ospf 1 area 0
  no shutdown
!
interface eth 0/2
  ip address 10.3.3.1 255.255.255.0
  ip ospf 1 area 0
  no shutdown
!
router ospf 1
  network 192.168.1.0 0.0.0.255 area 0
  network 10.3.3.0 0.0.0.255 area 0
```

Example 2: Multi-VRF OSPFv2 CLI Configuration Example

The following configuration example illustrates three OSPFv2 processes running in a multi-VRF router. One OSPFv2 instance is in the default VRF (Blue) while the other two are in non-default VRFs (**RED** and **GREEN**). Each OSPFv2 instance shows one connected neighbor router. This example is provided for illustration purposes only. You will need to make configuration changes to ensure the configuration will function in your network. The network topology for this example is shown in *Figure 9*.

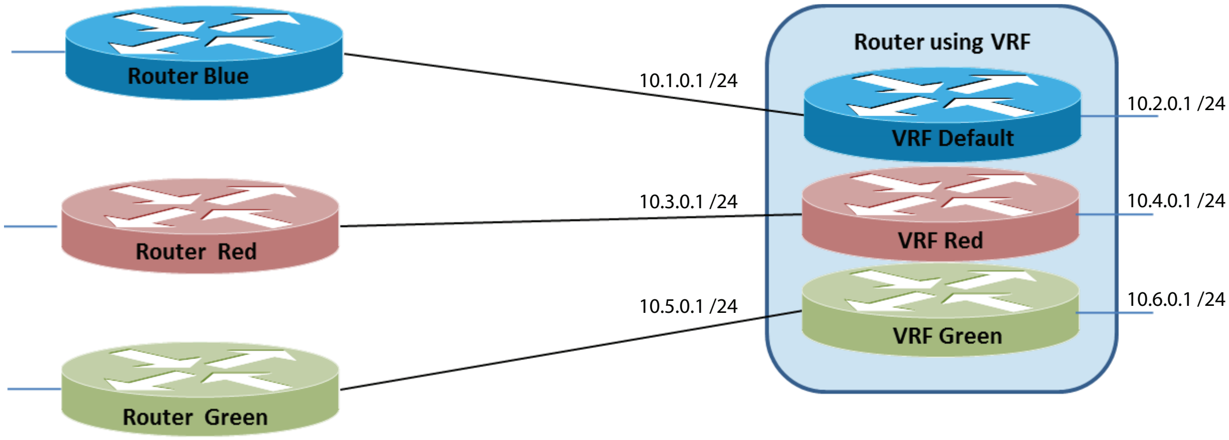

Figure 9. Multi-VRF OSPFv2 Example

The following is the sample configuration for the router using VRF:

```
!
vrf RED route-distinguisher 3:4
!
vrf GREEN route-distinguisher 5:6
!
interface eth 0/1
  no shutdown
!
interface eth 0/1.1
  ip address 10.1.0.1 255.255.255.0
  no shutdown
!
interface eth 0/1.2
  vrf forwarding RED
  ip address 10.3.0.1 255.255.255.0
  no shutdown
!
interface eth 0/1.3
  vrf forwarding GREEN
  ip address 10.5.0.1 255.255.255.0
  no shutdown
!
interface eth 0/2
  no shutdown
!
interface eth 0/2.1
  ip address 10.2.0.1 255.255.255.0
  no shutdown
!
interface eth 0/2.2
  vrf forwarding RED
  ip address 10.4.0.1 255.255.255.0
  no shutdown
```

