Cisco 1000 Series Connected Grid Routers Unicast Routing Software Configuration Guide

First Published: September 2012

Last Updated: June 10, 2013

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-25630-04

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

No combinations are authorized or intended under this document.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco 1000 Series Connected Grid Routers Unicast Routing Software Configuration Guide

CONTENTS

CHAPTER 1

Unicast Routing

This chapter introduces the underlying concepts for Layer 3 unicast routing protocols in Cisco 1000 Series Connected Grid Routers (*hereafter* referred to as the Cisco CG-OS router) and WAN backhaul redundancy.The system software for the router is identified as the Cisco CG-OS software.

This chapter includes the following sections:

- Information About Layer 3 Unicast Routing, page 1-1
- Routing Algorithms, page 1-7
- Summary of Layer 3 Unicast Routing Features, page 1-9
- WAN Backhaul Redundancy, page 1-10

Information About Layer 3 Unicast Routing

Layer 3 unicast routing involves determining optimal routing paths. You can use routing algorithms to calculate the optimal path from the router to a destination. This calculation depends on the algorithm selected, route metrics, and other considerations such as load balancing and alternate path discovery.

This section includes the following topics:

- Routing Fundamentals, page 1-2
- Routing Metrics, page 1-3
- Router IDs, page 1-5
- Autonomous Systems, page 1-5
- Convergence, page 1-6
- Load Balancing and Equal Cost Multipath, page 1-6
- Administrative Distance, page 1-6
- Stub Routing, page 1-6

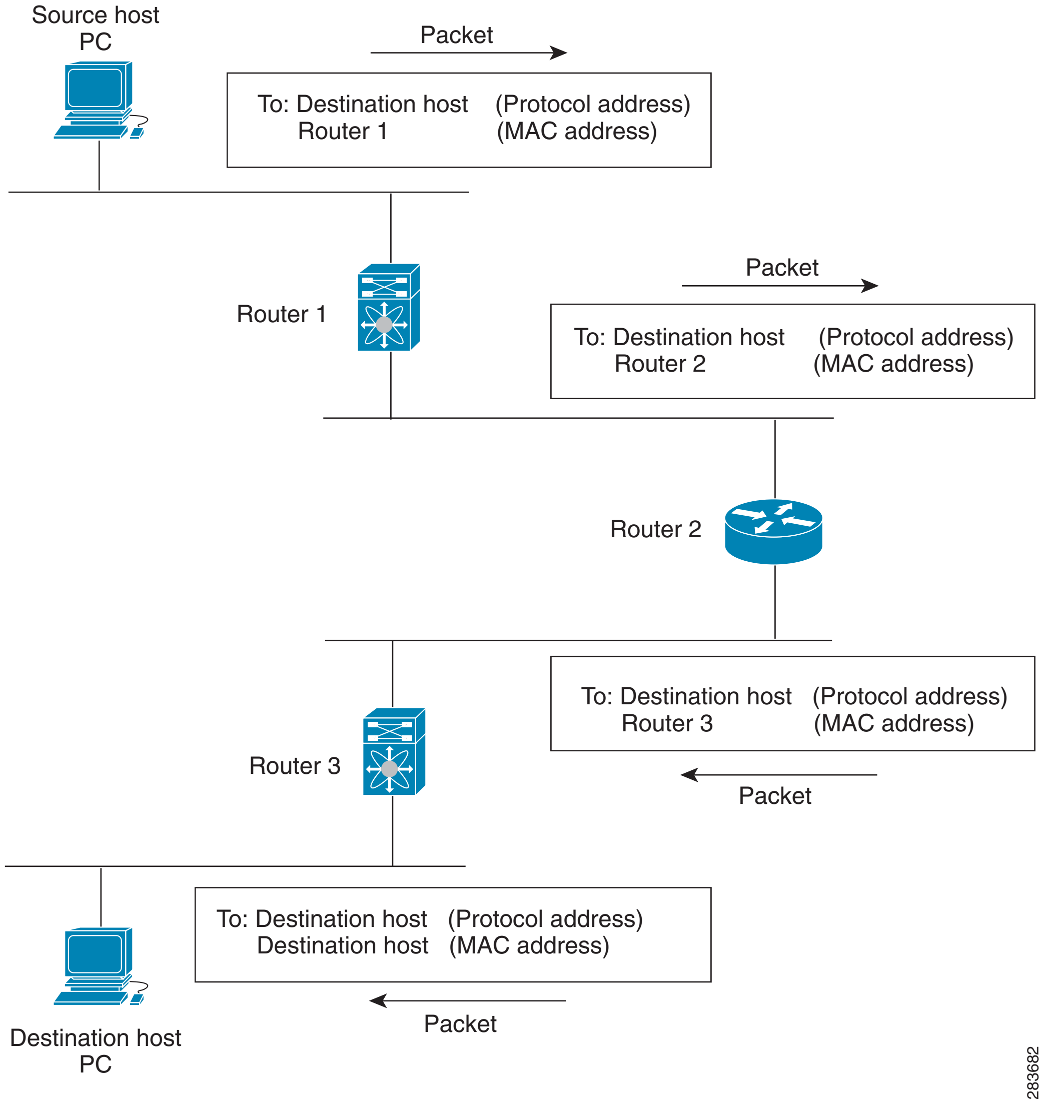
Routing Fundamentals

Routing protocols use a metric to evaluate the best path to the destination. A metric is a standard of measurement, such as a path bandwidth, that routing algorithms use to determine the optimal path to a destination. To aid path determination, routing algorithms initialize and maintain routing tables that contain route information such as the IP destination address, the address of the next router, or the next hop. Destination and next-hop associations tell a router that an IP destination can be reached optimally by sending the packet to a particular router that represents the next hop on the way to the final destination. When a router receives an incoming packet, it checks the destination address and attempts to associate this address with the next hop.

Routing tables can contain other information, such as the data about the desirability of a path. Routers compare metrics to determine optimal routes, and these metrics differ depending on the design of the routing algorithm used. See the Routing Metrics, page 1-3.

Routers communicate with one another and maintain their routing tables by transmitting a variety of messages. The routing update message is one such message that consists of all or a portion of a routing table. By analyzing routing updates from all other routers, a router can build a detailed picture of the network topology. A link-state advertisement, which is another example of a message sent between routers, informs other routers of the link state of the sending router. You can also use link information to enable routers to determine optimal routes to network destinations. For more information, see the Routing Algorithms, page 1-7.

Figure 1-1 Packet Header Updates Through a Network


Source host
PC
Packet
To: Destination host (Protocol address)
Router 1 (MAC address)
Router 1
Packet
To: Destination host (Protocol address)
Router 2 (MAC address)
Router 2
To: Destination host (Protocol address)
Router 3 (MAC address)
Router 3
Packet
To: Destination host (Protocol address)
Destination host (MAC address)
Packet
Destination host
PC
283682

Routing Metrics

Routing algorithms use many different metrics to determine the best route. Sophisticated routing algorithms can base route selection on multiple metrics.

This section includes the following metrics:

- Path Length, page 1-4
- Reliability, page 1-4
- Routing Delay, page 1-4
- Bandwidth, page 1-4
- Load, page 1-4
- Communication Cost, page 1-4

Path Length

The path length is the most common routing metric. Some routing protocols allow you to assign arbitrary costs to each network link. In this case, the path length is the sum of the costs associated with each link traversed. Other routing protocols define the hop count, which is a metric that specifies the number of passes through internetworking products, such as routers, that a packet must take from a source to a destination.

Reliability

The reliability, in the context of routing algorithms, is the dependability (in terms of the bit-error rate) of each network link. Some network links might go down more often than others. After a network fails, certain network links might be repaired more easily or more quickly than other links. The reliability factors that you can take into account when assigning the reliability rating are arbitrary numeric values that you usually assign to network links.

Routing Delay

The routing delay is the length of time required to move a packet from a source to a destination through the internetwork. The delay depends on many factors, including the bandwidth of intermediate network links, the port queues at each router along the way, the network congestion on all intermediate network links, and the physical distance that the packet must travel. Because the routing delay is a combination of several important variables, it is a common and useful metric.

Bandwidth

The bandwidth is the available traffic capacity of a link. Although the bandwidth is the maximum attainable throughput on a link, routes through links with greater bandwidth do not necessarily provide better routes than routes through slower links. For example, if a faster link is busier, the actual time required to send a packet to the destination could be greater.

Load

The load is the degree to which a network resource, such as a router, is busy. You can calculate the load in a variety of ways, including CPU usage and packets processed per second. Monitoring these parameters on a continual basis can be resource intensive.

Communication Cost

The communication cost is a measure of the operating cost to route over a link. The communication cost is another important metric, especially if you do not care about performance as much as operating expenditures. For example, the line delay for a private line might be longer than a public line, but you can send packets over your private line rather than through the public lines that cost money for usage time.

Router IDs

Each routing process has an associated router ID. You can configure the router ID to any interface. The Cisco CG-OS router supports cellular, Ethernet (Fast Ethernet and Gigabit Ethernet), and WiMax interfaces. When you do not configure the router ID, the Cisco CG-OS router selects the router ID based on the following criteria:

- The Cisco CG-OS router prefers loopback0 over any other interface. When loopback0 does not exist, then the router prefers the first loopback interface over any other interface type.
- When you do not configure a loopback interface, the Cisco CG-OS router uses the first interface in the configuration file as the router ID. When you configure any loopback interface after the Cisco CG-OS software selects the router ID, the loopback interface becomes the router ID. When the loopback interface is not loopback0 and you configure loopback0 with an IP address, the router ID changes to the IP address of loopback0.
- When the interface that the router ID is based on changes, that new IP address becomes the router ID. When any other interface changes its IP address, there is no router ID change.

Related Topics

Cisco 1000 Series Connected Grid Routers WiFi Software Configuration Guide

Cisco 1240 Connected Grid Router Hardware Installation Guide

Cisco Connected Grid Cellular 3G Module for CGR1000 Series Installation and Configuration Guide

Cisco Connected Grid WiMAX Module for CGR1000 Series Installation and Configuration Guide

Autonomous Systems

An autonomous system (AS) is a network controlled by a single technical administration entity. Autonomous systems divide global external networks into individual routing domains, where local routing policies are applied. This organization simplifies routing domain administration and simplifies consistent policy configuration.

Each autonomous system can support multiple interior routing protocols that dynamically exchange routing information through route redistribution. The Regional Internet Registries assign a unique number to each public autonomous system that directly connects to the Internet. This autonomous system number (AS number) identifies both the routing process and the autonomous system.

The Cisco CG-OS router supports 4-byte AS numbers. Table 1-1 lists the AS number ranges.

Table 1-1 AS Numbers

2-Byte Numbers	4-Byte Numbers in AS.dot Notation	4-Byte Numbers in plaintext Notation	Purpose
1 to 64511	0.1 to 0.64511	1 to 64511	Public AS (assigned by RIR)[1]
64512 to 65534	0.64512 to 0.65534	64512 to 65534	Private AS (assigned by local administrator)
65535	0.65535	65535	Reserved
N/A	1.0 to 65535.65535	65536 to 4294967295	Public AS (assigned by RIR)

[1] RIR=Regional Internet Registries

Private autonomous system numbers are used for internal routing domains but must be translated by the Cisco CG-OS router for traffic that is routed out to the Internet. It is important not to configure routing protocols to advertise private autonomous system numbers to external networks. By default, the Cisco CG-OS router does not remove private autonomous system numbers from routing updates.

Note The autonomous system number assignment for public and private networks is governed by the Internet Assigned Number Authority (IANA). For information about autonomous system numbers, including the reserved number assignment, or to apply to register an autonomous system number, see this URL: http://www.iana.org/

Convergence

A key aspect to measure for any routing algorithm is how much time a router takes to react to network topology changes. When a part of the network changes for any reason, such as a link failure, the routing information in different routers might not match. Some routers will have updated information about the changed topology, while other routers will still have the old information. The convergence is the amount of time before all routers in the network have updated, matching routing information. The convergence time varies depending on the routing algorithm. Fast convergence minimizes the chance of lost packets caused by inaccurate routing information.

Load Balancing and Equal Cost Multipath

Routing protocols can use load balancing or equal cost multipath (ECMP) to share traffic across multiple paths.When a router learns multiple routes to a specific network, it installs the route with the lowest administrative distance in the routing table. If the router receives and installs multiple paths with the same administrative distance and cost to a destination, load balancing can occur. Load balancing distributes the traffic across all the paths, sharing the load. The number of paths used is limited by the number of entries that the routing protocol puts in the routing table.

Administrative Distance

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

Stub Routing

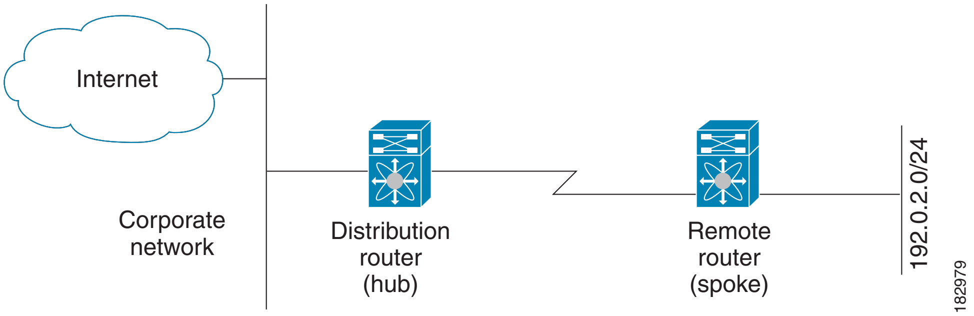
You can use stub routing in a hub-and-spoke network topology, where one or more end (stub) networks are connected to a remote router (the spoke) that is connected to one or more distribution routers (the hub). The remote router is adjacent only to one or more distribution routers. The only route for IP traffic to follow into the remote router is through a distribution router. This type of configuration is commonly used in WAN topologies in which the distribution router is directly connected to a WAN. The distribution router can be connected to many more remote routers. Often, the distribution router is connected to 100 or more remote routers. In a hub-and-spoke topology, the remote router must forward all nonlocal traffic to a distribution router, so it becomes unnecessary for the remote router to hold a complete routing table. Generally, the distribution router sends only a default route to the remote router.

Only specified routes are propagated from the remote (stub) router. The stub router responds to all queries for summaries, connected routes, redistributed static routes, external routes, and internal routes with the message “inaccessible.” A router that is configured as a stub sends a special peer information packet to all neighboring routers to report its status as a stub router.

Any neighbor that receives a packet that informs it of the stub status does not query the stub router for any routes, and a router that has a stub peer does not query that peer. The stub router depends on the distribution router to send the proper updates to all peers.

Figure 1-2 shows a simple hub-and-spoke configuration.

Figure 1-2 Simple Hub-and-Spoke Network



Stub routing does not prevent routes from being advertised to the remote router. Figure 1-2 shows that the remote router can access the corporate network and the Internet through the distribution router only. A full route table on the remote router, in this example, serves no functional purpose because the path to the corporate network and the Internet is always through the distribution router. A larger route table reduces only the amount of memory required by the remote router. The bandwidth and memory used can be lessened by summarizing and filtering routes in the distribution router. In this network topology, the remote router does not need to receive routes that have been learned from other networks because the remote router must send all nonlocal traffic, regardless of its destination, to the distribution router. To configure a true stub network, you should configure the distribution router to send only a default route to the remote router.

The Open Shortest Path First (OSPF) protocol supports stub areas. For more information on OSPF, see OSPF, page 1-9.

Routing Algorithms

Routing algorithms determine how a router gathers and reports reachability information, how it deals with topology changes, and how it determines the optimal route to a destination. Various types of routing algorithms exist, and each algorithm has a different impact on network and router resources. Routing algorithms use a variety of metrics that affect calculation of optimal routes. You can classify routing algorithms by type, such as static or dynamic, and interior or exterior.

This section includes the following topics:

- Static Routes and Dynamic Routing Protocols, page 1-8
- Interior and Exterior Gateway Protocols, page 1-8
- Link-State Protocols, page 1-8

Static Routes and Dynamic Routing Protocols

Static routes are route table entries that you manually configure. These static routes do not change unless you reconfigure them. Static routes are simple to design and work well in environments where network traffic is relatively predictable and where network design is relatively simple.

Because static routing systems cannot react to network changes, do not use them for large, constantly changing networks. Most routing protocols today use dynamic routing algorithms that adjust to changing network circumstances by analyzing incoming routing update messages. When the message indicates that a network change has occurred, the routing software recalculates routes and sends out new routing update messages. These messages permeate the network, triggering routers to rerun their algorithms and change their routing tables accordingly.

You can supplement dynamic routing algorithms with static routes where appropriate. For example, you can configure each subnetwork with a static route to the IP default gateway or router of last resort (the router to which all unrouteable packets are sent).

Interior and Exterior Gateway Protocols

You can separate networks into unique routing domains or autonomous systems. An autonomous system is a portion of an internetwork under common administrative authority that is regulated by a particular set of administrative guidelines. Routing protocols that route between autonomous systems are called exterior gateway protocols or interdomain protocols. Routing protocols used within an autonomous system are called interior gateway protocols or intradomain protocols. OSPF is an example of an interior gateway protocol that the Cisco CG-OS router supports.

Note The Cisco CG-OS router does not support any exterior gateway protocols.

Link-State Protocols

The link-state protocols, also known as shortest path first (SPF), share information with neighboring routers. Each router builds a link-state advertisement (LSA) that contains information about each link and directly connected neighbor router.

Each LSA has a sequence number. When a router receives an LSA and updates its link-state database, the LSA floods all adjacent neighbors. When a router receives two LSAs with the same sequence number (from the same router), the router does not flood its neighbors with the last LSA received because it wants to prevent an LSA update loop.