```
!
interface eth 0/2.3
  vrf forwarding GREEN
  ip address 10.6.0.1 255.255.255.0
  no shutdown
!
router ospf 12
  network 10.2.0.0 0.0.0.255 area 1
  network 10.1.0.0 0.0.0.255 area 0
!
router ospf 34 vrf RED
  network 10.4.0.0 0.0.0.255 area 1
  network 10.3.0.0 0.0.0.255 area 0
!
router ospf 56 vrf GREEN
  network 10.6.0.0 0.0.0.255 area 1
  network 10.5.0.0 0.0.0.255 area 0
!
```

Example 3: Redistributing BGP into OSPFv2

In this example, Router A receives BGP routes from its peer in the cloud (not shown). BGP routes are redistributed into OSPFv2. At redistribution in Router A, a route map (**BGP2OSPF**) is used with a prefix list (**OSPF-LIST**) to assign tags to the BGP routes redistributed into the OSPFv2 process. The routes that match the prefix list are assigned the route tag **10**, whereas those that do not match the prefix-list are assigned the route-tag **20**. Router B will use these tags to add only some of the BGP routes to its route table.

Router B uses a distribute list to specify that only a subset of the tagged routes from Router A are added to the local route table. A route map (**OSPF-IN**) is used to deny all routes with the route tag **20** (the routes that do not match the **OSPF-LIST** prefix list on Router A) and permit all other routes. This route map is then referenced by the **distribute-list route-map in** command to specify that only the permitted routes are added to the local route table.

NOTE

*Filtering routes with the **distribute-list route-map in** command does not affect the exchange of LSAs for those routes.*

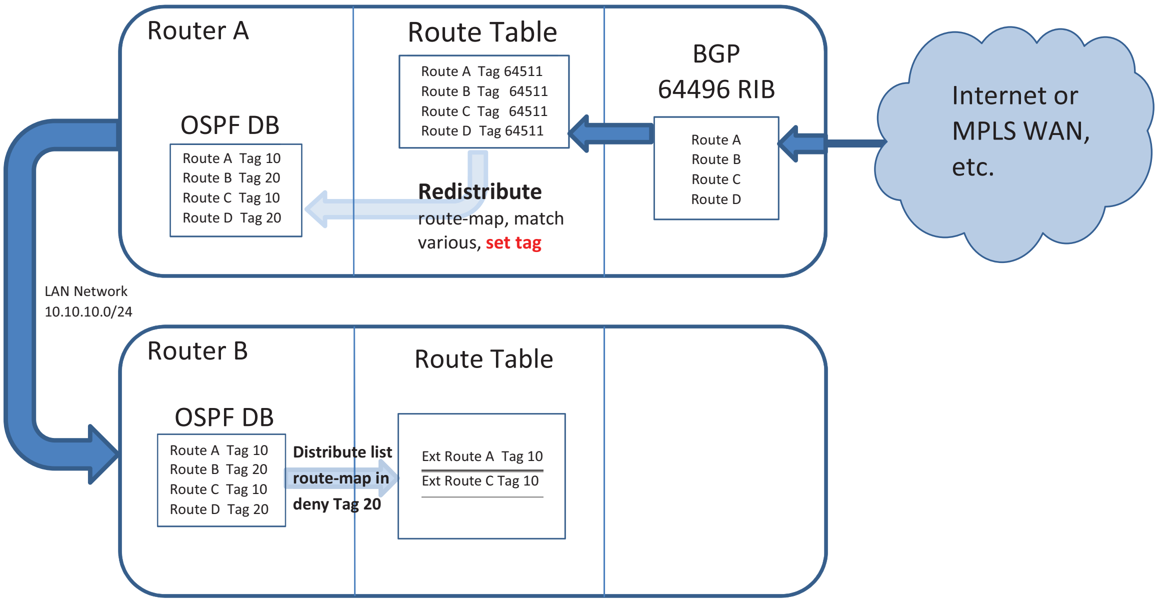

Figure 10. Redistributing BGP into OSPF with Tags for Filtering by other OSPF Routers

Router A

```
router ospf 1
  redistribute bgp subnets route-map BGP-TO-OSPF
  network 10.10.10.0 0.255.255.255 area 0
!
ip prefix-list OSPF-LIST seq 10 permit 1.1.1.1/32
!
route-map BGP-TO-OSPF permit 10
  match ip address prefix-list OSPF-LIST
  set tag 10
route-map BGP-TO-OSPF permit 20
  set tag 20
!
```

Router B