Discovering neighbors and establishing adjacency is an important part of a link state protocol. Neighbors use special Hello packets to discover on another. Hello packets also serve as keepalive notifications for each neighbor router. Adjacency establishes a common set of operating parameters for the link-state protocol between neighbor routers.

When a router receives an LSA, the router adds the LSA to its link-state database. Each entry consists of the following parameters:

- Router ID (for the router that originated the LSA)
- Neighbor ID
- Link cost
- Sequence number of the LSA

- Age of the LSA entry

The router runs the SPF algorithm on the link-state database, building the shortest path tree for that router. This SPF tree is used to populate the routing table.

In link-state algorithms, each router builds a picture of the entire network in its routing tables. The link-state algorithms send small updates everywhere, while distance vector algorithms send larger updates only to neighboring routers.

Because link-state algorithms converge more quickly, they are less likely to cause routing loops than distance vector algorithms. However, link-state algorithms require more CPU power and memory than distance vector algorithms and they can be more expensive to implement and support. Link-state protocols are generally more scalable than distance vector protocols.

OSPF is an example of a link-state protocol.

Summary of Layer 3 Unicast Routing Features

This section provides a brief introduction to the Layer 3 unicast features and protocols supported in the Cisco CG-OS router.

This section includes the following topics:

- IPv4 and IPv6, page 1-9
- IP Services, page 1-9
- OSPF, page 1-9
- Static Routing, page 1-10

IPv4 and IPv6

Layer 3 routing employs the IPv4 and/or the IPv6 protocol. IPv6 increases the number of network address bits to 128 bits from the 32 bits employed by IPv4.

The Cisco CG-OS router supports Address Resolution Protocol (ARP) and Internet Control Message Protocol (ICMP) when employing IPv4.

The Cisco CG-OS router supports Internet Control Message Protocol for IPv6 (ICMPv6) and Neighbor Discovery (ND) when employing IPv6.

For more information, see Chapter 2, “Configuring IPv4” and Chapter 3, “Configuring IPv6.”

IP Services

IP Services addresses Domain Name System (DNS) clients. For more information, see Chapter 4, “Configuring IP Services.”

OSPF

The Open Shortest Path First (OSPF) protocol is a link-state routing protocol that exchanges network reachability information within an autonomous system (AS). Each OSPF router advertises information about its active links to its neighbor routers. Link information consists of the link type, the link metric, and the neighbor router that is connected to the link. The advertisements that contain this link information are called link-state advertisements (LSAs). The Cisco CG-OS router supports both OSPFv2 for IPv4 networks and OSPFv3 for IPv6 networks.

For more information, see Chapter 5, "Configuring OSPFv2" and Chapter 6, "Configuring OSPFv3."

Static Routing

Static routing allows you to configure a fixed route to a destination. This feature is useful for small networks where the topology is simple. Static routing is also used with other routing protocols to control default routes. For more information, see Chapter 8, "Configuring Static Routing."

WAN Backhaul Redundancy

Redundant WAN backhauls can be configured on the Cisco CG-OS router within an Open Shortest Path First version 2 (OSPFv2) area by assigning link costs to cellular (3G) and WiMax interfaces. The interface with the lower assigned link cost remains the primary link until that link goes down; and, then traffic automatically goes to the secondary link with the next lowest cost. In cases where the link with the higher cost fails, no redirect of traffic occurs because the Cisco CG-OS router by default routes all traffic to the link with the lowest cost.

For more information, see Chapter 7, "Configuring WAN Backhaul Redundancy."

CHAPTER 2

Configuring IPv4

This chapter describes how to configure Internet Protocol version 4 (IPv4), which includes addressing, Address Resolution Protocol (ARP), Reverse ARP, and Internet Control Message Protocol (ICMP), on Cisco 1000 Series Connected Grid Routers (*hereafter* referred to as the Cisco CG-OS router). The system software for the router is identified as the Cisco CG-OS software.

This chapter includes the following sections:

Information About IPv4

You can configure IP on the Cisco CG-OS router to assign IP addresses to network interfaces. When you assign IP addresses, you enable the interfaces and allow communication with the hosts on those interfaces. The Cisco CG-OS router supports the following interfaces: cellular (3G), WiMax, and Ethernet (Fast Ethernet and Gigabit Ethernet).

You can configure an IP address as primary or secondary on a Cisco CG-OS router. An interface can have one primary IP address and multiple secondary addresses. All networking devices on an interface must share the same primary IP address because the packets that are generated by the Cisco CG-OS router always use the primary IPv4 address. Each IPv4 packet is based on the information from a source or destination IP address. For more information, see the Multiple IPv4 Addresses, page 2-2.

You can use a subnet to mask the IP addresses. A mask determines to which subnet an IP address belongs. An IP address contains the network address and the host address. A mask identifies the bits that denote the network number in an IP address. When you use the mask to subnet a network, the mask is then referred to as a subnet mask. Subnet masks are 32-bit values that allow the recipient of IP packets to distinguish the network ID portion of the IP address from the host ID portion of the IP address.

The IP feature is responsible for handling IPv4 packets that terminate on the Cisco CG-OS router, as well as forwarding of IPv4 packets, which includes IPv4 unicast and multicast route lookup, reverse path forwarding (RPF) checks, and software access control list and policy-based routing (ACL/PBR) forwarding. The IP feature also manages the network interface IP address configuration, duplicate address checks, static routes, and packet send and receive interface for IP clients.

This section includes the following topics:

Multiple IPv4 Addresses

The Cisco CG-OS router supports multiple IP addresses per interface. You can specify an unlimited number of secondary addresses for a variety of situations. The most common are as follows:

- When there are not enough host IP addresses for a particular network interface. For example, if your subnetting allows up to 254 hosts per logical subnet, but on one physical subnet you must have 300 host addresses, then you can use secondary IP addresses on the Cisco CG-OS router or access servers to allow you to have two logical subnets that use one physical subnet.
- Two subnets of a single network might otherwise be separated by another network. You can create a single network from subnets that are physically separated by another network by using a secondary address. In these instances, the first network is extended, or layered on top of the second network. A subnet cannot appear on more than one active interface of the Cisco CG-OS router at a time.

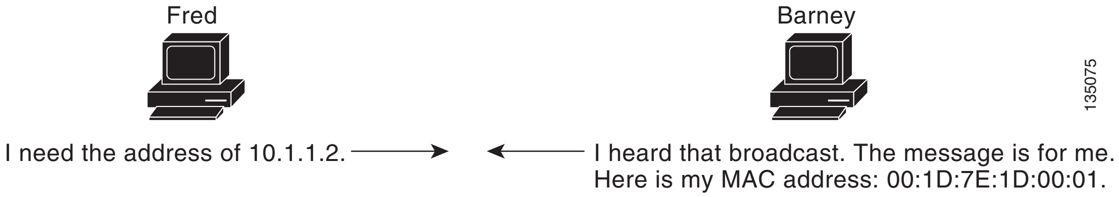
Note If any device on a network segment uses a secondary IPv4 address, then all other devices on that same network interface must also use a secondary address from the same network or subnet. The inconsistent use of secondary addresses on a network segment can quickly cause routing loops.

Address Resolution Protocol

Networking devices and Layer 3 switches and routers use Address Resolution Protocol (ARP) to map IP (network layer) addresses to (Media Access Control [MAC]-layer) addresses which enables IP packets to be sent across networks. Before a device sends a packet to another device, it looks in its own ARP cache to see if there is a MAC address and corresponding IP address for the destination device. If there is no entry, then the source device sends a broadcast message to every device on the network.

Each device compares the IP address to its own. Only the device with the matching IP address replies to the device that sends the data with a packet that contains the MAC address for the device. The source device adds the destination device MAC address to its ARP table for future reference, creates a data-link header and trailer that encapsulates the packet, and proceeds to transfer the data. Figure 2-1 shows the ARP broadcast and response process.

Figure 2-1 ARP Process


Fred
Barney
135075
I need the address of 10.1.1.2.
I heard that broadcast. The message is for me.
Here is my MAC address: 00:1D:7E:1D:00:01.

When the destination device lies on a remote network that is beyond another device, the process is the same except that the device that sends the data sends an ARP request for the MAC address of the default gateway. After the address is resolved and the default gateway receives the packet, the default gateway broadcasts the destination IP address over the networks connected to it. The device on the destination device network uses ARP to obtain the MAC address of the destination device and delivers the packet. ARP is enabled by default.

ARP Caching

ARP caching allows the Cisco CG-OS router to store information from previous ARP translations (IP to MAC address mappings for devices). The mapping of IP addresses to MAC addresses occurs at each hop (device) on the network for every packet sent over an internetwork.

By caching the network addresses and the associated data-link addresses in the memory for a period of time, the Cisco CG-OS router eliminates the need to reexamine each packet when it is broadcast. Instead, the Cisco CG-OS router can reference the ARP cache first and then apply the appropriate address to a packet that is destined for a known destination device. This process helps limit possible negative affects on network performance.

Static and Dynamic Entries in the ARP Cache

Static routing requires that you manually configure the IP addresses, subnet masks, gateways, and corresponding MAC addresses for each interface of each device. Static routing requires more work to maintain the route table. You must update the table each time you add or change routes.

Dynamic routing uses protocols that enable the devices in a network to exchange routing table information with each other. Dynamic routing is more efficient than static routing because the route table automatically updates unless you add a time limit to the cache. Although the default time limit is 25 minutes, you can modify the time limit when the network has a large number of additions and deletions of routes from the cache.

Devices That Do Not Use ARP

When a network is divided into two segments, a bridge joins the segments and filters traffic to each segment based on MAC addresses. The bridge builds its own address table, which uses MAC addresses only. A device has an ARP cache that contains both IP addresses and the corresponding MAC addresses.

Passive hubs are central-connection devices that physically connect other devices in a network. Hubs send messages out on all their ports to the devices and operate at Layer 1 but do not maintain an address table.

Layer 2 switches determine which port connects to the destination device for the message and forwards that message only to that port. However, Layer 3 switches build an ARP cache.

Proxy ARP

Proxy ARP enables a device that is physically located on one network to appear to be logically part of a different physical network that connects to the same device or firewall. Proxy ARP allows you to hide a device with a public IP address on a private network behind a router and still have the device appear to be on the public network in front of the router. By hiding its identity, the router accepts responsibility for routing packets to the real destination. Proxy ARP can help devices on a subnet reach remote subnets without configuring routing or a default gateway.

When devices are not in the same data link layer network but in the same IP network, they try to transmit data to each other as if they are on the local network. However, the router that separates the devices does not send a broadcast message because routers do not pass hardware-layer broadcasts and it cannot resolve the addresses.

When you enable Proxy ARP on the device and it receives an ARP request, it identifies the request as a request for a system that is not on the local LAN. The device responds as if it is the remote destination to which the broadcast is addressed by sending an ARP response that associates the MAC address of the device with the IP address of the remote destination. The local device believes that it is directly connected to the destination device, although their local device is forwarding packets from its local subnetwork to the destination subnetwork. By default, Proxy ARP is disabled.

Local Proxy ARP

You can use local Proxy ARP to enable a device to respond to ARP requests for IP addresses within a subnet in which routing is not generally required. When you enable local Proxy ARP, ARP responds to all ARP requests for IP addresses within the subnet and forwards all traffic between hosts in the subnet. Use this feature only on subnets where hosts do not communicate directly with one another by design. The devices that the hosts connect to are configured to prevent this direct communication.

Gratuitous ARP

Gratuitous ARP sends a request with an identical source IP address and a destination IP address to detect duplicate IP addresses.

Path MTU Discovery

Path maximum transmission unit (MTU) discovery is a method for maximizing the use of available bandwidth in the network between the endpoints of a TCP connection. It is described in RFC 1191. Existing connections are not affected when this feature is turned on or off.

ICMP

You can use the Internet Control Message Protocol (ICMP) to provide message packets that report errors and other information that is relevant to IP processing. ICMP generates error messages, such as ICMP destination unreachable messages, ICMP Echo Requests (which send a packet on a round trip between two hosts), and Echo Reply messages. ICMP also provides many diagnostic functions and can send and redirect error packets to the host. By default, ICMP is enabled on the Cisco CG-OS router.

Some of the ICMP message types are as follows:

- Network error messages
- Network congestion messages
- Troubleshooting information
- Timeout announcements

Note ICMP redirects are disabled on interfaces on which the local proxy ARP feature is enabled.

Prerequisites for IPv4

IPv4 can only be configured on Layer 3 interfaces. The Cisco CG-OS router supports the following Layer 3 interfaces: cellular (3G), WiMax, and Ethernet (Fast Ethernet and Gigabit Ethernet).

Guidelines and Limitations for IPv4

You can configure a secondary IP address only after you configure the primary IP address.

Default Settings

Table 2-1 lists the default settings for IP parameters.

Table 2-1 Default IP Parameters

Parameters	Default
ARP timeout	1500 seconds
proxy ARP	Disabled

Configuring IPv4

This section includes the following topics:

Configuring IPv4 Addressing

You can assign a primary IP address for a network interface.

BEFORE YOU BEGIN

Be aware of the IPv4 addressing plan employed in the network.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**interface ethernet** *slot/port*	Enters interface configuration mode.
Step 3	**ip address** *ip address mask* **[secondary]**	Specifies a primary or secondary IPv4 address for an interface. • The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means the corresponding address bit belongs to the network address. • The network mask can be indicated as a slash (/) and a number (a prefix length). The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash must precede the decimal value and there is no space between the IP address and the slash.
Step 4	**show ip interface**	(Optional) Displays interfaces configured for IPv4.
Step 5	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to assign an IPv4 address to an interface.

```
router# configure terminal
router(config)# interface ethernet 2/3
router(config-if)# ip address 192.168.1.1 255.0.0.0
router(config-if)# copy running-config startup-config
```

Configuring Multiple IP Addresses

BEFORE YOU BEGIN

Configure the primary IP address.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**interface ethernet** *slot/port*	Enters interface configuration mode.
Step 3	**ip address** *ip address mask* **[secondary]**	Specifies the configured address as a secondary IPv4 address.
Step 4	**show ip interface**	(Optional) Displays interfaces configured for IPv4.
Step 5	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to assign multiple, secondary IPv4 addresses to an interface.

```
router# configure terminal
router(config)# interface ethernet 2/3
router(config-if)# ip address 192.168.1.1 255.0.0.0 secondary
router(config-if)# copy running-config startup-config
```

Configuring a Static ARP Entry

You can configure a static ARP entry on the device to map IP addresses to MAC hardware addresses.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**interface ethernet** *slot/port*	Enters interface configuration mode.

	Command	Purpose
Step 3	**ip arp** *ip-address mac-address*	Associates an IP address with a MAC address as a static entry.
Step 4	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to configure a static ARP entry on an interface.

```
router# configure terminal
router(config)# interface ethernet 2/3
router(config-if)# ip arp 192.168.1.1 0019.076c.1a78
router(config-if)# copy running-config startup-config
```

Configuring Proxy ARP

You can configure Proxy ARP on the Cisco CG-OS router to determine the media addresses of hosts on other networks or subnets.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**interface ethernet** *slot/port*	Enters interface configuration mode.
Step 3	**ip proxy-arp**	Enables Proxy ARP on the interface.
Step 4	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to configure Proxy ARP on the Cisco CG-OS router.

```
router# configure terminal
router(config)# interface ethernet 2/3
router(config-if)# ip proxy-arp
router(config-if)# copy running-config startup-config
```

Configuring Local Proxy ARP

You can configure local proxy ARP on a Cisco CG-OS router interface.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**interface ethernet** *slot/port*	Enters interface configuration mode.
Step 3	**ip local-proxy-arp**	Enables Local Proxy ARP on the interface.
Step 4	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to configure local proxy ARP on an Ethernet interface.

```
router# configure terminal
router(config)# interface ethernet 2/3
router(config-if)# ip local-proxy-arp
router(config-if)# copy running-config startup-config
```

Configuring Gratuitous ARP

Although enabled by default on the Cisco CG-OS router, you can modify the request and update parameters for gratuitous ARP on the interface.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**interface ethernet** *slot/port*	Enters interface configuration mode.
Step 3	**ip arp gratuitous** {**request** | **update**}	Configures gratuitous ARP parameters on the interface. Gratuitous ARP is enabled by default. **request**—Enables sending of gratuitous ARP requests when the Cisco CG-OS router detects a duplicate address. **update**—Enables ARP cache updates for gratuitous ARP.
Step 4	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to enable sending of gratuitous ARP requests when the Cisco CG-OS router detects duplicate addresses:

```
router# configure terminal
router(config)# interface ethernet 2/3
router(config-if)# ip arp gratuitous request
router(config-if)# copy running-config startup-config
```

To disable gratuitous ARP requests or updates, use the **no ip arp gratuitous** command.

Configuring Path MTU Discovery

You can configure path MTU discovery to determine the maximum transmit unit (MTU) that you can transmit within the network without requiring fragmentation.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**ip tcp path-mtu-discovery**	Enables path MTU discovery.
Step 3	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to configure path MTU discovery on the Cisco CG-OS router.

```
router# configure terminal
router(config)# ip tcp path-mtu-discovery
router(config)# copy running-config startup-config
```

Configuring IP Directed Broadcasts

An IP directed broadcast is an IP packet whose destination address is a valid broadcast address for some IP subnet but which originates from a device that is not itself part of that destination subnet.

A device that is not directly connected to its destination subnet forwards an IP directed broadcast in the same way that it forwards unicast IP packets destined to a host on that subnet.

When you enable directed broadcast on an interface, the Cisco CG-OS router broadcasts those incoming IP packets identified as directed broadcast to the subnet on which that interface is attached. Then, the destination address in the IP header of the packet is rewritten to the configured IP broadcast address for the subnet, and the packet is sent as a link-layer broadcast.

To enable IP directed broadcasts, use the appropriate command in the interface configuration mode.

:

Command	Purpose
ip directed-broadcast	Enables the translation of a directed broadcast to physical broadcasts for an interface. To disable directed broadcasts, enter the **no ip directed-broadcast** command.

Verifying the IPv4 Configuration

To display IPv4 configuration information, enter any or all of the following commands:

Note The Cisco CG-OS router does not support the optional parameter, [**vrf** *vrf-name*], when present in any of the **show** commands listed below.

Command	Purpose
show ip adjacency	Displays the adjacency table.
show ip adjacency summary	Displays the summary of number of throttle adjacencies.
show ip arp	Displays the ARP table.
show ip arp summary	Displays the summary of the number of throttle adjacencies.
show ip adjacency throttle statistics	Displays only the throttled adjacencies.
show ip interface	Displays IP-related interface information.
show ip arp statistics	Displays the ARP statistics.

Configuration Example for IPv4

This example shows how to configure an IPv4 address:

```
configure terminal
 interface e 2/1
  ip address 192.2.1.1/16
```

CHAPTER 3

Configuring IPv6

This chapter describes how to configure Internet Protocol version 6 (IPv6), which includes addressing, Neighbor Discovery (ND) Protocol, and Internet Control Message Protocol version 6 (ICMPv6), on the Cisco 1000 Series Connected Grid Routers (*hereafter* referred to as the Cisco CG-OS router).
The system software for the router is identified as the Cisco CG-OS software.

This chapter includes the following sections:

- Information About IPv6, page 3-1
- Prerequisites for IPv6, page 3-17
- Guidelines and Limitations for IPv6, page 3-18
- Default Settings, page 3-18
- Configuring IPv6, page 3-18
- Verifying the IPv6 Configuration, page 3-22
- Configuration Example for IPv6, page 3-22

Information About IPv6

IPv6, which is designed to replace IPv4, increases the number of network address bits from 32 bits (in IPv4) to 128 bits. IPv6 is based on IPv4 but it includes a much larger address space and other improvements such as a simplified main header and extension headers.

The larger IPv6 address space allows networks to scale and provide global reachability; and, the simplified IPv6 packet header format handles packets more efficiently. The flexibility of the IPv6 address space reduces the need for private addresses and the use of Network Address Translation (NAT), which translates private (not globally unique) addresses into a limited number of public addresses. IPv6 enables new application protocols that do not require special processing by border routers at the edge of networks.

IPv6 functionality, such as prefix aggregation, simplified network renumbering, and IPv6 site multihoming capabilities, enable more efficient routing. IPv6 provides support for Open Shortest Path First version 3 (OSPFv3). For configuration details on OSPFv3, refer to Chapter 6, “Configuring OSPFv3.”

This section includes the following topics:

- IPv6 Address Formats, page 3-2
- IPv6 Unicast Addresses, page 3-3
- IPv6 Anycast Addresses, page 3-6

- IPv6 Multicast Addresses, page 3-7
- IPv4 Packet Header, page 3-8
- Simplified IPv6 Packet Header, page 3-8
- DNS for IPv6, page 3-11
- Path MTU Discovery for IPv6, page 3-12
- CDP IPv6 Address Support, page 3-12
- ICMP for IPv6, page 3-12
- IPv6 Neighbor Discovery, page 3-13
- IPv6 Neighbor Solicitation Message, page 3-13
- IPv6 Router Advertisement Message, page 3-15
- IPv6 Neighbor Redirect Message, page 3-16

IPv6 Address Formats

An IPv6 address has 128 bits or 16 bytes. The address is divided into eight, 16-bit hexadecimal blocks separated by colons (:) in the format: x:x:x:x:x:x:x:x. Two examples of IPv6 addresses are as follows:

```
2001:0DB8:7654:3210:FEDC:BA98:7654:3210
2001:0DB8:0:0:8:800:200C:417A
```

IPv6 addresses contain consecutive zeros within the address. You can use two colons (::) at the beginning, middle, or end of an IPv6 address to replace the consecutive zeros. Table 3-1 shows a list of compressed IPv6 address formats.

Note You can use two colons (::) only once in an IPv6 address to replace the longest string of consecutive zeros within the address.

You can use a double colon as part of the IPv6 address when consecutive 16-bit values are denoted as zero. You can configure multiple IPv6 addresses per interface but only one link-local address.

The hexadecimal letters in IPv6 addresses are not case sensitive.

Table 3-1 Compressed IPv6 Address Formats

IPv6 Address Type	Preferred Format	Compressed Format
Unicast	2001:0:0:0:0DB8:800:200C:417A	2001::0DB8:800:200C:417A
Multicast	FF01:0:0:0:0:0:0:101	FF01::101
Loopback	0:0:0:0:0:0:0:1	::1
Unspecified	0:0:0:0:0:0:0:0	::

A node may use the loopback address listed in Table 3-1 to send an IPv6 packet to itself. The loopback address in IPv6 is the same as the loopback address in IPv4. For more information, see Router IDs, page 1-5.

Note You cannot assign the IPv6 loopback address to a physical interface. A packet that contains the IPv6 loopback address as its source or destination address must remain within the node that created the packet. IPv6 routers do not forward packets that have the IPv6 loopback address as their source or destination address.

Note You cannot assign an IPv6 unspecified address to an interface. Additionally, do not use the unspecified IPv6 addresses as destination addresses in IPv6 packets or the IPv6 routing header.

The IPv6 address is specified in hexadecimal using 16-bit values between colons. The IPv6-prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). For example, 2001:0DB8:8086:6502::/32 is a valid IPv6 prefix. For more information, refer to RFC 4291.

IPv6 Unicast Addresses

An IPv6 unicast address is an identifier for a single interface on a single node. A packet that is sent to a unicast address is delivered to the interface identified by that address. This section includes the following topics:

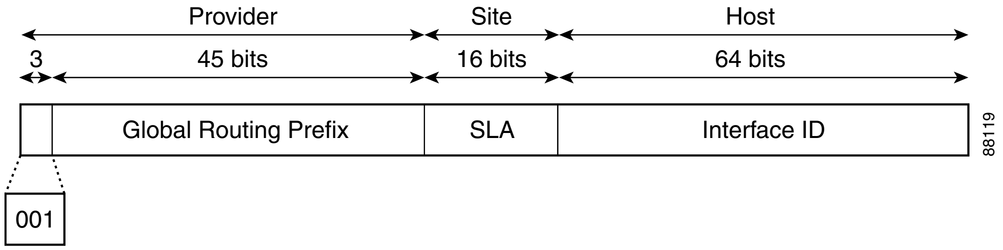
- Aggregatable Global Addresses, page 3-3
- Link-Local Addresses, page 3-5
- Unique Local Addresses, page 3-5
- Unique Local Addresses, page 3-5
- Site-Local Address, page 3-6

Aggregatable Global Addresses

An aggregatable global address is an IPv6 address from the aggregatable global unicast prefix. The structure of aggregatable global unicast addresses enables strict aggregation of routing prefixes that limits the number of routing table entries in the global routing table. Aggregatable global addresses are used on links that are aggregated upward through organizations and eventually to the Internet service providers (ISPs).

Aggregatable global IPv6 addresses are defined by a global routing prefix, a subnet ID, and an interface ID. Except for addresses that start with binary 000, all global unicast addresses have a 64-bit interface ID. The IPv6 global unicast address allocation uses the range of addresses that start with binary value 001 (2000::/3). Figure 3-1 shows the structure of an aggregatable global address.

Figure 3-1 Aggregatable Global Address Format



Addresses with a prefix of 2000::/3 (001) through E000::/3 (111) must have 64-bit interface identifiers in the extended universal identifier (EUI)-64 format. The Internet Assigned Numbers Authority (IANA) allocates the IPv6 address space in the range of 2000::/16 to regional registries.

The aggregatable global address consists of a 48-bit global routing prefix and a 16-bit subnet ID or Site-Level Aggregator (SLA). In the IPv6 aggregatable global unicast address format document (RFC 2374), the global routing prefix included two other hierarchically structured fields called Top-Level Aggregator (TLA) and Next-Level Aggregator (NLA). The IETF decided to remove the TLS and NLA fields from the RFCs because these fields are policy based. Some existing IPv6 networks deployed before the change might still use networks that are on the older architecture.

A subnet ID, which is a 16-bit subnet field, can be used by individual organizations to create a local addressing hierarchy and to identify subnets. A subnet ID is similar to a subnet in IPv4, except that an organization with an IPv6 subnet ID can support up to 65,535 individual subnets.

An interface ID identifies interfaces on a link. The interface ID is unique to the link. In many cases, an interface ID is the same as or based on the link-layer address of an interface. Interface IDs used in aggregatable global unicast and other IPv6 address types have 64 bits and are in the modified EUI-64 format.

Interface IDs are in the modified EUI-64 format in one of the following ways:

- For all IEEE 802 interface types (for example, Ethernet, and Fiber Distributed Data interfaces), the first three octets (24 bits) are the Organizationally Unique Identifier (OUI) of the 48-bit link-layer address (MAC address) of the interface, the fourth and fifth octets (16 bits) are a fixed hexadecimal value of FFFE, and the last three octets (24 bits) are the last three octets of the MAC address. The Universal/Local (U/L) bit, which is the seventh bit of the first octet, has a value of 0 or 1. Zero indicates a locally administered identifier; 1 indicates a globally unique IPv6 interface identifier.
- For all other interface types (for example, serial, loopback, ATM, Frame Relay, and tunnel interface types—except tunnel interfaces used with IPv6 overlay tunnels), the interface ID is similar to the interface ID for IEEE 802 interface types; however, the router uses the first MAC address from the pool of MAC addresses as the identifier (because the interface does not have a MAC address).
- For tunnel interface types that employ IPv6 overlay tunnels, the interface ID is the IPv4 address assigned to the tunnel interface with all zeros in the high-order 32 bits of the identifier.

Note For interfaces that use the Point-to-Point Protocol (PPP), where the interfaces at both ends of the connection might have the same MAC address, the interface identifiers negotiate at both ends of the connection (picked randomly and, if necessary, reconstructed) until both identifiers are unique. The router uses the first MAC address as the identifier for interfaces using PPP.

When no IEEE 802 interface types are in the router, link-local IPv6 addresses are generated on the interfaces in the router in the following sequence:

1. The router is queried for MAC addresses (from the pool of MAC addresses in the router).

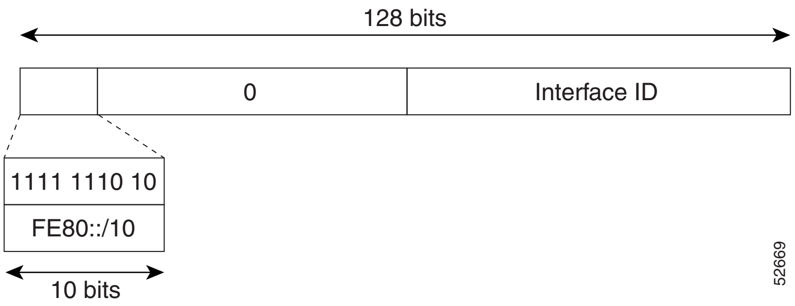
2. When there are no MAC addresses available in the router, the router serial number is used to form the link-local addresses.
3. When the serial number of the router cannot be used to form the link-local addresses, the router uses a Message Digest 5 (MD5) hash to determine the MAC address of the router from the hostname of the router.

Link-Local Addresses

A link-local address is an IPv6 unicast address that can be automatically configured on any interface using the link-local prefix FE80::/10 (1111 1110 10) and the interface identifier in the modified EUI-64 format. Link-local addresses are used in the Neighbor Discovery (ND) protocol and the stateless autoconfiguration process. Devices on a local link can use link-local addresses to communicate; the devices do not need globally unique addresses to communicate. Figure 3-2 shows the structure of a link-local address.

IPv6 routers cannot forward packets that have link-local source or destination addresses to other links.

Figure 3-2 Link-Local Address Format



Unique Local Addresses

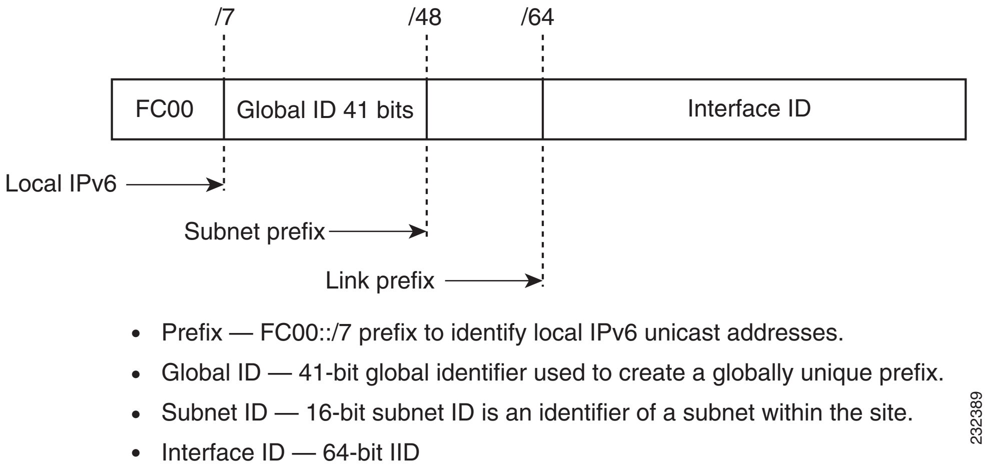
A unique local address is an IPv6 unicast address that is globally unique and is intended for local communications. It is not expected to be routable on the global Internet and is routable inside of a limited area, such as a site, and it may be routed between a limited set of sites. Applications might treat unique local addresses like global scoped addresses.

A unique local address has the following characteristics:

- It has a globally unique prefix (it has a high probability of uniqueness).
- It has a well-known prefix to allow for easy filtering at site boundaries.
- It allows sites to be combined or privately interconnected without creating any address conflicts or requiring renumbering of interfaces that use these prefixes.
- It is ISP-independent and can be used for communications inside of a site without having any permanent or intermittent Internet connectivity.
- When it is accidentally leaked outside of a site through routing or the Domain Name Server (DNS), there is no conflict with any other addresses.

Figure 3-3 shows the structure of a unique local address.

Figure 3-3 Unique Local Address Structure



- Prefix — FC00::/7 prefix to identify local IPv6 unicast addresses.
- Global ID — 41-bit global identifier used to create a globally unique prefix.
- Subnet ID — 16-bit subnet ID is an identifier of a subnet within the site.
- Interface ID — 64-bit IID

Site-Local Address

Because RFC 3879 deprecates the use of site-local addresses, you should follow the recommendations of unique local addressing (ULA) in RFC 4193 when you configure private IPv6 addresses.

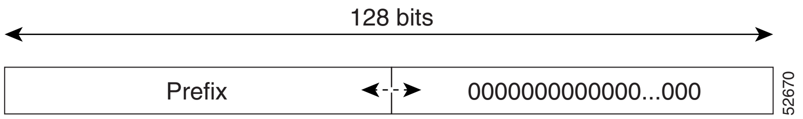
IPv6 Anycast Addresses

An anycast address is an address that is assigned to a set of interfaces that belong to different devices. A packet sent to an anycast address is delivered to the closest interface—as defined by the routing protocols in use—identified by the anycast address. Anycast addresses are syntactically indistinguishable from unicast addresses because anycast addresses are allocated from the unicast address space. Assigning a unicast address to more than one interface turns a unicast address into an anycast address. You must configure the devices to which the anycast address to recognize that the address is an anycast address.

Note Anycast addresses can be used only by a router, not a host. Anycast addresses cannot be used as the source address of an IPv6 packet.

Figure 3-4 shows the format of the subnet router anycast address; the address has a prefix concatenated by a series of zeros (the interface ID). The subnet router anycast address can be used to reach a router on the link that is identified by the prefix in the subnet router anycast address.

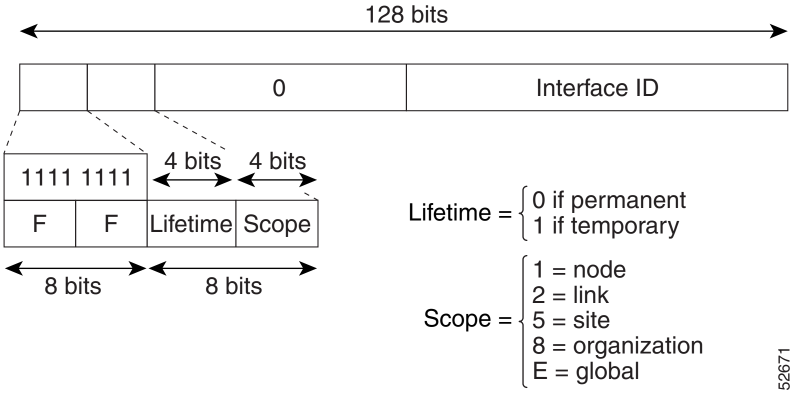
Figure 3-4 Subnet Router Anycast Address Format



IPv6 Multicast Addresses

An IPv6 multicast address is an IPv6 address that has a prefix of FF00::/8 (1111 1111). An IPv6 multicast address is an identifier for a set of interfaces that belong to different devices. A packet sent to a multicast address is delivered to all interfaces identified by the multicast address. The second octet following the prefix defines the lifetime and scope of the multicast address. A permanent multicast address has a lifetime parameter equal to 0; a temporary multicast address has a lifetime parameter equal to 1. A multicast address that has the scope of a device, link, site, or organization, or a global scope, has a scope parameter of 1, 2, 5, 8, or E, respectively. For example, a multicast address with the prefix FF02::/16 is a permanent multicast address with a link scope. Figure 3-5 shows the format of the IPv6 multicast address.