```
router ospf 1
  distribute-list route-map OSPF-IN in
  network 10.10.10.0 0.0.0.255 area 0
!
route-map OSPF-IN deny 10
  match tag 20
route-map OSPF-IN permit 20
```

6. OSPFv2 Command Summary

The following tables summarize the commands used to configure OSPFv2 on an AOS product.

Table 6. OSPFv2 Global Configuration Commands

Prompt	Command	Description
(config)#	**[no] router ospf** *<process id>* **[vrf** *<name>***]**	Enables OSPFv2, creates an OSPFv2 process, and enters the OSPFv2 Configuration mode. OSPFv2 is disabled by default. Valid process ID range is **1** to **65535**.
(config-ospf)#	**[no] area** *<area id | ipv4 address>* **default-cost** *<cost>*	Specifies the cost of the default summary route injected into the specified OSPFv2 area. Area IDs can be an integer, with a range of **1** to **4294967295**, or an IPv4 address. IPv4 addresses are entered in dotted decimal notation (**10.10.10.1**). Valid cost range is **0** to **16777214**. Default cost value is **0**.
(config-ospf)#	**[no] area** *<area id | ipv4 address>* **range** *<ipv4 address> <subnet mask>* **[advertise | not-advertise]**	Controls the route summarization between OSPFv2 areas. The *<area id | ipv4 address>* parameter specifies the ID of the area of which the router is a member. Area IDs can be an integer, with a range of **1** to **4294967295**, or an IPv4 address. IPv4 addresses are entered in dotted decimal notation (**10.10.10.1**). The *<ipv4 address>* parameter is the IPv4 address to be advertised. The *<subnet mask>* parameter specifies the subnet mask that corresponds to a range of IPv4 addresses (network) or a specific host. Subnet masks can be expressed in dotted decimal notation (for example, **255.255.255.0**). The optional advertise and not-advertise parameters specify whether the summary will or will not be advertised to other areas. Summaries are advertised by default.
(config-ospf)#	**[no] area** *<area id | ipv4 address>* **stub [no-summary]**	Specifies the area type as a stub or total stub. By default, area types are normal and are not specified as a stub. Area IDs can be an integer, with a range of **1** to **4294967295**, or an IPv4 address. IPv4 addresses are entered in dotted decimal notation (**10.10.10.1**). The no-summary parameter specifies the area as a total stub.
(config-ospf)#	**[no] auto-cost reference-bandwidth** *<bandwidth>*	Specifies the reference value used to calculate the OSPFv2 cost of an interface. Default value is 100 Mbps. Valid *<bandwidth>* range is **1** to **4294967** Mbps.
(config-ospf)#	**[no] distance [ospf external | ospf inter-area | ospf intra-area]** *<distance>*	Specifies the administrative distance for OSPFv2 routes. The optional **ospf external**, **ospf inter-area**, and **ospf intra-area** parameters allow you to specify the administrative distance for routes in the specified area. The *<distance>* parameter specifies the administrative distance to be used by the route type. Valid distance range is **0** to **255**. By default, all routes have an administrative distance of **110**.
(config-ospf)#	**[no] maximum-paths** *<value>*	Specifies the maximum number of equal cost routes to a given prefix that OSPFv2 can enter into the route table. Valid range is **1** to **6**, with a default value of **4**.

Table 6. OSPFv2 Global Configuration Commands (Continued)

Prompt	Command	Description
(config-ospf)#	**[no] network** *<ipv4 address> <wildcard mask>* **area** *<area id | ipv4 address>*	Enables routing on an IP stack and specifies area IDs for the interfaces on which OSPFv2 will run. The *<ipv4 address>* parameter specifies a valid IPv4 address. Enter IPv4 addresses in dotted decimal notation (for example, **10.10.10.1**). The *<wildcard mask>* parameter specifies the wildcard mask that corresponds to a range of IPv4 addresses. Wildcard masks are expressed in dotted decimal notation (for example, **0.0.255**). The **area** *<area id | ipv4 address>* parameter specifies the ID of the area that you are defining. The area ID can be specified as an integer or an IPv4 address. Valid integer area ID range is **1** to **4294967295**. Enter IPv4 addresses in dotted decimal notation (for example, **10.10.10.1**).
(config-ospf)#	**rfc1583compatibility**	Specifies that the algorithm defined in RFC 1583 is used to determine AS external routes. Use the no form of this command to specify that the algorithm defined in RFC 2328 is used to determine AS external routes. By default, RFC 1583 compatibility is enabled.
(config-ospf)#	**[no] router-id** *<ipv4 address>*	Specifies the value to be used by the OSPFv2 process as the router ID. The *<ipv4 address>* parameter is a 32-bit value represented in an IPv4 address format used by the specific OSPFv2 instance as the router ID. IPv4 addresses should be entered in dotted decimal notation. Valid IPv4 address range is **0.0.0.1** to **255.255.255.255**.

Table 6. OSPFv2 Global Configuration Commands (Continued)

Prompt	Command	Description
(config-ospf)#	**[no] summary-address** *<ipv4 address> <subnet mask>* **[not-advertise] [tag** *<value>***]**	Controls route summarization and router advertisement that are redistributed into this OSPFv2 process. The *<ipv4 address>* specifies the IP address of the route being redistributed into OSPF2 area of which the router is a member, and that contains the prefixes being summarized. Enter IPv4 addresses in dotted decimal notation (for example, **10.10.10.1**). The *<subnet mask>* parameter specifies the subnet mask that corresponds to a range of IPv4 addresses (network) or a specific host. Subnet masks can be expressed in dotted decimal notation (for example, **255.255.255.0**).The optional not-advertise parameter specifies that the summary will not be advertised to other OSPFv2 areas. The tag parameter specifies the route tag for the summarized route. Route tags can be used to implement routing policies using route maps. Route tags only apply to External (Type 5) LSAs. Range is **1** to **4294967295**.
(config-ospf)#	**[no] timers spf** *<delay>* **[***<hold time>***]**	Specifies the OSPFv2 SPF calculation and hold interval timers. The *<delay>* parameter is the time between receipt of OSPFv2 topology changes and the beginning of SPF calculations. Valid range is **0** to **65535** seconds with a default value of **5** seconds. The optional *<hold time>* parameter specifies the time between consecutive SPF calculations. Valid range is **0** to **65535** seconds with a default value of **10** seconds.

Table 7. OSPFv2 Redistribution Configuration Commands

Prompt	Command	Description