Figure 3-5 IPv6 Multicast Address Format



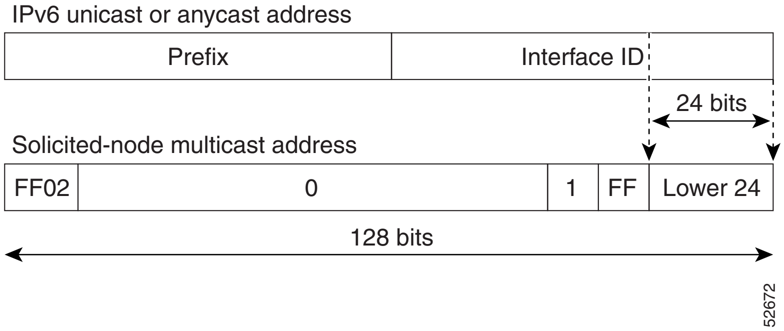
IPv6 nodes (hosts and routers) must join the following multicast groups:

- All-nodes multicast group FF02:0:0:0:0:0:0:1 (the scope is link-local)
- Solicited-node multicast group FF02:0:0:0:0:1:FF00:0000/104 for each of its assigned unicast and anycast addresses

IPv6 routers must also join the all-routers multicast group FF02:0:0:0:0:0:0:2 (the scope is link-local).

The solicited-node multicast address is a multicast group that corresponds to an IPv6 unicast or anycast address. IPv6 nodes must join the associated solicited-node multicast group for every unicast and anycast address to which it is assigned. The IPv6 solicited-node multicast address has the prefix FF02:0:0:0:0:1:FF00:0000/104 concatenated with the 24 low-order bits of a corresponding IPv6 unicast or anycast address (see Figure 3-6). For example, the solicited-node multicast address that corresponds to the IPv6 address 2037::01:800:200E:8C6C is FF02::1:FF0E:8C6C. Solicited-node addresses are used in neighbor solicitation messages.

Figure 3-6 IPv6 Solicited-Node Multicast Address Format

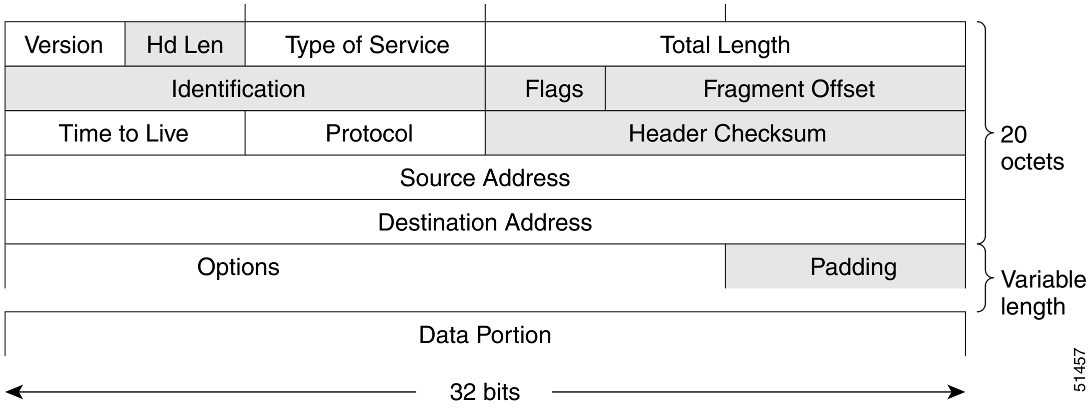

IPv6 unicast or anycast address
Prefix
Interface ID
24 bits
Solicited-node multicast address
FF02
0
1
FF
Lower 24
128 bits
52672

Note IPv6 has no broadcast addresses. IPv6 multicast addresses are used instead of broadcast addresses.

IPv4 Packet Header

The basic IPv4 packet header has 12 fields with a total size of 20 octets (160 bits) (see Figure 3-7). The 12 fields might be followed by an Options field, which is followed by a data portion that is usually the transport-layer packet. The variable length of the Options field adds to the total size of the IPv4 packet header. The shaded fields of the IPv4 packet header are not included in the IPv6 packet header.

Figure 3-7 IPv4 Packet Header Format



Simplified IPv6 Packet Header

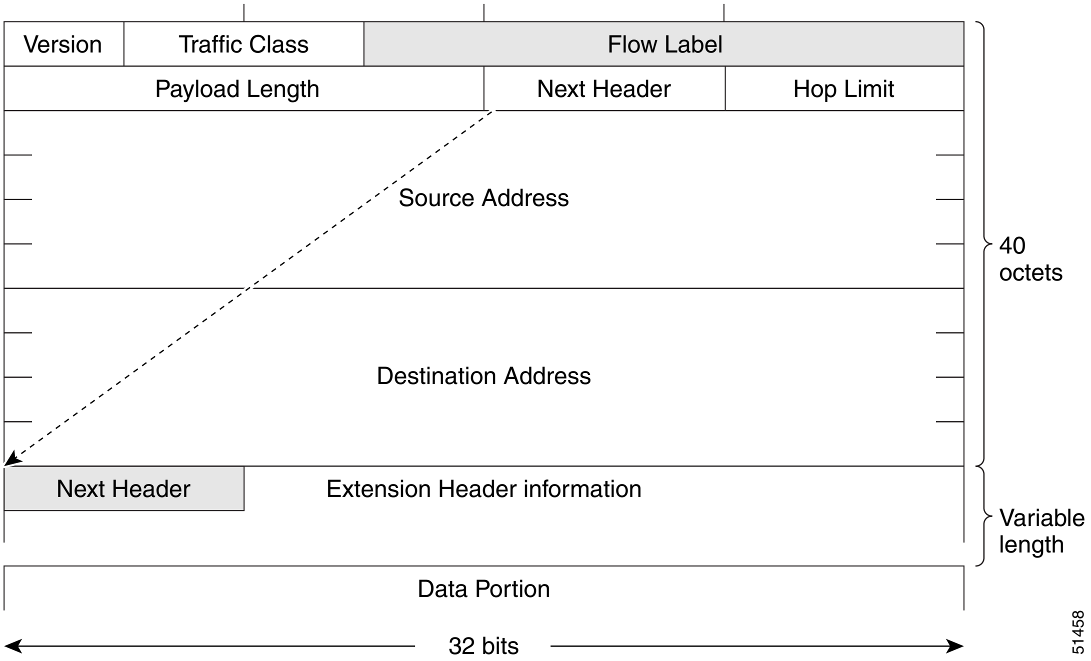
The basic IPv6 packet header has 8 fields with a total size of 40 octets (320 bits) (see Figure 3-8). Fragmentation is handled by the source of a packet and checksums at the data link layer and transport layer are used. The User Datagram Protocol (UDP) checksum checks the integrity of the inner packet and the basic IPv6 packet header and Options field are aligned to 64 bits, which can facilitate the processing of IPv6 packets.

Table 3-2 lists the fields in the basic IPv6 packet header.

Table 3-2 Basic IPv6 Packet Header Fields

Field	Description
Version	Similar to the Version field in the IPv4 packet header, except that the field lists number 6 for IPv6 instead of number 4 for IPv4.
Traffic Class	Similar to the Type of Service field in the IPv4 packet header. The Traffic Class field tags packets with a traffic class that is used in differentiated services.
Flow Label	New field in the IPv6 packet header. The Flow Label field tags packets with a specific flow that differentiates the packets at the network layer.
Payload Length	Similar to the Total Length field in the IPv4 packet header. The Payload Length field indicates the total length of the data portion of the packet.
Next Header	Similar to the Protocol field in the IPv4 packet header. The value of the Next Header field determines the type of information that follows the basic IPv6 header. The type of information that follows the basic IPv6 header can be a transport-layer packet, for example, a TCP or UDP packet, or an Extension Header, as shown in Figure 3-8.
Hop Limit	Similar to the Time to Live field in the IPv4 packet header. The value of the Hop Limit field specifies the maximum number of routers that an IPv6 packet can pass through before the packet is considered invalid. Each router decrements the value by one. Because no checksum is in the IPv6 header, the router can decrement the value without needing to recalculate the checksum, which saves processing resources.
Source Address	Similar to the Source Address field in the IPv4 packet header, except that the field contains a 128-bit source address for IPv6 instead of a 32-bit source address for IPv4.
Destination Address	Similar to the Destination Address field in the IPv4 packet header, except that the field contains a 128-bit destination address for IPv6 instead of a 32-bit destination address for IPv4.

Figure 3-8 IPv6 Packet Header Format



Optional extension headers and the data portion of the packet are after the eight fields of the basic IPv6 packet header. When present, each extension header is aligned to 64 bits. There is no fixed number of extension headers in an IPv6 packet. Each extension header is identified by the Next Header field of the previous header. Typically, the final extension header has a Next Header field of a transport-layer protocol, such as TCP or UDP. Figure 3-9 shows the IPv6 extension header format.

Figure 3-9 IPv6 Extension Header Format

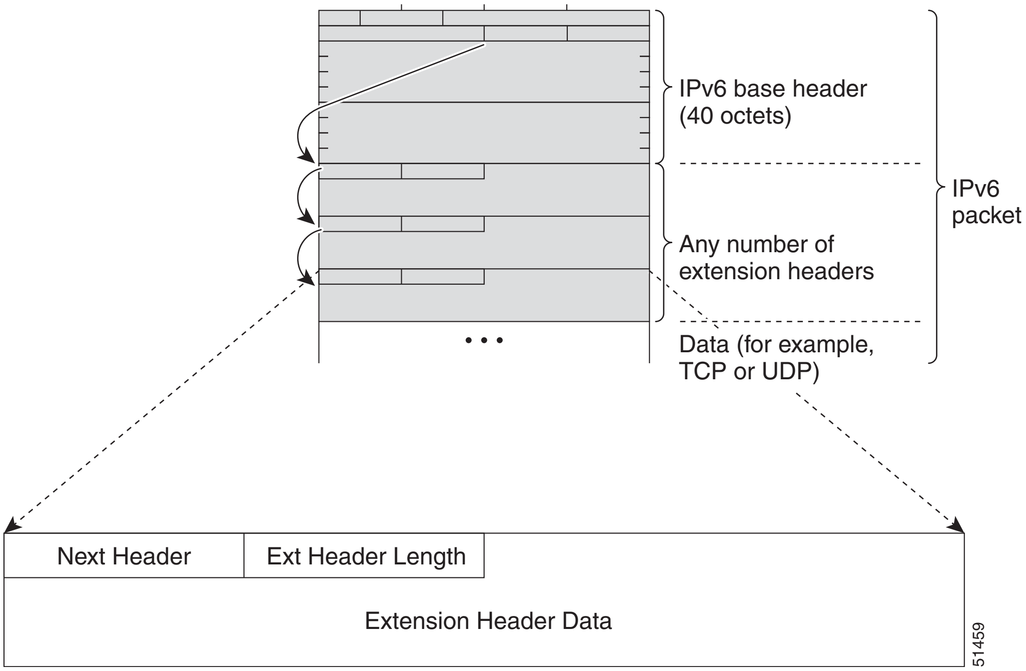

IPv6 base header (40 octets)
IPv6 packet
Any number of extension headers
Data (for example, TCP or UDP)
• • •
Next Header
Ext Header Length
Extension Header Data
51459

Table 3-3 lists the extension header types and their Next Header field values along with their recommended order in the packet.

Table 3-3 IPv6 Extension Header Types and Recommended Packet Order

Header Type	Next Header Value	Description
Hop-by-hop options header	0	Header that is processed by all hops in the path of a packet. When present, the hop-by-hop options header always follows immediately after the basic IPv6 packet header.
Destination options header	6	Header that can follow any hop-by-hop options header. The header is processed at the final destination and at each visited address specified by a routing header. Alternatively, the destination options header can follow any Encapsulating Security Payload (ESP) header. The destination options header is processed only at the final destination.
Routing header	43	Header that is used for source routing**.**
Fragment header	44	Header that is used when a source fragments a packet that is larger than the maximum transmission unit (MTU) for the path between itself and a destination. The Fragment header is used in each fragmented packet.
Upper-layer headers	6 (TCP) 17 (UDP)	Headers within a packet that transport the data. The two main transport protocols are TCP and UDP.

DNS for IPv6

IPv6 supports DNS record types that are supported in the DNS name-to-address and address-to-name lookup processes. The DNS record types support IPv6 addresses (see Table 3-4).

Note IPv6 also supports the reverse mapping of IPv6 addresses to DNS names.

Table 3-4 IPv6 DNS Record Types

Record Type	Description	Format
AAAA	Maps a hostname to an IPv6 address. (Equivalent to an A record in IPv4.)	www.abc.test AAAA 3FFE:YYYY:C18:1::2
PTR	Maps an IPv6 address to a hostname. (Equivalent to a PTR record in IPv4.)	2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8.1.c.0 .y.y.y.y.e.f.f.3.ip6.int PTR www.abc.test

Path MTU Discovery for IPv6

As in IPv4, you can use path MTU discovery in IPv6 to allow a host to dynamically discover and adjust to differences in the MTU size of every link along a data path. In IPv6, however, fragmentation is handled by the source of a packet when the path MTU of one link along a given data path is not large enough to accommodate the size of the packets. Having IPv6 hosts handle packet fragmentation saves IPv6 router processing resources and helps IPv6 networks run more efficiently. Once the path MTU is reduced by the arrival of an ICMP Too Big message, the Cisco CG-OS router retains the lower value. The connection does not increase the segment size to gauge the throughput.

Note In IPv6, the minimum link MTU is 1280 octets. Cisco recommends that you use an MTU value of 1500 octets for IPv6 links.

CDP IPv6 Address Support

You can use the Cisco Discovery Protocol (CDP) IPv6 address support for the neighbor information feature to transfer IPv6 addressing information between two Cisco devices. Cisco Discovery Protocol support for IPv6 addresses provides IPv6 information to network management products and troubleshooting tools.

ICMP for IPv6

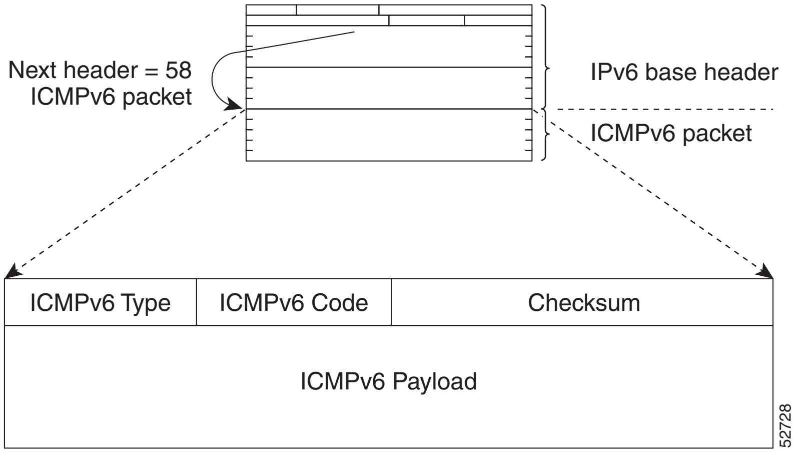
You can use ICMP in IPv6 to provide information about the health of the network. ICMPv6, the version that works with IPv6, reports errors if packets cannot be processed correctly and sends informational messages about the status of the network. For example, if a router cannot forward a packet because it is too large to be sent out on another network, the router sends out an ICMPv6 message to the originating host. Additionally, ICMP packets in IPv6 are used in IPv6 neighbor discovery and path MTU discovery. The path MTU discovery process ensures that a packet is sent using the largest possible size that is supported on a specific route.

A value of 58 in the Next Header field of the basic IPv6 packet header identifies an IPv6 ICMP packet. The ICMP packet follows all the extension headers and is the last piece of information in the IPv6 packet.Within the IPv6 ICMP packets, the ICMPv6 Type and ICMPv6 Code fields identify IPv6 ICMP packet specifics, such as the ICMP message type. The value in the Checksum field is computed by the sender and checked by the receiver from the fields in the IPv6 ICMP packet and the IPv6 pseudo header.

Note The IPv6 header does not have a checksum. However, a checksum on the transport layer can determine when packets are not delivered correctly. All checksum calculations that include the IP address in the calculation must be modified for IPv6 to accommodate the new128-bit address. A pseudo header generates the checksum.

The ICMPv6 Data field contains error or diagnostic information that relates to IP packet processing. Figure 3-10 shows the IPv6 ICMP packet header format.

Figure 3-10 IPv6 ICMP Packet Header Format



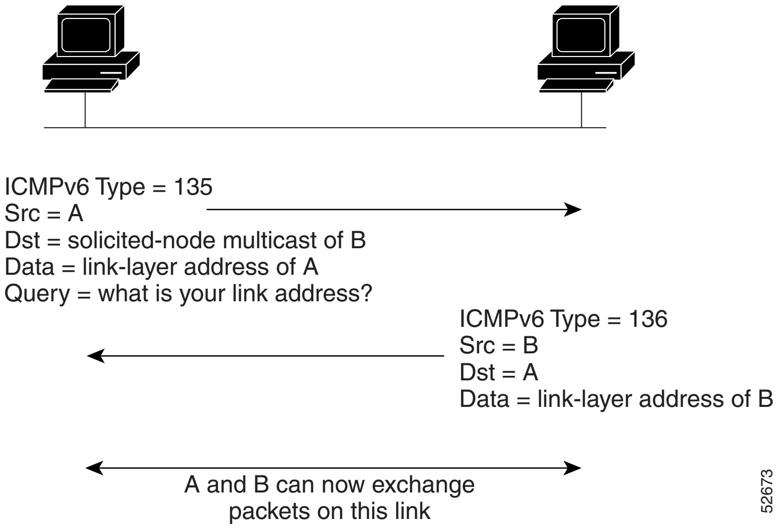
IPv6 Neighbor Discovery

You can use the IPv6 Neighbor Discovery Protocol (NDP) to determine whether a neighboring router is reachable. IPv6 nodes use neighbor discovery to determine the addresses of nodes on the same network (local link), to find neighboring routers that can forward their packets, to verify whether neighboring routers are reachable or not, and to detect changes to link-layer addresses. NDP uses ICMP messages to detect whether packets are sent to neighboring routers that are unreachable.

IPv6 Neighbor Solicitation Message

A node sends a neighbor solicitation message, which has a value of 135 in the Type field of the ICMP packet header, on the local link when it wants to determine the link-layer address of another node on the same local link (see Figure 3-11). The source address is the IPv6 address of the node that sends the neighbor solicitation message. The destination address is the solicited-node multicast address that corresponds to the IPv6 address of the destination node. The neighbor solicitation message also includes the link-layer address of the source node.

Figure 3-11 IPv6 Neighbor Discovery—Neighbor Solicitation Message


ICMPv6 Type = 135
Src = A
Dst = solicited-node multicast of B
Data = link-layer address of A
Query = what is your link address?
ICMPv6 Type = 136
Src = B
Dst = A
Data = link-layer address of B
A and B can now exchange packets on this link
52673

After receiving the neighbor solicitation message, the destination node replies by sending a neighbor advertisement message, which has a value of 136 in the Type field of the ICMP packet header, on the local link. The source address is the IPv6 address of the node (the IPv6 address of the node interface that sends the neighbor advertisement message). The destination address is the IPv6 address of the node that sends the neighbor solicitation message. The data portion includes the link-layer address of the node that sends the neighbor advertisement message.

After the source node receives the neighbor advertisement, the source node and destination node can communicate.

Neighbor solicitation messages can verify the reachability of a neighbor after a node identifies the link-layer address of a neighbor. When a node wants to verify the reachability of a neighbor, it uses the destination address in a neighbor solicitation message as the unicast address of the neighbor.

Neighbor advertisement messages are also sent when there is a change in the link-layer address of a node on a local link. When there is a change, the destination address for the neighbor advertisement is the all-nodes multicast address.

Neighbor unreachability detection identifies the failure of a neighbor or the failure of the forward path to the neighbor and is used for all paths between hosts and neighboring nodes (hosts or routers). Neighbor unreachability detection is performed for neighbors to which only unicast packets are being sent and is not performed for neighbors to which multicast packets are being sent.

A neighbor is considered reachable when a positive acknowledgment is returned from the neighbor (indicating that packets previously sent to the neighbor have been received and processed). A positive acknowledgment—from an upper-layer protocol (such as TCP)—indicates that a connection is making forward progress (reaching its destination). If packets are reaching the peer, they are also reaching the next-hop neighbor of the source. Forward progress is also a confirmation that the next-hop neighbor is reachable.

For destinations that are not on the local link, forward progress implies that the first-hop router is reachable. When acknowledgments from an upper-layer protocol are not available, a node probes the neighbor using unicast neighbor solicitation messages to verify that the forward path is still working. The return of a solicited neighbor advertisement message from the neighbor is a positive acknowledgment that the forward path is still working (neighbor advertisement messages that have the solicited flag set to a value of 1 are sent only in response to a neighbor solicitation message). Unsolicited messages confirm only the one-way path from the source to the destination node; solicited neighbor advertisement messages indicate that a path is working in both directions.

Note A neighbor advertisement message that has the solicited flag set to a value of 0 is not considered as positive acknowledgment that the forward path is still working.

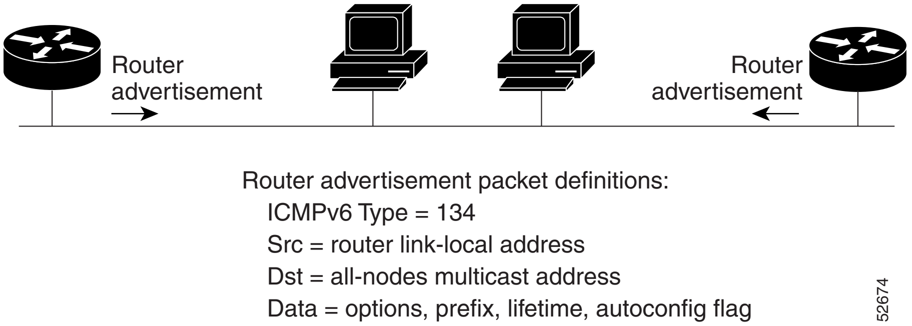
Neighbor solicitation messages are also used in the stateless autoconfiguration process to verify the uniqueness of unicast IPv6 addresses before the addresses are assigned to an interface. Duplicate address detection is performed first on a new, link-local IPv6 address before the address is assigned to an interface (the new address remains in a tentative state while duplicate address detection is performed). A node sends a neighbor solicitation message with an unspecified source address and a tentative link-local address in the body of the message.When another node is already using that address, the node returns a neighbor advertisement message that contains the tentative link-local address. When another node is simultaneously verifying the uniqueness of the same address, that node also returns a neighbor solicitation message. If no neighbor advertisement messages are received in response to the neighbor solicitation message and no neighbor solicitation messages are received from other nodes that are attempting to verify the same tentative address, then the node that sent the original neighbor solicitation message considers the tentative link-local address to be unique and assigns the address to the interface.

IPv6 Router Advertisement Message

IPv6 periodically sends out router advertisement (RA) messages to each configured interface of an IPv6 router. RA messages have a value of 134 in the Type field of the ICMP packet header. For stateless autoconfiguration to work properly, the advertised prefix length in RA messages must always be 64 bits.

The RA messages are sent to the all-nodes multicast address (see Figure 3-12).

Figure 3-12 IPv6 Neighbor Discovery—RA Message



Router advertisement packet definitions:
ICMPv6 Type = 134
Src = router link-local address
Dst = all-nodes multicast address
Data = options, prefix, lifetime, autoconfig flag

RA messages typically include the following information:

- One or more onlink IPv6 prefixes that nodes on the local link can use to automatically configure their IPv6 addresses
- Life-time information for each prefix included in the advertisement
- Sets of flags that indicate the type of autoconfiguration (stateless or stateful) that can be completed
- Default router information (indicates whether the router sending the advertisement is available as a default router and, if so, the amount of time in seconds that the router can be used as a default router)
- Additional information for hosts, such as the hop limit and MTU that a host must use within packets that it originates

RAs are also sent in response to router solicitation messages. Router solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately auto-configure without needing to wait for the next scheduled RA message. The source address is usually the unspecified IPv6 address (0:0:0:0:0:0:0:0). When the host has a configured unicast address, the unicast address of the interface that sends the router solicitation message is used as the source address in the message. The destination address is the all-routers multicast address with a scope of the link. When an RA is sent in response to a router solicitation, the destination address in the RA message is the unicast address of the source of the router solicitation message.

You can configure the following RA message parameters:

- The time interval between periodic RA messages
- The router life-time value, which indicates the usefulness of a router as the default router (for use by all nodes on a given link)
- The network prefixes in use on a given link
- The time interval between neighbor solicitation message retransmissions (on a given link)
- The amount of time that a node considers a neighbor reachable (for use by all nodes on a given link)

The configured parameters are specific to an interface. The sending of RA messages (with default values) is automatically enabled on Ethernet interfaces. For other interface types, you must enter the **no ipv6 nd suppress-ra** command to send RA messages. You can disable the RA message feature on individual interfaces by entering the **ipv6 nd suppress-ra** command.

IPv6 Neighbor Redirect Message

Routers send neighbor redirect messages to inform hosts of better first-hop nodes on the path to a destination (see Figure 3-13). A value of 137 in the Type field of the ICMP packet header identifies an IPv6 neighbor redirect message.

Figure 3-13 IPv6 Neighbor Discovery—Neighbor Redirect Message

Device B
Host H
Device A

Note A router must be able to determine the link-local address for each of its neighboring routers in order to ensure that the target address (the final destination) in a redirect message identifies the neighbor router by its link-local address. For static routing, you should specify the address of the next-hop router using the link-local address of the router. For dynamic routing, you must configure all IPv6 routing protocols to exchange the link-local addresses of neighboring routers.

After forwarding a packet, a router sends a redirect message to the source of the packet under the following circumstances:

- The destination address of the packet is not a multicast address
- The packet was not addressed to the router
- The packet is about to be sent out the interface on which it was received
- The router determines that a better first-hop node for the packet resides on the same link as the source of the packet
- The source address of the packet is a global IPv6 address of a neighbor on the same link or a link-local address

Prerequisites for IPv6

You must be familiar with IPv6 basics such as IPv6 addressing, IPv6 header information, ICMPv6, and the IPv6 Neighbor Discovery (ND) Protocol.

Ensure that you follow the memory and processing guidelines when you make a device a dual-stack device (IPv4/IPv6).

Guidelines and Limitations for IPv6

IPv6 packets are transparent to Layer 2 LAN switches because the switches do not examine Layer 3 packet information before forwarding IPv6 frames. IPv6 hosts can directly attach to Layer 2 LAN switches.

You can configure multiple IPv6 global addresses within the same prefix on an interface. However, you cannot have multiple IPv6 link-local addresses on an interface.

Because RFC 3879 deprecates the use of site-local addresses, you must configure private IPv6 addresses according to the recommendations of unique local addressing (ULA) in RFC 4193.

Default Settings

Table 3-5 lists the default settings for IPv6 parameters.

Table 3-5 Default IPv6 Parameters

Parameters	Default
Neighbor Discovery (ND) reachable time	0 milliseconds
Neighbor solicitation retransmit interval	1000 milliseconds

Configuring IPv6

This section includes the following topics:

- Configuring IPv6 Addressing, page 3-18
- Configuring IPv6 Neighbor Discovery, page 3-20

Configuring IPv6 Addressing

You must configure an IPv6 address on an interface so that the interface can forward IPv6 traffic. When you configure a global IPv6 address on an interface, it automatically configures a link-local address and activates IPv6 for that interface.

BEFORE YOU BEGIN

Be aware of the IPv6 addressing plan employed in the network.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**interface ethernet** *slot/port*	Enters interface configuration mode.

	Command	Purpose
Step 3	**ipv6 address** {*addr* [**eui64**] [**route-preference** *preference*] [**secondary**] **tag** *tag-id*]	Specifies an IPv6 address for an interface and enables IPv6 processing on the interface. Entering the **ipv6 address** command configures a global IPv6 address for the interface. (Optional) Specifying **eui64** in the command assigns an interface identifier (ID) in the low-order 64 bits of the IPv6 address. Only the 64-bit network prefix for the address needs to be specified; the last 64 bits are automatically computed from the interface ID. (Optional) Specifying **router preference** in the command allows a preference for local or direct routes to be set. Range is 0 to 255. (Optional) Specifying **secondary** in the command allows for definition of a secondary IPv6 address. (Optional) Specifying **tag** in the command assigns a value for the local or direct routes.
	ipv6 address *ipv6-address* **use-link-local-only**	Entering the **ipv6 address use-link-local-only** command configures a link-local address for the interface rather than the link-local address that is automatically configured when IPv6 is enabled on the interface. This command enables IPv6 processing on an interface without configuring an IPv6 address.
Step 4	**show ipv6 interface**	(Optional) Displays interfaces configured for IPv6.
Step 5	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to configure an IPv6 address on an interface.

```
router# configure terminal
router(config)# interface ethernet 3/1
router(config-if)# ipv6 address dc3:dc3::/64 eui64
```

This example shows how to configure a specific local-link address on the interface that the interface uses instead of the link-local address that is automatically configured when IPv6 is enabled on the interface.

```
router# configure terminal
router(config)# interface ethernet 3/1
router(config-if)# ipv6 address use-link-local-only
```

This example shows how to display details for an IPv6 interface.

```
router(config-if)# show ipv6 interface ethernet 3/1
Ethernet3/1, Interface status: protocol-down/link-down/admin-down, iod: 36
    IPv6 address: 0dc3:0dc3:0000:0000:0218:baff:fed8:239d
    IPv6 subnet:  0dc3:0dc3:0000:0000:0000:0000:0000:0000/64
    IPv6 link-local address: fe80::0218:baff:fed8:239d (default)
    IPv6 multicast routing: disabled
    IPv6 multicast groups locally joined:
        ff02::0001:ffd8:239d  ff02::0002  ff02::0001  ff02::0001:ffd8:239d
    IPv6 multicast (S,G) entries joined: none
```

```
IPv6 MTU: 1500 (using link MTU)
IPv6 RP inbound packet-filtering policy: none
IPv6 RP outbound packet-filtering policy: none
IPv6 inbound packet-filtering policy: none
IPv6 outbound packet-filtering policy: none
IPv6 interface statistics last reset: never
IPv6 interface RP-traffic statistics: (forwarded/originated/consumed)
    Unicast packets: 0/0/0
    Unicast bytes: 0/0/0
    Multicast packets: 0/0/0
    Multicast bytes: 0/0/0
```

Configuring IPv6 Neighbor Discovery

You can configure IPv6 neighbor discovery (ND) parameters on the router. ND enables IPv6 nodes and routers to determine the link-layer address of a neighbor on the same link, find neighboring routers, and keep track of neighbors.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**interface ethernet** *slot/port*	Enters interface configuration mode.
Step 3	**ipv6 nd reachable-time** *time*	Advertises the time (in seconds) that the router considers neighbor considers a neighbor up after receiving a reachability confirmation within an ICMPv6 router advertisement message. Time is a value between 0 and 9000 seconds. When the router has a reachable time value of 0, it is not the default router.
Step 4	**show ipv6 nd interface** [*type number*] [**detail**]	(Optional) Displays interfaces configured for IPv6 neighbor discovery.
Step 5	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to configure IPv6 neighbor discovery reachable time.

```
router# configure terminal
router(config)# interface ethernet 3/1
router(config-if)# ipv6 nd reachable-time 10
router(config-if)# copy running-config startup-config
```

This example shows how to display an IPv6 neighbor discovery interface:

Note Ignore all references to VRF in the display below. The router does not support the VRF feature; however, it might be seen in command displays.

```
router(config-if)# show ipv6 nd interface ethernet 3/1
ICMPv6 ND Interfaces for VRF "default"
Ethernet3/1, Interface status: protocol-down/link-down/admin-down
IPv6 address: 0dc3:0dc3:0000:0000:0218:baff:fed8:239d
    ICMPv6 active timers:
        Last Neighbor-Solicitation sent: never
        Last Neighbor-Advertisement sent: never
        Last Router-Advertisement sent:never
        Next Router-Advertisement sent in: 0.000000
    Router-Advertisement parameters:
        Periodic interval: 200 to 600 seconds
        Send "Managed Address Configuration" flag: false
        Send "Other Stateful Configuration" flag: false
        Send "Current Hop Limit" field: 64
        Send "MTU" option value: 1500
        Send "Router Lifetime" field: 1800 secs
        Send "Reachable Time" field: 10 ms
        Send "Retrans Timer" field: 0 ms
    Neighbor-Solicitation parameters:
        NS retransmit interval: 1000 ms
    ICMPv6 error message parameters:
        Send redirects: false
        Send unreachables: false
```

OPTIONAL COMMANDS

You can use the following optional IPv6 Neighbor Discovery commands.

Command	Purpose
ipv6 nd hop-limit	Configures the maximum number of hops used in router advertisements and all IPv6 packets that the router originates.
ipv6 nd managed-config-flag	Sets the managed address configuration flag in IPv6 router advertisements. **Note** Command must be configured when DHCPv6 is active in the network.
ipv6 nd mtu	Sets the maximum transmission unit (MTU) size of IPv6 packets sent on an interface.
ipv6 nd ns-interval	Configures the interval between IPv6 neighbor solicitation retransmissions on an interface.
ipv6 nd other-config-flag	Configures the other stateful configuration flag in IPv6 router advertisements.
ipv6 nd prefix *ipv6 address*	Advertises the IPv6 prefix in the router advertisement (RA).
ipv6 nd ra-interval	Configures the interval between IPv6 RA transmissions on an interface.

Command	Purpose
ipv6 nd ra-lifetime	Configures the router lifetime value in IPv6 RAs on an interface.
ipv6 nd redirects	Enables ICMPv6 redirect messages to be sent.
ipv6 nd retrans-timer	Configures the advertised time between neighbor solicitation messages in router advertisements.
ipv6 nd suppress-ra	Suppresses IPv6 router advertisement transmissions on a LAN interface.

Verifying the IPv6 Configuration

To display the IPv6 configuration, use the following commands.

Command	Purpose
show ipv6 interface	Displays IPv6-related interface information.
show ipv6 adjacency	Displays the adjacency table.
show ipv6 icmp	Displays ICMPv6 information.
show ipv6 nd	Displays IPv6 neighbor discovery interface information.
show ipv6 neighbor	Displays IPv6 neighbor entry.

Configuration Example for IPv6

This example shows how to configure IPv6 on an interface.

```
configure terminal
 interface ethernet 3/1
  ipv6 address dc3:dc3::/64 eui64
   ipv6 nd reachable-time 10
```

CHAPTER 4

Configuring IP Services

This chapter describes how to configure the Domain Name Server (DNS) client and enable the Dynamic Host Configuration Protocol version 4 (DHCPv4) client on the Cisco 1000 Series Connected Grid Routers (*hereafter* referred to as the Cisco CG-OS router). The system software for the Cisco CG-OS router is identified as the Cisco CG-OS software.

This chapter includes the following sections:

- Information About IP Services, page 4-1
- Prerequisites, page 4-2
- Guidelines and Limitations, page 4-3
- Default Settings, page 4-3
- Configuring IP Services, page 4-3
- Verifying the Configuration, page 4-5
- Configuration Examples, page 4-6

Information About IP Services

This section includes the following topics:

- DNS Client, page 4-1
- DHCP Client Version 4, page 4-2

DNS Client

When your network devices require connectivity with devices in networks for which you do not control the name assignment, you can assign device names that uniquely identify your devices within the entire internetwork using the domain name server (DNS). DNS uses a hierarchical scheme for establishing host names for network nodes, which allows local control of the segments of the network through a client-server scheme. The DNS system can locate a network device by translating the hostname of the device into its associated IP address.