(config-ospf)#	**[no] default-information originate [always] [metric** *<value>*] **[metric-type** *<type>*] **[tag** *<value>*]	Specifies that a default external route is injected into this OSPFv2 process. The optional always parameter allows the ASBR to inject a default route, even if the ASBR lacks a default route. The optional **metric** *<value>* parameter assigns an OSPFv2 metric value to the route being injected. A metric of **10** is used if the **default-metric** command has not been used. Valid metric range is **0** to **16777214**. The optional **metric-type** parameter specifies an external metric type for the route being injected. Metric type **1** specifies that routes are assigned a metric beginning with the metric value specified by this command and the cost of the OSPFv2 path is added as they are advertised throughout the autonomous system. Metric type **2** is not affected by the OSPFv2 path cost and retains the original metric value. By default, metric type **2** is used. The **tag** parameter specifies the route tag for the default external route being injected. Route tags can be used to implement routing policies using route maps. Route tags only apply to External (Type 5) LSAs. Range is **1** to **4294967295**.
(config-ospf)#	**[no] default-metric** *<value>*	Specifies the metric value used for redistributed routes when the value is not otherwise specified. The *<value>* parameter specifies the OSPFv2 metric value assigned to the route being injected into OSPFv2. Valid range is **0** to **6777214**. By default, the metric is set to **20**.
(config-ospf)#	**[no] redistribute [bgp | connected | ospf** *<process id>* **[include-connected] | rip | static] [metric** *<value>*] **[metric-type** *<type>*] **[route-map** *<name>*] **[tag** *<value>*] **subnets**	Redistributes routes from a specified source into the OSPFv2 process. The **bgp**, **connected**, **ospf**, **rip**, and **static** parameters define the source from which routes are redistributed. If routes are redistributed from an OSPFv2 process, the optional **include-connected** parameter allows the redistribution of corresponding connected routes. The optional **metric** parameter assigns an OSPFv2 metric value to the route. By default, the metric is **10**. Valid metric range is **0** to **16777214**. The optional metric-type parameter specifies the external metric type for the route. The optional **route-map** parameter specifies a route map to be applied to routes being redistributed by this command. The **tag** parameter specifies the route tag for the redistributed routes. Route tags can be used to implement routing policies using route maps. Route tags only apply to External (Type 5) LSAs. Range is **1** to **4294967295**.The optional subnets parameter specifies subnet redistribution when redistributing routes into OSPFv2.

Table 7. OSPFv2 Redistribution Configuration Commands (Continued)

Prompt	Command	Description
(config-ospf)#	**distribute-list** *<ipv4 acl name>* **out**	Applies an ACL to routes that are redistributed into the OSPFv2 network. This command allows you to filter what traffic gets redistributed into the OSPFv2 network. For example, if the OSPFv2 process is configured to redistribute RIP routes, and RIP has a route to 10.22.8.0/24, OSPFv2 would normally generate a Type 5 external LSA to distribute the traffic through the AS. However, if an ACL is applied that blocks the 10.22.8.0/24 prefix, the external LSA will not be generated and the traffic will not be distributed through the AS. The *<ipv4 acl name>* parameter specifies an IPv4 ACL name. This is a standard IPv4 ACL against which the redistributed routes are matched. By default, no distribute list filters exist for the OSPFv2 routing process
(config-ospf)#	**distribute-list route-map** *<name>* **in**	Applies a route map to filter the routes added to the OSPFv2 route table. This commands allows you to filter what OSPFv2 is allowed to add to the route table. For example, if OSPF calculations determine that a route to 10.22.8.0/24 should be added to the route table, the route map matching rules are applied to the prefix to determine whether the route should be added to the route table. This command does not affect the distribution of LSAs or the LS database (every router in the same area should still have the same database for that area). The *<name>* parameter specifies a route map name. By default, no distribute list filters exist for the OSPFv2 routing process

Table 8. OSPFv2 Interface Configuration Commands

Prompt	Command	Description
(config-eth 0/1)#	**[no] ip ospf** *<process id>* **area** *<area id | ipv4 address>*	Adds an interface to an OSPFv2 process. The *<process id>* parameter is the OSPFv2 routing process to which this interface belongs. Valid range is **1** to **65535**. By default, a process ID is not defined. The **area** *<area id | ipv4 address>* parameter specifies the ID of the area that you are defining. The area ID can be specified as an integer or an IPv4 address. Valid integer area ID range is **1** to **4294967295**. Enter IPv4 addresses in dotted decimal notation (for example, **10.10.10.1**).
(config-eth 0/1)#	**[no] ip ospf** *<process id>* **authentication [message-digest | null]**	Specifies the type of OSPFv2 authentication used on an interface. By default, no authentication is used. The *<process id>* parameter specifies the OSPFv2 routing process to which this interface belongs. Valid process ID range is **1** to **65535**. The optional **message-digest** parameter specifies the message-digest authentication type. The **null** parameter specifies that no OSPFv2 authentication is used on the interface.

Table 8. OSPFv2 Interface Configuration Commands (Continued)

Prompt	Command	Description
(config-eth 0/1)#	**[no] ip ospf** *<process id>* **authentication-key** *<password>*	Specifies a simple-text authentication password to be used by other routers using the OSPFv2 simple-text password authentication. By default, no authentication is used. The *<process id>* parameter specifies the OSPFv2 routing process to which this interface belongs. Valid process ID range is **1** to **65535**. The **authentication-key** *<password>* parameter specifies the simple-text password.
(config-eth 0/1)#	**[no] ip ospf** *<process id>* **cost** *<cost>*	Specifies the OSPFv2 cost of the interface. By default, the cost of the interface is automatically computed. The *<process id>* parameter specifies the OSPFv2 process to which this interface belongs. Valid process ID range is **1** to **65535**. The *<cost>* parameter is the OSPFv2 cost of the interface. This value overrides any computed cost value. Valid range is **1** to **65535**.
(config-eth 0/1)#	**[no] ip ospf** *<process id>* **dead-interval** *<value>*	Specifies the maximum interval allowed between OSPFv2 Hello packets on this interface. This value must be the same across all interfaces on a link. The *<process id>* parameter specifies the OSPFv2 process to which this interface belongs. Valid process ID range is **1** to **65535**. The *<value>* parameter is the maximum number of seconds allowed between Hello packets. It is recommended that this value be **4** times the Hello packet interval value. Valid range is **1** to **65535** seconds, with a default value of **40** seconds.