On the Internet, a domain is a portion of the naming hierarchy tree that refers to general groupings of networks based on the organization type or geography. Domain names are pieced together with periods (.) as the delimiting characters. For example, Cisco is a commercial organization that the Internet identifies by a *com* domain, so its domain name is *cisco.com*. A specific hostname in this domain, the File Transfer Protocol (FTP) system, for example, is identified as *ftp.cisco.com*.

Name Servers

Name servers keep track of domain names and know the parts of the domain tree for which they have complete information. A name server might also store information about other parts of the domain tree. To map domain names to IP addresses on the Cisco CG-OS router, you must identify the hostnames, specify a name server, and enable the DNS service.

The Cisco CG-OS software allows you to statically map IP addresses to domain names. You can also configure the Cisco CG-OS software to use one or more domain name servers to find an IP address for a host name.

DNS Operation

A name server handles client-issued queries to the DNS server for locally defined hosts within a particular zone as follows:

- An authoritative name server responds to DNS user queries for a domain name that is under its zone of authority by using the permanent and cached entries in its own host table. When the query is for a domain name that is under its zone of authority but for which it does not have any configuration information, the authoritative name server replies that no such information exists.
- A name server that is not configured as the authoritative name server responds to DNS user queries by using information that it has cached from previously received query responses. When no Cisco CG-OS router is configured as the authoritative name server for a zone, queries to the DNS server for locally defined hosts receive nonauthoritative responses.

Name servers answer DNS queries (forward incoming DNS queries or resolve internally generated DNS queries) according to the forwarding and lookup parameters configured for the specific domain.

Stateless Restart

The Cisco CG-OS router supports stateless restarts for the DNS client. After a system reboot, the Cisco CG-OS software applies the running configuration.

DHCP Client Version 4

DHCP client version 4 (DHCPv4) support is enabled on a per interface basis on the Cisco CG-OS router for automatic IPv4 address assignment.

The DHCPv4 server, which supplies the IP addresses to the Cisco CG-OS router interfaces, does not need to be on the same subnet as the Cisco CG-OS router. However, when the DHCPv4 server and the Cisco CG-OS router are on different subnets, DHCPv4 relay must be active in the network. Generally, DHCPv4 relay is configured on a router in the path between the Cisco CG-OS router and the DHCPv4 server.

Prerequisites

DNS

You must have a DNS name server accessible to the Cisco CG-OS router.

DHCPv4

You must have a DHCPv4 server accessible to the Cisco CG-OS router.

When the Cisco CG-OS router and the DHCPv4 server are on different subnets, DHCPv4 relay must be enabled on a router in the path between the Cisco CG-OS router and the DHCPv4 server.

Guidelines and Limitations

DHCPv4

DHCPv4 server must be active in the network before you install the Cisco CG-OS router when you employ DHCPv4 client for interface IP address assignment.

DNS

DNS Security Extension (DNSSEC) is not supported on the Cisco CG-OS router.

Default Settings

Table 4-1 lists the default settings for IP Services parameters.

Table 4-1 Default IP Services Parameters

Parameters	Default
DNS client	Enabled
DHCPv4 server	Disabled

Configuring IP Services

This section includes the following topics:

- Configuring DNS Clients, page 4-3
- Enabling DHCPv4 on Interfaces, page 4-4

Configuring DNS Clients

You can configure the DNS client to use a DNS server on your network.

BEFORE YOU BEGIN

Ensure that you have a domain name server on your network.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**ip host** *name address1* [*address2... address6*]	Defines up to six static hostname-to-address mappings in the hostname cache. The address can be either an IPv4 address or an IPv6 address.
Step 3	**ip domain-name** *name*	(Optional) Defines the default domain name that the Cisco CG-OS router uses to complete unqualified host names. The Cisco CG-OS router appends the default domain name to any hostname that does not contain a complete domain name before starting a domain-name lookup.
Step 4	**ip domain-list** *name*	(Optional) Defines additional domain names that the Cisco CG-OS router uses to complete unqualified hostnames. The Cisco CG-OS router uses each entry in the domain list to append that domain name to any hostname that does not contain a complete domain name before starting a domain-name lookup. The process continues for each entry in the domain list until it finds a match.
Step 5	**ip name-server** *address1* [*address2... address6*]	(Optional) Defines up to six name servers. The address can be either an IPv4 address or an IPv6 address.
Step 6	**ip domain-lookup**	(Optional) Enables DNS-based address translation.
Step 7	**show hosts**	(Optional) Displays information about DNS.
Step 8	**copy running-config startup-config**	(Optional) Saves this configuration change by copying it from the running-config file to the startup-config file.

EXAMPLE

This example shows how to configure a default domain name and enable DNS lookup.

```
router# configure terminal
router(config)# ip host cisco-rtp 192.0.2.1
router(config)# ip domain-name myserver.com
router(config)# ip domain-list mycompany.com
router(config)# copy running-config startup-config
```

Enabling DHCPv4 on Interfaces

BEFORE YOU BEGIN

Verify that the Cisco CG-OS router has access to the DHCPv4 server.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface {ethernet | wimax}** *slot/port*	Specifies an interface and enters the interface configuration mode.
Step 3	**ip address dhcp**	Enables DHCPv4 client on the interface to allow automatic assignment of IP addresses to the specified interface. To disable DHCPv4 on the interface, enter **no ip address dhcp** for the interface.
Step 4	**show ip interface brief**	(Optional) Displays abbreviated configuration details for interfaces configured with IPv4 addresses.
Step 5	**show ip interface** *slot/port*	(Optional) Displays configuration details for a specific interface.
Step 6	**copy running-config startup-config**	(Optional) Saves this configuration change by copying it from the running-config file to the startup-config file.

EXAMPLE

This example shows how to enable DHCPv4 client on a specific interface.

```
router# configure terminal
router(config)# interface wimax 3/1
router(config-if)# ip address dhcp
router(config-if)# copy running-config startup-config
```

Verifying the Configuration

To display the DNS client configuration, enter the following command:

Command	Purpose
show hosts	Displays information about DNS.
show interface	Displays configuration information about the interface.
show ip interface *slot/port*	Displays configuration details for a specific IPv4 interface.
show ip interface brief	Displays abbreviated configuration details for interfaces configured with IPv4 addresses.
show running-config interface *slot/port*	Displays the running configuration for the specified interface.

Configuration Examples

DNS Client

This example shows how to establish a domain list with several alternate domain names.

```
ip domain-list csi.com
ip domain-list telecomprog.edu
ip domain-list merit.edu
```

This example shows how to locate local name servers, configure the hostname-to-address mapping, configure the addresses of the two name servers, and the default domain name.

```
ip domain lookup
ip name-server 192.168.1.111 192.168.1.2
ip domain-name cisco.com
```

DHCPv4 Client

This example shows how to enable DHCPv4 client on a specific interface.

```
router# configure terminal
router(config)# interface ethernet 3/1
router(config-if)# ip address dhcp
router(config-if)# copy running-config startup-config
```

CHAPTER 5

Configuring OSPFv2

This chapter describes how to configure Open Shortest Path First version 2 (OSPFv2) for IPv4 networks on the Cisco 1000 Series Connected Grid Routers (*hereafter* referred to as the Cisco CG-OS router). The system software for the router is identified as the Cisco CG-OS software.

This chapter includes the following sections:

Information About OSPFv2

OSPFv2 is an IETF link-state protocol (see Link-State Protocols, page 1-8) for IPv4 networks. An OSPFv2 router sends a special message, called a hello packet, out each OSPF-enabled interface to discover other OSPFv2 neighbor routers. Once a neighbor is discovered, the two routers compare information in the Hello packet to determine if the routers have compatible configurations. The neighbor routers try to establish adjacency, which means that the routers synchronize their link-state databases to ensure that they have identical OSPFv2 routing information. Adjacent routers share link-state advertisements (LSAs) that include information about the operational state of each link, the cost of the link, and any other neighbor information. The routers then flood these received LSAs out every OSPF-enabled interface so that all OSPFv2 routers eventually have identical link-state databases. When all OSPFv2 routers have identical link-state databases, the network is converged (see Convergence, page 1-6). Each router then uses Dijkstra's Shortest Path First (SPF) algorithm to build its route table.

You can divide OSPFv2 networks into areas. Routers send most LSAs only within one area, which reduces the CPU and memory requirements for an OSPF-enabled router.

OSPFv2 supports IPv4, while OSPFv3 supports IPv6. For more information, see Configuring OSPFv3.

This section includes the following topics:

Hello Packet

OSPFv2 routers periodically send Hello packets on every OSPF-enabled interface. The hello interval determines how frequently the router sends these Hello packets. You configure the hello interval on interfaces. OSPFv2 uses Hello packets for the following tasks:

- Neighbor discovery
- Keepalives
- Bidirectional communications
- Designated router election (see Designated Routers, page 5-3)

The Hello packet contains information about the originating OSPFv2 interface and router, including the assigned OSPFv2 cost of the link, the hello interval, and optional capabilities of the originating router. An OSPFv2 interface that receives these Hello packets determines if the settings are compatible with the receiving interface settings. Compatible interfaces are considered neighbors and are added to the neighbor table (see Neighbors, page 5-2).

Hello packets also include a list of router IDs for the routers that communicate with the originating interface. When the receiving interface sees its own router ID in this list, that state confirms that bidirectional communication between the two interfaces exists.

OSPFv2 uses Hello packets as a keepalive message to determine if a neighbor is still communicating. If a router does not receive a Hello packet by the configured dead interval (usually a multiple of the hello interval), then the router removes the neighbor from the local neighbor table.

Neighbors

An OSPFv2 interface must have a compatible configuration with a remote interface before the two can be considered neighbors. The two OSPFv2 interfaces must match the following criteria:

- Hello interval
- Dead interval
- Area ID (see Areas, page 5-4)
- Authentication
- Optional capabilities

When there is a match, the following information is entered into the neighbor table:

- Neighbor ID—The router ID of the neighbor.
- Priority—Priority of the neighbor, which is a component in designated router election (see Designated Routers, page 5-3).
- State—Indication of whether the neighbor has just been heard from, is in the process of setting up bidirectional communications, is sharing the link-state information, or has achieved full adjacency.
- Dead time—Indication of the time since the last Hello packet was received from this neighbor.
- IP Address—The IP address of the neighbor.
- Designated Router—Indication of whether the neighbor has been declared as the designated router or as the backup designated router (see Designated Routers, page 5-3).
- Local interface—The local interface that received the Hello packet for this neighbor.

Adjacency

Adjacency is the path from each router to its local designated router (DR). Not all neighbors establish adjacency. Depending on the network type and designated router establishment, some neighbors become fully adjacent and share LSAs with all their neighbors, while other neighbors do not. For more information, see Designated Routers, page 5-3.

Adjacency is established using Database Description packets, Link State Request packets, and Link State Update packets in OSPFv2. The Database Description packet includes just the LSA headers from the link-state database of the neighbor (see Link-State Database, page 5-7). The local router compares these headers with its own link-state database and determines which LSAs are new or updates. The local router sends a Link State Request packet for each LSA for which it needs new or updated information. The neighbor responds with a Link State Update packet. This exchange continues until both routers have the same link-state information.

Designated Routers

Networks with multiple routers present a unique situation for OSPFv2. When every router floods the network with LSAs, this results in multiple resources sending the same link-state information. Depending on the type of network, OSPFv2 might use a single router, the designated router (DR), to control the LSA floods and represent the network to the rest of the OSPFv2 area (see Areas, page 5-4). When the DR fails, OSPFv2 selects a backup designated router (BDR).

Network types are as follows:

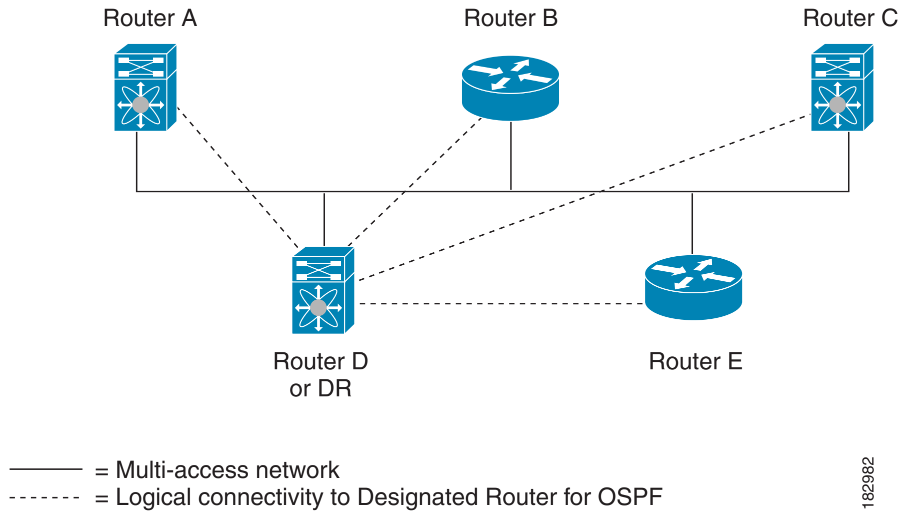
- Point-to-point—A network that exists only between two routers. All neighbors on a point-to-point network establish adjacency and there is no DR.
- Broadcast—A network with multiple routers that can communicate over a shared medium that allows broadcast traffic such as Ethernet. OSPFv2 routers establish a DR and BDR that controls LSA flooding on the network. OSPFv2 uses the well-known IPv4 multicast addresses 224.0.0.5 and a MAC address of 0100.5300.0005 to communicate with neighbors.

OSPFv2 selects the DR and BDR based on the information in the Hello packet. When an interface sends a Hello packet, it sets the priority field and the DR and BDR field when it knows details on the DR and BDR. The routers follow an election procedure based on which routers declare themselves in the DR and BDR fields and the priority field in the Hello packet. As a final determinant, OSPFv2 chooses the router with the highest router IDs as the DR and BDR.

All other routers establish adjacency with the DR and the BDR and use the IPv4 multicast address 224.0.0.6 to send LSA updates to the DR and BDR. Figure 5-1 shows this adjacency relationship between all routers and the DR.

DRs are tied to router interfaces. A router might be the DR for one network and not for another network that it connects through a different interface.

Figure 5-1 Designated Router in Multi-Access Network



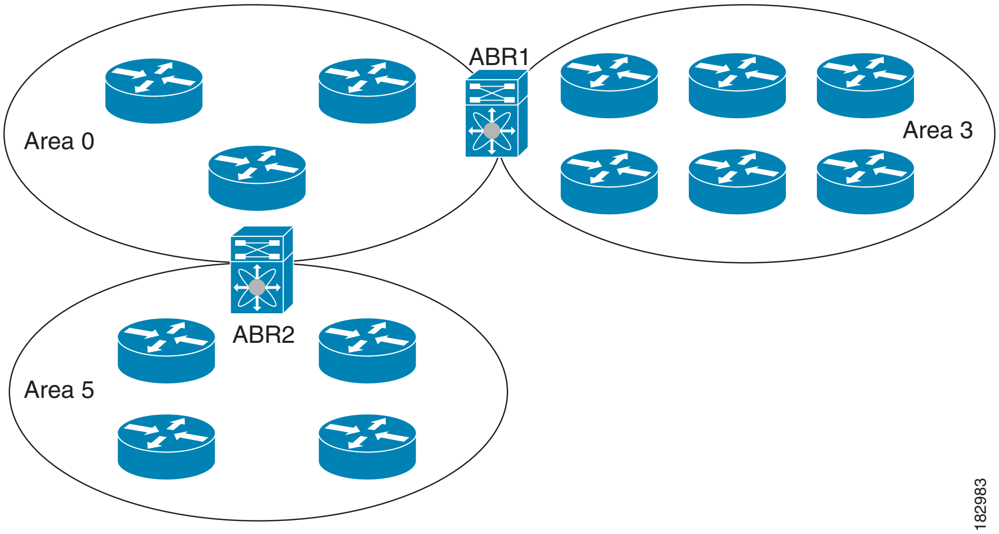
Areas

You can limit the CPU and memory requirements that OSPFv2 puts on the routers by dividing an OSPFv2 network into areas. An area is a logical division of routers and links within an OSPFv2 domain that creates separate subdomains. LSA flooding is contained within an area, and the link-state database can only access links within the area. You can assign an area ID to the interfaces within the defined area. The Area ID is a 32-bit value that you can enter as a number or in dotted decimal notation such as 10.2.3.1.

The Cisco CG-OS software always displays the area in dotted decimal notation.

When you define more than one area in an OSPFv2 network, you must also define the backbone area, which has the reserved area ID of 0. If you have more than one area, then one or more routers become area border routers (ABRs). An ABR connects to both the backbone area and at least one other defined area (see Figure 5-2).

Figure 5-2 OSPFv2 Areas



The ABR has a separate link-state database for each area to which it connects. The ABR sends Network Summary (type 3) LSAs (see Route Summarization, page 5-10) from one connected area to the backbone area. The backbone area sends summarized information about one area to another area. In Figure 5-2, Area 0 sends summarized information about Area 5 to Area 3.

OSPFv2 defines one other router type: the autonomous system boundary router (ASBR). This router connects an OSPFv2 area to another autonomous system. OSPFv2 can redistribute its routing information into another autonomous system or receive redistributed routes from another autonomous system. For more information, see Advanced Features, page 5-8.

Link-State Advertisements

OSPFv2 uses link-state advertisements (LSAs) to build its routing table.

This section includes the following topics:

- LSA Types, page 5-5
- Link Cost, page 5-6
- Flooding and LSA Group Pacing, page 5-6
- Link-State Database, page 5-7
- Opaque LSAs, page 5-7

LSA Types

Table 5-1 shows the LSA types supported by the Cisco CG-OS software.

Table 5-1 LSA Types

Type	Name	Description
1	Router LSA	LSA sent by every router. This LSA includes the state and the cost of all links and a list of all OSPFv2 neighbors on the link. Router LSAs trigger an SPF recalculation. Router LSAs flood the local OSPFv2 area.
2	Network LSA	LSA sent by the DR. This LSA lists all routers in the multi-access network. Network LSAs trigger an SPF recalculation. See Designated Routers, page 5-3.
3	Network Summary LSA	LSA sent by the area border router to an external area for each destination in the local area. This LSA includes the link cost from the area border router to the local destination. See Areas, page 5-4.
4	ASBR Summary LSA	LSA sent by the area border router to an external area. This LSA advertises the link cost to the ASBR only. See Areas, page 5-4.
5	AS External LSA	LSA generated by the ASBR. This LSA includes the link cost to an external autonomous system destination. AS External LSAs flood the autonomous system. See Areas, page 5-4.
7	NSSA External LSA	LSA generated by the ASBR within a not-so-stubby area (NSSA). This LSA includes the link cost to an external autonomous system destination. NSSA External LSAs are flooded only within the local NSSA. See Areas, page 5-4.
9–11	Opaque LSAs	LSA used to extend OSPF. See Opaque LSAs, page 5-7.

Link Cost

Each OSPFv2 interface has a link cost. The cost is an arbitrary number. By default, the Cisco CG-OS software assigns a cost that is the configured reference bandwidth divided by the interface bandwidth. By default, the reference bandwidth is 40 Gb/s. Each LSA update contains the link cost.

Flooding and LSA Group Pacing

When an OSPFv2 router receives an LSA, it forwards that LSA out every OSPF-enabled interface, flooding the OSPFv2 area with this information. This LSA flooding guarantees that all routers in the network have identical routing information. LSA flooding depends on the OSPFv2 area configuration (see Areas, page 5-4). The link-state refresh time (every 30 minutes by default) determines the frequency of LSA flooding within an area. Each LSA has its own link-state refresh time.

You can control the flooding rate of LSA updates in your network by using the LSA group pacing feature. LSA group pacing can reduce high CPU or buffer usage. This feature groups LSAs with similar link-state refresh times to allow OSPFv2 to pack multiple LSAs into an OSPFv2 Update message.

By default, LSAs with link-state refresh times within four minutes of each other are grouped together. You can lower this value for large link-state databases or raise it for smaller databases to optimize the OSPFv2 load on your network.

Link-State Database

Each router maintains a link-state database for the OSPFv2 network. This database contains all the collected LSAs, and includes information on all the routes through the network. OSPFv2 uses this information to calculate the best path to each destination and populates the routing table with these best paths.

The router removes LSAs from the link-state database when it does not receive an LSA update within a set interval (MaxAge). Routers flood a repeat of the LSA every 30 minutes to prevent accurate link-state information from aging out. The Cisco CG-OS software supports the LSA grouping feature to prevent all LSAs from refreshing at the same time. For more information, see Flooding and LSA Group Pacing, page 5-6.

Opaque LSAs

Opaque LSAs allow you to extend OSPF functionality. Opaque LSAs consist of a standard LSA header followed by application-specific information. This information might be used by OSPFv2 or by other applications. OSPFv2 uses Opaque LSAs to support OSPFv2 Graceful Restart capability. Three Opaque LSA types are defined as follows:

- LSA type 9—Flooded to the local network.
- LSA type 10—Flooded to the local area.
- LSA type 11—Flooded to the local autonomous system.

OSPFv2 and the Unicast RIB

OSPFv2 runs the Dijkstra shortest path first algorithm on the link-state database. This algorithm selects the best path to each destination based on the sum of all the link costs for each link in the path. The shortest path for each destination is then put in the OSPFv2 route table. When the OSPFv2 network is converged, this route table feeds into the unicast RIB. OSPFv2 communicates with the unicast RIB to do the following:

- Add or remove routes
- Provide convergence updates to remove stale OSPFv2 routes and for stub router advertisements (see OSPFv2 Stub Router Advertisements, page 5-11).

OSPFv2 also runs a modified Dijkstra algorithm for fast recalculation for summary and external (type 3, 4, 5, and 7) LSA changes.

Authentication

You can configure authentication on OSPFv2 messages to prevent unauthorized or invalid routing updates in your network. The Cisco CG-OS software supports two authentication methods:

- Simple password authentication
- MD5 authentication digest

You can configure the OSPFv2 authentication for an OSPFv2 area or per interface.

Simple Password Authentication

Simple password authentication uses a simple clear-text password that the router sends as part of the OSPFv2 message. You must ensure that the receiving OSPFv2 router has the same clear-text password configured so that it accepts the OSPFv2 message as a valid route update. Because the password is in clear text, anyone who can watch traffic on the network can learn the password.

MD5 Authentication

Cisco recommends using MD5 authentication to authenticate OSPFv2 messages. With MD5 authentication, you configure a password that is shared by the local router and all remote OSPFv2 neighbors. For each OSPFv2 message, the Cisco CG-OS software creates an MD5 one-way message digest based on the message itself and the encrypted password. The interface sends this digest with the OSPFv2 message. The receiving OSPFv2 neighbor validates the digest using the same encrypted password. If the message has not changed, then the digest calculation is identical and the OSPFv2 message is considered valid.

MD5 authentication includes a sequence number with each OSPFv2 message to ensure that no message is replayed in the network.

Advanced Features

The Cisco CG-OS software supports advanced OSPFv2 features that enhance the usability and scalability of OSPFv2 in the network. This section includes the following topics:

- Stub Area, page 5-8
- Not-So-Stubby Area, page 5-9
- Virtual Links, page 5-9
- Route Summarization, page 5-10
- Configuring Graceful Restart, page 5-29
- OSPFv2 Stub Router Advertisements, page 5-11
- Multiple OSPFv2 Instances, page 5-11
- SPF Optimization, page 5-11

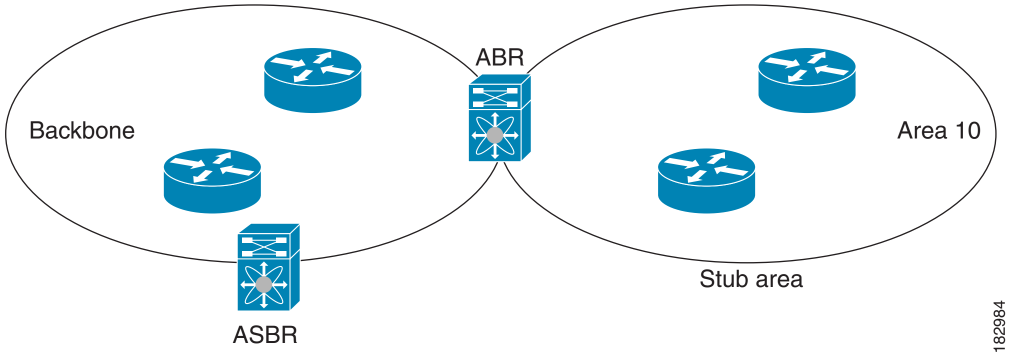
Stub Area

You can limit the amount of external routing information that floods an area by making it a stub area. A stub area is an area that does not allow AS External (type 5) LSAs (see Link-State Advertisements, page 5-5). These LSAs usually flood the local autonomous system to propagate external route information. Stub areas have the following requirements:

- All routers in the stub area must be stub routers. See Stub Routing, page 1-6.
- No ASBR routers exist in the stub area.
- You cannot configure virtual links in the stub area.

Figure 5-3 shows an example of an OSPFv2 autonomous system where all routers in area 0.0.0.10 have to go through the ABR to reach external autonomous systems. Area 0.0.0.10 can be configured as a stub area.

Figure 5-3 Stub Area



Stub areas use a default route for all traffic that must go through the backbone area to the external autonomous system. The default route is 0.0.0.0 for IPv4.

Not-So-Stubby Area

A Not-so-Stubby Area (NSSA) is similar to a stub area, except that an NSSA allows you to import autonomous system external routes within an NSSA by using redistribution. The NSSA ASBR redistributes these routes and generates NSSA External (type 7) LSAs that it floods throughout the NSSA. You can optionally configure the ABR that connects the NSSA to other areas to translate this NSSA External LSA to AS External (type 5) LSAs. The ABR then floods these AS External LSAs throughout the OSPFv2 autonomous system. The router supports summarization and filtering during the translation. See Link-State Advertisements, page 5-5 for information about NSSA External LSAs.

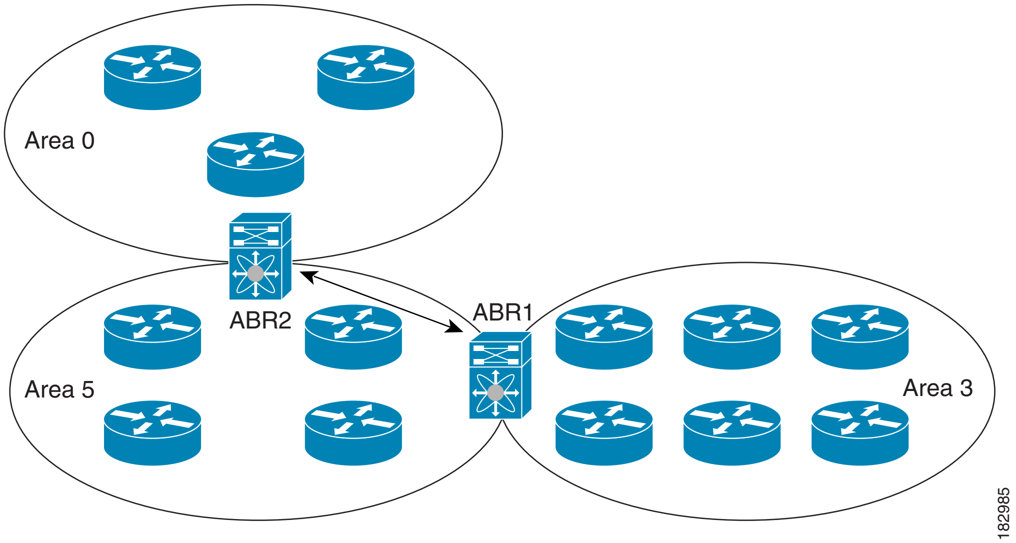
You can, for example, use NSSA to simplify administration when you are connecting a central site using OSPFv2 to a remote site that is using a different routing protocol. Before NSSA, the connection between the corporate site border router and a remote router could not be run as an OSPFv2 stub area because routes for the remote site could not be redistributed into a stub area. With NSSA, you can extend OSPFv2 to cover the remote connection by defining the area between the corporate router and the remote router as an NSSA (see Configuring NSSA, page 5-22).

The backbone Area 0 cannot be an NSSA.

Virtual Links

Virtual links allow you to connect an OSPFv2 area ABR to a backbone area ABR when a direct physical connection is not available. Figure 5-4 shows a virtual link that connects Area 3 to the backbone area through Area 5.

Figure 5-4 Virtual Links



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

Route Summarization

Because OSPFv2 shares all learned routes with every OSPF-enabled router, you might want to use route summarization to reduce the number of unique routes that are flooded to every OSPF-enabled router. Route summarization simplifies route tables by replacing more-specific addresses with an address that represents all the specific addresses. For example, you can replace 10.1.1.0/24, 10.1.3.0/24, and 10.1.5.0/24 with one summary address, 10.1.0.0/24.

Typically, you would summarize at the boundaries of area border routers (ABRs). Although you could configure summarization between any two areas, it is better to summarize in the direction of the backbone so that the backbone receives all the aggregate addresses and injects them, already summarized, into other areas.

Inter-area route summarization

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

Graceful Restart

OSPFv2 automatically restarts when the process experiences problems. After the restart, OSPFv2 initiates a graceful restart so that the platform is not taken out of the network topology. When you manually restart OSPF, it performs a graceful restart, which is similar to a stateful switchover. The router applies the running configuration in both cases.

A graceful restart, known as nonstop forwarding (NSF), allows OSPFv2 to remain in the data forwarding path through a process restart. When OSPFv2 needs to restart, it first sends a link-local opaque (type 9) LSA, called a grace LSA (see Opaque LSAs, page 5-7).

The grace LSA includes a grace period, which is a specified time that the neighbor OSPFv2 interfaces hold onto the LSAs from the restarting OSPFv2 interface. (Typically, OSPFv2 tears down the adjacency and discards all LSAs from a down or restarting OSPFv2 interface.) The participating neighbors, which are called NSF helpers, keep all LSAs that originate from the restarting OSPFv2 interface as if the interface were still adjacent.

When the restarting OSPFv2 interface is operational again, it rediscovers its neighbors, establishes adjacency, and starts sending its LSA updates again. At this point, the NSF helpers recognize that graceful restart has finished.

Note When the restarting OSPFv2 interface does not come back up before the end of the grace period, or if the network experiences a topology change, the OSPFv2 neighbors tear down adjacency with the restarting OSPFv2 and treat it as a normal OSPFv2 restart.

OSPFv2 Stub Router Advertisements

You can configure an OSPFv2 interface to act as a stub router using the OSPFv2 Stub Router Advertisements feature. Use this feature when you want to limit the OSPFv2 traffic through this router, such as when you want to introduce a new router to the network in a controlled manner or limit the load on a router that is already overloaded. You might also want to use this feature for various administrative or traffic engineering reasons.

OSPFv2 stub router advertisements do not remove the OSPFv2 router from the network topology, but they do prevent other OSPFv2 routers from using this router to route traffic to other parts of the network. Only the traffic that is destined for this router or directly connected to this router is sent.

OSPFv2 stub router advertisements mark all stub links (directly connected to the local router) to the cost of the local OSPFv2 interface. The router marks all remote links with the maximum cost (0xFFFF).

Multiple OSPFv2 Instances

The Cisco CG-OS software supports multiple instances of the OSPFv2 protocol that run on the same node. You cannot configure multiple instances over the same interface. By default, every instance uses the same system router ID. You must manually configure the router ID for each instance if the instances are in the same OSPFv2 autonomous system.

SPF Optimization

The Cisco CG-OS software optimizes the SPF algorithm in the following ways:

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

Prerequisites for OSPFv2

Your OSPFv2 network strategy and planning for your network is complete. For example, you must decide whether your network requires multiple areas.

You must be familiar with routing fundamentals to configure OSPFv2.

You are logged on to the router.

You have enabled the OSPFv2 feature on your router (see Enabling OSPFv2, page 5-13).

You have configured at least one interface for IPv4 that can communicate with a remote OSPFv2 neighbor.

Guidelines and Limitations for OSPFv2

The Cisco CG-OS software displays areas in dotted decimal notation regardless of whether you enter the area in decimal or dotted decimal notation.

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

You cannot add a virtual link or ASBRs to a stub area.

You cannot add a virtual link to a Not-So-Stubby-Area (NSSA).

You cannot add a NSSA in a backbone area.

Default Settings

Table 5-2 lists the default settings for OSPFv2 parameters.

Table 5-2 Default OSPFv2 Parameters

Parameters	**Default**
Hello interval	10 seconds
Dead interval	40 seconds
Graceful restart grace period	60 seconds
Graceful restart notify period	15 seconds
OSPFv2 feature	Disabled
Stub router advertisement announce time	600 seconds
Reference bandwidth for link cost calculation	40 Gb/s
LSA minimal arrival time	1000 milliseconds
LSA group pacing	240 seconds
SPF calculation initial delay time	200 milliseconds
SPF minimum hold time	5000 milliseconds
SPF calculation initial delay time	1000 milliseconds

Configuring Basic OSPFv2

Configure OSPFv2 after you design your OSPFv2 network.

This section includes the following topics:

- Enabling OSPFv2, page 5-13
- Creating an OSPFv2 Instance, page 5-13
- Configuring Optional Parameters on an OSPFv2 Instance, page 5-14
- Configuring Networks in OSPFv2, page 5-15
- Configuring Authentication for an Area, page 5-16
- Configuring Authentication for an Interface, page 5-18

Enabling OSPFv2

You must enable the OSPFv2 feature before you can configure OSPFv2.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**feature ospf**	Enables the OSPFv2 feature.
Step 3	**show feature**	(Optional) Displays enabled and disabled features.
Step 4	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to enable OSPFv2 on the router.

```
router# configure terminal
router(config)# feature ospf
router(config)# copy running-config startup-config
```

To disable the OSPFv2 feature and remove all associated configuration, use the **no feature ospf** command in configuration mode.

Creating an OSPFv2 Instance

The first step in configuring OSPFv2 is to create an OSPFv2 instance; and, then to assign a unique instance tag for this OSPFv2 instance. The instance tag can be any string.

For more information about OSPFv2 instance parameters, see Configuring Advanced OSPFv2, page 5-19.

BEFORE YOU BEGIN

Enter the **show feature** command to verify that OSPFv2 is enabled (see Enabling OSPFv2, page 5-13).

Enter the **show ip ospf** command to verify that the instance tag is not in use.

Ensure that OSPFv2 can obtain a router identifier (router ID) such as a configured loopback address. If not, you must configure the router ID option as shown in the Detailed Steps section below.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag and enters the router configuration area.
Step 3	**router-id** *32-bit number*	(Optional) Configures the OSPFv2 router ID. This 32-bit number identifies this OSPFv2 instance and must exist on a configured interface in the system.
Step 4	**show ip ospf**	(Optional) Displays OSPF information.
Step 5	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to create an OSPFv2 instance.

```
router# configure terminal
router(config)# router ospf 201
router(config-router)# copy running-config startup-config
```

To remove the OSPFv2 instance and all associated configuration, use the **no router ospf** *instance-tag* command in configuration mode.