(config-eth 0/1)#	**[no] ip ospf** *<process id>* **hello-interval** *<value>*	Specifies the interval between OSPFv2 Hello packets sent on this interface. The *<process id>* parameter specifies the OSPFv2 process to which this interface belongs. Valid process ID range is **1** to **65535**. The *<value>* parameter is the number of seconds allowed between OSPFv2 Hello packets. Valid range is **1** to **65535**, with a default value of **10** seconds.
(config-eth 0/1)#	**[no] ip ospf** *<process id>* **message-digest-key [1 | 2] md5** *<key>*	Specify the OSPFv2 message digest 5 (MD5) authentication (16 byte maximum) keys. By default, no authentication is used. The *<process id>* parameter specifies the OSPFv2 routing process to which this interface belongs. Valid process ID range is **1** to **65535**. The **1** and **2** parameters specifies the key ID number. The **md5** *<key>* parameter specifies the OSPFv2 MD5 authentication key.
(config-eth 0/1)#	**[no] ip ospf** *<process id>* **network [broadcast | point-to-point]**	Specifies the OSPFv2 network type for the interface. The *<process id>* parameter specifies the OSPFv2 process to which this interface belongs. Valid process ID range is **1** to **65535**. The **broadcast** parameter sets the OSPFv2 network type to broadcast, and the **point-to-point** parameter sets the network type as point-to-point. By default, Ethernet interfaces are set to **broadcast**, and PPP, Frame Relay, GRE, and loopback interfaces are set to **point-to-point**.

Table 8. OSPFv2 Interface Configuration Commands (Continued)

Prompt	Command	Description
(config-eth 0/1)#	**[no] ip ospf** *<process id>* **priority** *<value>*	Specifies the interface OSPFv2 priority. The *<process id>* parameter specifies the OSPFv2 process to which this interface belongs. Valid process ID range is **1** to **65535**. The *<value>* parameter specifies the OSPFv2 priority. Valid range is **0** to **255**, with a default value of **1**. A value of **0** indicates the router is ineligible to become either the DR or BDR.
(config-eth 0/1)#	**[no] ip ospf** *<process id>* **retransmit-interval** *<value>*	Specifies the interval between OSPFv2 LSAs sent on the interface. The *<process id>* parameter specifies the OSPFv2 process to which this interface belongs. Valid process ID range is **1** to **65535**. The *<value>* parameter specifies the number of seconds between LSAs. Valid range is **1** to **65535** seconds, with a default value of **5** seconds.
(config-eth 0/1)#	**[no] ip ospf** *<process id>* **transmit-delay** *<value>*	Specifies the estimated time required to propagate an LSA on the interface. The *<process id>* parameter specifies the OSPFv2 process to which this interface belongs. Valid process ID range is **1** to **65535**. The *<value>* parameter specifies the number of seconds required to send LSAs from the interface. Valid range is **1** to **65535**, with a default value of **1** second.
(config-eth 0/1)#	**[no] ip ospf** *<process id>* **shutdown**	Disables an OSPFv2 process on the interface. The *<process id>* parameter specifies the OSPFv2 process to shut down. Valid process ID range is **1** to **65535**.

Table 9. OSPFv2 Route Map Configuration Commands

Prompt	Command	Description
(config)#	**route-map** *<name>* **[deny | permit]** *<number>*	Creates a route map and enters the Route Map Configuration mode. The *<name>* parameter specifies a name for the route map. The **deny** parameter specifies not to process routes matching the specified route map attributes. The **permit** parameter specifies to processes routes matching the specified route map attributes. The *<number>* parameter specifies a sequence number for this route entry. Range is **1** to **4294967295**.
(config)#	**ip prefix-list** *<name>* **seq** *<number>* **[deny | permit]** *<network ip/length>* **[ge | le]** *<value> <value>*	Specifies a prefix to be matched or a range of mask lengths. The *<name>* parameter specifies the name of the list. Up to **80** characters are allowed in a name. The **seq** *<number>* parameter specifies the entry's unique sequence number that determines the processing order. Lower numbered entries are processed first. Range is **1** to **4294967294**. The deny *<network ip/length>* parameter denies access to entries matching the specified network IPv4 address and the corresponding network prefix length (for example, **10.10.10.0 /24**).The **permit** *<network ip/length>* parameter permits access to entries matching the specified network IPv4 address and the corresponding network prefix length (for example, **10.10.10.0 /24**). The **ge** *<value>* parameter specifies the lower end of the range. Range is **0** to **32**. The **le** <value> parameter specifies the upper end of the range. Range is **0** to **32**.
(config)#	**match ip address prefix-list** *<name>*	Configures a route map to route traffic based on the defined IPv4 prefix list route filter. The *<name>* parameter specifies the name of the prefix list.
(config)#	**match ip address** *<ipv4 acl name>*	Configures a route map to match traffic based on an IPv4 ACL name defined with the **ip access-list** command. The *<ipv4 acl name>* parameter specifies the name of the IPv4 ACL to match.
(config)#	**match metric** *<value>*	Configures a route map to match routes based on a specified metric value. The *<value>* parameter specifies the metric value to match. Valid range is **1** to **4294967295**.
(config)#	**match tag** *<value>*	Configures a route map to match routes based on a specified route tag. The *<value>* parameter specifies the route tag value to match. Valid range is **1** to **4294967295**.
(config-route-map)#	**set metric** *<value>*	Specifies a a metric value for the routes that have been matched by a route map. The *<value>* parameter specifies the metric value. Valid range is **0** to **4294967295**.

Table 9. OSPFv2 Route Map Configuration Commands (Continued)

Prompt	Command	Description
(config-route-map)#	**set metric-type [type-1 | type-2]**	Specifies the metric type for routes that have been matched by a route map. The **type-1** parameter specifies the Type 1 metric. The **type-2** specifies the Type 2 metric.
(config-route-map)#	**set tag** *<value>*	Specifies the route tag for routes that have been matched by a route map. The *<value>* parameter specifies the route tag value. Range is **1** to **4294967295**.

7. Troubleshooting OSPFv2

Various commands can be used from the Enable mode to view OSPFv2 configurations, clear certain OSPFv2 settings, and create debug messages for OSPFv2 events. These commands are included in the following sections.

OSPFv2 Show Commands

Show commands can be used to view current OSPFv2 configurations, including OSPFv2 processes, link state database information, and router interfaces configured for OSPFv2. The following section describes the **show** commands available for OSPFv2 troubleshooting.

Use the **show ip ospf [**<*process id*>**]** command to display general information regarding OSPFv2 processes. You can limit the output of this command to a single OSPFv2 process by entering the process ID. The following is sample output from this command:

```
>enable
#show ip ospf 61
Summary of OSPF Process with ID: 9.8.8.1, VRF RED
  Supports only single Type Of Service routes (TOS 0)
  SPF delay timer: 5 seconds, Hold time between SPFs: 10 seconds
  LSA interval: 240 seconds
  Number of external LSAs: 2, Checksum Sum: 0x1508f
  Number of areas: 2, normal: 2, stub: 0, NSSA: 0
  Area (0) 9.8.8.1
     Number of interfaces in this area: 1
     Authentication type: 0
     SPF algorithm execution count: 4
     Number of LSAs: 5, Checksum Sum: 0x271bc
  Area (1) 9.8.8.1
     Number of interfaces in this area: 2
     Authentication type: 0
     SPF algorithm execution count: 7
     Number of LSAs: 4, Checksum Sum: 0x28037
```

Use the **show ip ospf [**<*process id*> **[**<*area id*>**]] database [adv-router** <*router id*>**] [self-originate]** command to display information contained in the OSPFv2 link state database. You can limit the output of this command to a specified OSPFv2 process by entering the process ID, a specified OSPFv2 area by entering the area ID, a specified advertising router by entering the router ID, or to LSAs originated by the router by entering the self-originate parameter. The following is sample output from this command:

```
>enable
#show ip ospf 61 database
```

```
OSPF router with ID: 9.8.8.1 (Process ID 61, VRF RED)

Router Link States, Area 0
  Adv Router        Age   Seq #       Checksum    Link count     Bits
  9.8.8.1           1705  0x80000093  0x065E         1           None
  192.168.120.200   895   0x80000087  0x6271         1           B

Net Link States, Area 0
  Adv Router        Age    Seq #      Checksum    Rtr Count
  192.168.120.2     895    0x80000081 0x0643      2

Summary Net Link States, Area 0
  Adv Router        Age   Seq #       Checksum    Prefix/Link ID
  9.8.8.1           31    0x80000091  0x0F99      9.8.8.1
  192.168.120.200   895   0x80000081  0xDE9C      192.168.120.0

Router Link States, Area 1
  Adv Router        Age   Seq #       Checksum    Link count     Bits
  9.8.8.1           1769  0x80000095  0xE95A      2              None

Summary Net Link States, Area 1
  Adv Router        Age   Seq #       Checksum    Prefix/Link ID
  9.8.8.1           31    0x80000091  0x4ED6      10.24.106.0
  9.8.8.1           600   0x80000090  0xB11E      192.168.120.0
  9.8.8.1           600   0x80000090  0x4015      200.200.200.2

Type 5 AS External Net Link States
  Adv Router        Age   Seq #       Checksum    Prefix/Link ID
  9.8.8.1           670   0x80000091  0x69CA      0.0.0.0
  9.8.8.1           30    0x80000091  0xBBDA      10.24.204.128
```

Use the **show ip ospf [**<*process id*> **[**<*area id*>**]] database asbr-summary [**<*link state id*>**] [adv-router** <*router id*>**] [self-originate] [internal]** command to display information about the autonomous system boundary router (ASBR) summary LSAs. You can limit the output of this command to a specified OSPFv2 process by entering a process ID, to an OSPFv2 area by entering the area ID, to a specified LSA by entering the link state ID, to a specified advertising router by entering a router ID, or to LSAs originated by the router by entering the **self-originate** parameter. The optional **internal** parameter displays SPF calculation results for this LSA, and whether this LSA was used in route calculation. The following is sample output from this command:

```
>enable
#show ip ospf 61 database asbr-summary

OSPF router with ID: 35.35.35.35 (Process ID 61)

Summary ASB Link States, Area 0
Link State age: 153
Link State type: Summary-ASB-LSA (0x0004)
Link State ID: 2.2.2.2
Advertising Router: 92.92.92.92
Sequence Number: 0x800003CC
Checksum: 0xA170
Options:
Length: 28
Metric: 2
```

Use the **show ip ospf [**<*process id*> **[**<*area id*>**]] database database-summary** command to display a summary of information about the OSPFv2 link state database. You can limit the output of this command to a specified OSPFv2 process by entering the process ID, or to a specified OSPFv2 area by entering the area ID. The following is sample output from this command:

```
>enable
#show ip ospf 61 database database-summary
OSPF router with ID: 9.8.8.1 (Process ID 61, VRF RED)
Area 0 database summary
  LSA Type          Count
  Router            2
  Network           1
  Summary-Network   2
  Summary-ASBR      0
  Opaque            0
  Subtota           5
  External          2
  AS Opaque         0

Area 0 database summary
  LSA Type          Count
  Router            1
  Network           0
  Summary-Network   3
  Summary-ASBR      0
  Opaque            0
  Subtotal          4
  External          2
  AS Opaque         0
```

Use the **show ip ospf [**<*process id*> **[**<*area id*>**]] database external [**<*link state id*>**] [adv-router** <*router id*> **| self-originate]** to provide details from the OSPFv2 link state database of external LSAs. You can limit the output of this command to a specified OSPFv2 process by entering a process ID, to an OSPFv2 area by entering the area ID, to a specified LSA by entering the link state ID, to a specified advertising router by entering a router ID, or to LSAs originated by the router by entering the **self-originate** parameter. The following is sample output from this command:

```
>enable
#show ip ospf 61 database external
OSPF router with ID: 9.8.8.1 (Process ID 61, VRF RED)
  Type 5 AS External Net Link States
     Link State age: 1626
     Link State type: AS External (0x0005)
     Link State ID: 0.0.0.0
     Advertising Router: 9.8.8.1
     Sequence Number: 0x80000093
     Checksum: 0x65CC
     Options: E
     Length: 36
     Network Mask: 255.255.255.255
        Metric Type: 1 (Comparable directly to link state metric)
        Metric: 22222
        Forward Address: 10.24.106.2
        External Route Tag: 0x00000000
```

Use the **show ip ospf [**<*process id*> **[**<*area id*>**]] database network [**<*link state id*>**] [adv-router <***router id*> **| self-originate]** command to provide details from the OSPFv2 link state database of network LSAs. You can limit the output of this command to a specified OSPFv2 process by entering a process ID, to an OSPFv2 area by entering the area ID, to a specified LSA by entering the link state ID, to a specified advertising router by entering a router ID, or to LSAs originated by the router by entering the **self-originate** parameter. The following is sample output from this command:

```
>enable
#show ip ospf 61 database network
OSPF router with ID: 9.8.8.1 (Process ID 61, VRF RED)
  Network Link States, Area 0
     Link State age: 60
     Link State type: Network-LSA (0x0002)
     Link State ID: 10.24.106.10
     Advertising Router: 9.8.8.1
     Sequence Number: 0x80000002
     Checksum: 0x3282
     Options: E
     Length: 32
        Network Mask: 255.255.255.0
        Number of Attached Routers: 2
        Attached Router: 9.8.8.1
        Attached Router: 4.4.4.4
```

Use the **show ip ospf [**<*process id*> **[**<*area id*>**]] database router [**<*link state id*>**] [adv-router <***router id*> **| self-originate] [internal]** command to provide details from the OSPFv2 link state database of router LSAs. You can limit the output of this command to a specified OSPFv2 process by entering a process ID, to an OSPFv2 area by entering the area ID, to a specified LSA by entering the link state ID, to a specified advertising router by entering a router ID, or to LSAs originated by the router by entering the **self-originate** parameter. The optional **internal** parameter displays SPF calculation results for this LSA, and whether this LSA was used in route calculation. The following is sample output from this command:

```
>enable
#show ip ospf 61 database router
OSPF router with ID: 9.8.8.1 (Process ID 61, VRF RED)
  Router Link States, Area 0
  Link State age: 350
  Link State type: Router-LSA (0x0001)
  Link State ID: 9.8.8.1
  Advertising Router: 9.8.8.1
  Sequence Number: 0x8000009D
  Checksum: 0x4210
  Options: E
  Length: 40
     Flags: Area Border Router, AS Boundary Router
     Number of Links: 1
        Link connected to: Transit network Link
           (Link ID} Designated Router address: 5.5.5.5
           (Link Data) Router Interface address: 9.8.8.1
        TOS 0 Metric: 1
```

Use the **show ip ospf [**<*process id*> **[**<*area id*>**]] database summary [adv-router <***router id*> **| self-originate]** command to display information contained in the OSPFv2 link state database. You can limit the output of this command to a specified OSPFv2 process by entering a process ID, to an OSPFv2 area by entering the area ID, to a specified advertising router by entering a router ID, or to LSAs originated by the router by entering the **self-originate** parameter. The following is sample output from this command:

```
>enable
#show ip ospf 61 database summary
OSPF router with ID: 9.8.8.1 (Process ID 61, VRF RED)
  Summary Net Link States, Area 1
  Link State age: 689
  Link State type: Summary Network (0x0003)
  Link State ID: 10.24.106.0
  Advertising Router: 9.8.8.1
  Sequence Number: 0x80000009
  Checksum: 0x5F4E
  Options: E
  Length: 28
  Network Mask: /24
  Metric: 1
```

Use the **show ip ospf [**<*process id*> **[**<*area id*>**]] interface [**<*interface*>**]** command to display OSPFv2 information related to router interfaces. You can limit the output of this command to a specified OSPFv2 process by entering a process ID, to an OSPFv2 area by entering the area ID, to a specified interface by entering the interface. The following is sample output from this command:

```
>enable
#show ip ospf 61 interface
eth 0/1.106 is UP, line protocol is UP
  IP address: 10.24.106.10 255.255.255.0, Area: 0
  Process ID 61, Router ID: 9.8.8.1, Network type: Broadcast, Cost: 1
  Transmit delay: 1, State: DR, Priority: 1
  Designated Router (ID): 9.8.8.1, Interface Address: 10.24.106.10
  Backup Designated Router (ID): 200.200.200.2, Interface Address:
   10.24.106.2
  Timer intervals: Hello: 10, Dead: 40, Retransmit: 5
     Hello due in: 00:00:08
  Number of neighbors: 1, Adjacent neighbors: 1
     Adjacent with neighbor: 200.200.200.2 (Backup Designated Router)
loop 97 is UP, line protocol is UP
  IP address: 9.8.7.1 255.255.255.0, Area: 1
  Process ID 61, Router ID: 9.8.8.1, Network type: Point-to-point, Cost: 1
  Transmit delay: 1, State: PTPT, Priority: 1
  Timer intervals: Hello: 10, Dead: 40, Retransmit: 5
     Hello due in: 00:00:10
  Number of neighbors: 0, Adjacent neighbors: 0
loop 98 is UP, line protocol is UP
  IP address: 9.8.8.1 255.255.255.0, Area: 1
  Process ID 61, Router ID: 9.8.8.1, Network type: Point-to-point, Cost: 1
  Transmit delay: 1, State: PTPT, Priority: 1
  Timer intervals: Hello: 10, Dead: 40, Retransmit: 5
     Hello due in: 00:00:10
  Number of neighbors: 0, Adjacent neighbors: 0
```

Use the **show ip ospf [**<*process id*> **[**<*area id*>**]] neighbor [**<*interface*>**] [**<*neighbor id*>**] [detail]** command to display OSPFv2 information related to OSPFv2 neighbors. You can limit the output of this command to a specified OSPFv2 process by entering a process ID, to an OSPFv2 area by entering the area ID, to a specified neighbor by entering the connecting interface, or to a neighbor having the specific neighbor ID. The optional **detail** parameter specifies that more detailed information is displayed. The following is sample output from this command:

```
>enable
#show ip ospf 61 neighbor
OSPF router with ID: 9.8.8.1, Process ID 61, VRF RED

Neighbor ID      Pri    State     Dead Time    Interface ID    Interface
204.204.204.61   1      FULL/BDR 00:00:38     20              eth 0/1.106
```

Use the **show ip ospf [<*process id*>] summary-address** command to provide details about redistributed routes that have been summarized using the **summary-address** configuration command (refer to Step 7 on *Step on page 23*). You can limit the output of this command to a specified OSPFv2 process by entering a process ID. The following is sample output of from this command:

```
>enable
#show ip ospf 61 summary-address
OSPF Summary Addresses, Process ID 61, VRF RED:
  8.7.0.0/255.255.0.0 Metric 11111, Type 1, advertise
```

OSPFv2 Clear Commands

Clear commands are used to reset and restart specific OSPFv2 processes and to refresh routes distributed into OSPFv2 processes.

Use the **clear ip ospf [<*process id*>] process** command to reset and restart all OSPFv2 processes. To only reset or restart one OSPFv2 process, enter the process ID. For example, to restart OSPFv2 process **5**, enter the command from the Enable mode as follows:

```
>enable
#clear ip ospf 5 process
```

Use the **clear ip ospf [<*process id*>] redistribution** command to refresh routes distributed into the specified OSPFv2 process(es). To only refresh routes distributed to a single OSPFv2 process, enter the process ID. For example, to refresh routes distributed into OSPFv2 process **5**, enter the command from the Enable mode as follows:

```
>enable
#clear ip ospf 5 redistribution
```

OSPFv2 Debug Commands

Debug commands are used to enable debug messages for the OSPFv2 feature. Use the **debug ip ospf [<*process id*>] [adj [errors] | database-timer | events | flood [errors] | hello [errors] | lsa-generation | packet [errors] | packet [rx [summary] | tx [summary]] | retransmission | spf [route-calculation]]** command to enable OSPFv2 debug messages. You can limit debug messages to a specific OSPFv2 process by entering a process ID. In addition, you can specify that OSPFv2 adjacency events are displayed (**adj**), that OSPFv2 database timer information is displayed (**database-timer**), that OSPFv2 events are displayed (**events**), that OSPFv2 flooding information is displayed (**flood**), that OSPFv2 Hello events are displayed (**hello**), that OSPFv2 LSA generation information is displayed (**lsa-generation**), that OSPFv2 packet details in either directions (**rx** and **tx**) are displayed, that OSPFv2 errors with received packets are displayed (**packet errors**), that OSPFv2 retransmission events are displayed (**retransmission**), or that OSPFv2 SPF events (**spf**) and route calculations are displayed (**spf route-calculation**). In addition, you can specify that errors about certain information are displayed using the **errors** parameters, and that summary information about packet details is displayed using the **summary** parameter (this option limits packet information to a single line). For example, to display debug event messaging for all of OSPFv2 on the unit, enter the command from the Enable mode as follows:

```
>enable
#debug ip ospf
Receiving OSPF packet from 192.168.120.200 to 9.8.8.1 on eth 0/1.6
  SysUpTime=1222577915 ms.
  Hello packet from Router ID: 1.1.1.4; Ver:2 Length:48
  Area ID: 0.0.0.0 Checksum: 0x8659; Using Null Authentication: 0:0
  PrefixLenV4: /64; Hello Interval: 10 Options: 0x13 Router Priority: 1 Router
   Dead Interval: 40
  Designated Router: 10.24.106.4 Backup Designated Router: 10.24.106.5
  1 Neighbors:
     123.1.1.1
  16:35:24: OSPF: HELLO received form 1.1.1.4, neighbor state is FULL
```

8. Warranty and Contact Information

Warranty and contact information for all ADTRAN products can be obtained using the information in the following sections.

Warranty

Warranty information can be found online by visiting www.adtran.com/warranty.

Contact Information

To contact ADTRAN, choose one of the following methods:

Department	Contact Information	
Customer Care	From within the U.S.: From outside the U.S.:	(888) 4ADTRAN ((888)-423-8726) +1 (256) 963-8716
Technical Support	Support Community Product Support:	https://supportcommunity.adtran.com www.adtran.com/support
Training	Email: ADTRAN University:	training@adtran.com www.adtran.com/training
Sales	For pricing and availability:	1 (800) 827-0807