Note When configured in the interface mode, the **no router ospf** *instance-tag* command does not remove the OSPF configuration. You must manually remove all OSPFv2 commands configured in interface mode.

Configuring Optional Parameters on an OSPFv2 Instance

You can configure optional parameters for OSPFv2.

For more information about OSPFv2 instance parameters, see Configuring Advanced OSPFv2, page 5-19.

BEFORE YOU BEGIN

Ensure that you have enabled the OSPFv2 feature on the router (see Enabling OSPFv2, page 5-13).

OSPFv2 must be able to obtain a router identifier (for example, a configured loopback address) or you must configure the router ID option.

OPTIONAL PARAMETERS

You can configure the following optional parameters for OSPFv2 in the router configuration mode by entering the **router ospf** *instance-tag* command.

Command	Purpose
distance *number*	Configures the administrative distance for this OSPFv2 instance. The range is from 1 to 255. The default is 110.
log-adjacency-changes [detail]	Generates a system message whenever a neighbor changes state.
maximum-paths *path-number*	Configures the maximum number of equal OSPFv2 paths to a destination in the route table. This command is used for load balancing. The range is from 1 to 16. The default is 8.

Configuring Networks in OSPFv2

You can configure a network to OSPFv2 by associating it through the interface that the router uses to connect to that network (see Neighbors, page 5-2). You can add all networks to the default backbone area (Area 0), or you can create new areas using any decimal number or an IP address.

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

Note You must assign a valid IP address to an interface before you can enable OSPF on that interface.

BEFORE YOU BEGIN

Ensure that you have enabled OSPF on the router (see Enabling OSPFv2, page 5-13).

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	**ip address** *ip-prefix/length*	Assigns an IP address and subnet mask to this interface
Step 4	**ip router ospf** *instance-tag* **area** *area-id* [**secondaries none**]	Adds the interface to the OSPFv2 instance and area.
Step 5	**show ip ospf interface** [*interface-type*] [*slot/port*]	(Optional) Displays OSPF information.
Step 6	**copy running-config startup-config**	(Optional) Saves this configuration change.

OPTIONAL COMMANDS

You can configure the following optional parameters for OSPFv2 in interface configuration mode.

Command	Purpose
ip ospf cost *interface-cost*	Configures the OSPFv2 cost metric for this interface. The default is to calculate cost metric, based on the reference bandwidth and interface bandwidth. The range is from 1 to 65535. Use the no **ip ospf cost** *interface-cost* command to return to the default setting which is the reference.
ip ospf dead-interval *seconds*	Configures the OSPFv2 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.
ip ospf hello-interval *seconds*	Configures the OSPFv2 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.
ip ospf mtu-ignore	Configures OSPFv2 to ignore any IP MTU mismatch with a neighbor. The default is to not establish adjacency if the neighbor MTU does not match the local interface MTU.
ip ospf passive-interface	Suppresses routing updates on the interface.
ip ospf priority *number*	Configures the OSPFv2 priority, that the router uses to determine the DR for an area. The range is from 0 to 255. The default is 1. See Designated Routers, page 5-3.
ip ospf shutdown	Shuts down the OSPFv2 instance on this interface.

EXAMPLE

This example shows how to add an interface into area 0.0.0.10 in OSPFv2 instance 201.

```
router# configure terminal
router(config)# interface ethernet 2/1
router(config-if)# ip address 192.0.2.1/16
router(config-if)# ip router ospf 201 area 0.0.0.10
router(config-if)# copy running-config startup-config
```

Use the **show ip ospf interface** command to verify the interface configuration. Use the **show ip ospf neighbor** command to see the neighbors for this interface.

To remove the area, use the **no ip router ospf** *instance-tag* **area** *area-id* command.

Configuring Authentication for an Area

You can configure authentication for all networks in an area or for individual interfaces in the area. Interface authentication configuration overrides area authentication.

BEFORE YOU BEGIN

Ensure that OSPFv2 is enabled on the router (see Enabling OSPFv2, page 5-13).

Ensure that all neighbors on an interface share the same authentication configuration, including the shared authentication key.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	**area** *area-id* **authentication** [**message-digest**]	Configures the authentication mode for an area. Area identifier (*area-id*) for an OSPF area can be an IP address or a positive integer value.
Step 4	**interface** *interface-type slot/port*	Enters interface configuration mode.
Step 5	**ip ospf authentication-key** [**0** | **3**] *password*	(Optional) Configures simple password authentication for this interface. Use this command if the authentication is not set to key-chain or message-digest. • Assigning an authentication key value of zero (0) configures an unencrypted password. • Assigning an authentication key value of 3 configures a 3DES encrypted password.
	ip ospf message-digest-key *key-id* **md5 [0 | 3]** *key*	(Optional) Configures message digest authentication for this interface. The *key-id* range is from 1 to 255. The MD5 option 0 configures the password in clear text and 3 configures the key as 3DES encrypted.
Step 6	**show ip ospf interface** *interface-type slot/port*	(Optional) Displays OSPF information.
Step 7	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to configure authentication for area 0.0.0.10 in OSPFv2 instance 201.

```
router# configure terminal
router(config)# router ospf 201
router(config-router)# area 0.0.0.10 authentication
router(config-router)# interface ethernet 2/1
router(config-if)# ip ospf authentication-key 0 mypass
router(config-if)# copy running-config startup-config
```

To terminate an OSPF routing process, use the **no router ospf** *instance-tag* command.

Configuring Authentication for an Interface

You can configure authentication for individual interfaces in an area.

Note Interface authentication configuration overrides area authentication settings.

BEFORE YOU BEGIN

Ensure that OSPFv2 is enabled on the router (see Enabling OSPFv2, page 5-13).

Ensure that all neighbors on an interface share the same authentication configuration, including the shared authentication key.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	**ip ospf authentication [message-digest | null]**	Enables interface authentication mode for OSPFv2 as either key-chain, message-digest type, or null. **Note** Entering the *null* option specifies that no authentication is in use. Entering this command overrides any previously defined OSPF area-based authentication for this interface. All neighbors must share this authentication type.

	Command	Purpose
Step 4	**ip ospf authentication-key [0 | 3 | 7]** *key-name*	(Optional) Configures simple password authentication for this interface. Use this command if the authentication is set to key-chain. The options are as follows: • 0—Configures the password in clear text (unencrypted). • 3—Configures the pass key as 3DES encrypted. • 7—Configures the key as Cisco type 7 encrypted.
	ip ospf message-digest-key *key-id* **md5 [0 | 3 | 7]** *key*	(Optional) Configures message-digest authentication for this interface. Use this command if the authentication is set to message-digest. The *key-id* range is from 1 to 255. The MD5 options are as follows: • 0—Configures the password in clear text (unencrypted). • 3—Configures the pass key as 3DES encrypted. • 7—Configures the key as Cisco type 7 encrypted.
Step 5	**show ip ospf interface** *interface-type slot/port*	(Optional) Displays OSPF information.
Step 6	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to set an interface for simple, unencrypted passwords, and set the password for Ethernet interface 2/1.

```
router# configure terminal
router(config)# router ospf 201
router(config-router)# exit
router(config)# interface ethernet 2/1
router(config-if)# ip router ospf 201 area 0.0.0.10
router(config-if)# ip ospf authentication-key 0 ifpass
router(config-if)# copy running-config startup-config
```

To terminate an OSPF routing process, use the **no router ospf** *instance-tag* command.

Configuring Advanced OSPFv2

Configure OSPFv2 after you design your OSPFv2 network.

This section includes the following topics:

- Configuring Filter Lists for Border Routers, page 5-20
- Configuring Stub Areas, page 5-21
- Configuring a Totally Stubby Area, page 5-22

- Configuring NSSA, page 5-22
- Configuring Virtual Links, page 5-23
- Configuring Route Summarization, page 5-25
- Configuring Stub Route Advertisements, page 5-26
- Modifying the Default Timers, page 5-27
- Configuring Graceful Restart, page 5-29
- Restarting an OSPFv2 Instance, page 5-30

Configuring Filter Lists for Border Routers

You can separate your OSPFv2 domain into a series of areas that contain related networks. All areas must connect to the backbone area through an area border router (ABR). OSPFv2 domains can connect to external domains through an *autonomous system border router* (ASBR). See Areas, page 5-4.

ABRs have the following optional configuration parameters:

- Area range—Configures route summarization between areas. See Configuring Route Summarization, page 5-25.
- Filter list—Filters the Network Summary (type 3) LSAs that are allowed in from an external area.

ASBRs also support filter lists.

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see Enabling OSPFv2, page 5-13).

Create the route map that the filter list uses to filter IP prefixes in incoming or outgoing Network Summary (type 3) LSAs.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	**area** *area-id* **filter-list route-map** *map-name* {**in** | **out**}	Filters incoming or outgoing Network Summary (type 3) LSAs on an ABR.
Step 4	**show ip ospf policy statistics area** *id* **filter-list** {**in** | **out**}	(Optional) Displays OSPF policy information.
Step 5	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to configure a filter list in area 0.0.0.10.

```
router# configure terminal
router(config)# router ospf 201
router(config-router)# area 0.0.0.10 filter-list route-map FilterLSAs in
router(config-router)# copy running-config startup-config
```

To terminate an OSPF routing process, use the **no router ospf** *instance-tag* command.

Configuring Stub Areas

You can configure a stub area for part of an OSPFv2 domain where external traffic is not necessary. Stub areas block AS External (type 5) LSAs and limit unnecessary routing to and from selected networks. (see Stub Area, page 5-8. You can optionally block all summary routes from going into the stub area.

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see Enabling OSPFv2, page 5-13).

Ensure that there are no virtual links or ASBRs in the proposed stub area.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	**area** *area-id* **stub**	Creates this area as a stub area.
Step 4	**area** *area-id* **default-cost** *cost*	(Optional) Sets the cost metric for the default summary route sent into this stub area. The range is from 0 to 16777215. The default is 1.
Step 5	**show ip ospf**	(Optional) Displays OSPF information.
Step 6	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to create a stub area within an OSPFv2 area.

```
router# configure terminal
router(config)# router ospf 201
router(config-router)# area 0.0.0.10 stub
router(config-router)# copy running-config startup-config
```

To terminate an OSPF routing process, use the **no router ospf** *instance-tag* command.

Configuring a Totally Stubby Area

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

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

Command	Purpose
area *area-id* **stub no-summary**	Creates this area as a totally stubby area.

Configuring NSSA

You can configure an NSSA for part of an OSPFv2 domain where limited external traffic is required. For information about NSSAs, see Not-So-Stubby Area, page 5-9. You can optionally translate this external traffic to an AS External (type 5) LSA and flood the OSPFv2 domain with this routing information. An NSSA can be configured with the following optional parameters:

- No redistribution—Redistributed routes bypass the NSSA and are redistributed to other areas in the OSPFv2 autonomous system. Use this option when the NSSA ASBR is also an ABR.
- Default information originate—Generates an NSSA External (type 7) LSA for a default route to the external autonomous system. Use this option on an NSSA ASBR if the ASBR contains the default route in the routing table. This option can be used on an NSSA ABR whether or not the ABR contains the default route in the routing table.
- Route map—Filters the external routes to limit those routes that the router floods throughout NSSA and other areas.
- Translate—Translates NSSA External LSAs to AS External LSAs for areas outside the NSSA. Use this command on an NSSA ABR to flood the redistributed routes throughout the OSPFv2 autonomous system. You can optionally suppress the forwarding address in these AS External LSAs. If you choose this option, the forwarding address is set to 0.0.0.0.
- No summary—Blocks all summary routes from flooding the NSSA. Use this option on the NSSA ABR.

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see Enabling OSPFv2, page 5-13).

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

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	**area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate** [**route-map** *map-name*]] [**no-summary**] [**translate type7** {**always** | **never**} [**suppress-fa**]]	Creates this area as an NSSA.

	Command	Purpose
Step 4	**area** *area-id* **default-cost** *cost*	(Optional) Sets the cost metric for the default summary route sent into this NSSA.
Step 5	**show ip ospf**	(Optional) Displays OSPF information.
Step 6	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

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

```
router# configure terminal
router(config)# router ospf 201
router(config-router)# area 0.0.0.10 nssa no-summary
router(config-router)# copy running-config startup-config
```

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

```
router# configure terminal
router(config)# router ospf 201
router(config-router)# area 0.0.0.10 nssa default-info-originate
router(config-router)# copy running-config startup-config
```

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

```
router# configure terminal
router(config)# router ospf 201
router(config-router)# area 0.0.0.10 nssa route-map ExternalFilter no-summary
router(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA that always translates NSSA External (type 5) LSAs to AS External (type 7) LSAs.

```
router# configure terminal
router(config)# router ospf 201
router(config-router)# area 0.0.0.10 nssa translate type 7 always
router(config-router)# copy running-config startup-config
```

To terminate an OSPF routing process, use the **no router ospf** *instance-tag* command.

Configuring Virtual Links

A virtual link connects an isolated area to the backbone area through an intermediate area. (see Virtual Links, page 5-9). You can configure the following optional parameters for a virtual link:

- Authentication—Sets a simple password or MD5 message digest authentication and associated keys.
- Dead interval—Sets the time that a neighbor waits for a Hello packet before declaring the local router as dead and tearing down adjacencies.
- Hello interval—Sets the time between successive Hello packets.
- Retransmit interval—Sets the estimated time between successive LSAs.
- Transmit delay—Sets the estimated time to transmit an LSA to a neighbor.

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

Note You cannot add a virtual link to a stub area.

BEFORE YOU BEGIN

Ensure that you have enabled OSPF (see Enabling OSPFv2, page 5-13).

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	**area** *area-id* **virtual-link** *router-id*	Creates one end of a virtual link on a local router that will connect to a remote router. Be sure to create a virtual link on the remote router to complete the link.
Step 4	**show ip ospf virtual-link** [**brief**]	(Optional) Displays OSPF virtual link information.
Step 5	**copy running-config startup-config**	(Optional) Saves this configuration change.

OPTIONAL COMMANDS

You can configure the following optional commands in the virtual link configuration mode.

Command	Purpose
authentication [message-digest | null]	(Optional) Overrides area-based authentication for this virtual link.
authentication-key [**0** | **3**] *key*	(Optional) Configures a simple password for this virtual link. Use this command when the authentication is not set to either key-chain or message-digest. Entering the value of zero (0) configures the password in clear text. Entering the value of 3 configures the password as 3DES encrypted.
dead-interval *seconds*	(Optional) Configures the OSPFv2 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.
hello-interval *seconds*	(Optional) Configures the OSPFv2 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.
message-digest-key *key-id* **md5** [**0** | **3**] *key*	(Optional) Configures message digest authentication for this virtual link. Use this command if the authentication is set to message-digest. Entering the value of zero (0) configures the password in cleartext. Entering the value of 3 configures the pass key as 3DES encrypted.

Command	Purpose
retransmit-interval *seconds*	(Optional) Configures the OSPFv2 retransmit interval, in seconds. The range is from 1 to 65535. The default is 5.
transmit-delay *seconds*	(Optional) Configures the OSPFv2 transmit-delay, in seconds. The range is from 1 to 450. The default is 1.

EXAMPLE

This example shows how to create a simple virtual link between two ABRs.

The configuration for ABR 1 (router ID 27.0.0.55) is as follows.

```
router# configure terminal
router(config)# router ospf 201
router(config-router)# area 0.0.0.10 virtual-link 10.1.2.3
router(config-router-vlink)# copy running-config startup-config
```

The configuration for ABR 2 (Router ID 10.1.2.3) is as follows.

```
router# configure terminal
router(config)# router ospf 101
router(config-router)# area 0.0.0.10 virtual-link 27.0.0.55
router(config-router-vlink)# copy running-config startup-config
```

Configuring Route Summarization

You can configure route summarization for inter-area routes by configuring an address range that is summarized. For more information, see Route Summarization, page 5-10.

BEFORE YOU BEGIN

Ensure that you have enabled OSPF (see Enabling OSPFv2, page 5-13).

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	**area** *area-id* **range** *ip-prefix/length* [**no-advertise**]	Creates a summary address on an ABR for a range of addresses and optionally does not advertise this summary address in a Network Summary (type 3) LSA.
	summary-address *ip-prefix/length* [**no-advertise** | **tag** *tag-id*]	Creates a summary address on an ASBR for a range of addresses and optionally assigns a tag for this summary address that can be used for redistribution with route maps.

	Command	Purpose
Step 4	**show ip ospf summary-address**	(Optional) Displays information about OSPF summary addresses.
Step 5	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

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

```
router# configure terminal
router(config)# router ospf 201
router(config-router)# area 0.0.0.10 range 10.3.0.0/16
router(config-router)# copy running-config startup-config
```

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

```
router# configure terminal
router(config)# router ospf 201
router(config-router)# summary-address 10.5.0.0/16
router(config-router)# copy running-config startup-config
```

Configuring Stub Route Advertisements

Use stub route advertisements when you want to limit the OSPFv2 traffic through this router for a short time. For more information, see OSPFv2 Stub Router Advertisements, page 5-11.

Note When you configure the router for a graceful shutdown, do not save the running configuration because the router continues to advertise a maximum metric after it reloads.

BEFORE YOU BEGIN

Ensure that you have enabled OSPF on the router (see Enabling OSPFv2, page 5-13).

DETAILED STEPS

]

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	**max-metric router-lsa** [**on-startup** [*seconds*]	Configures OSPFv2 stub route to advertise a maximum metric (in seconds) so that other routers do not prefer the router as an intermediate hop in their shortest path first (SPF) calculations. The configurable range is 5 to 86400, the default value is 600. By entering the *on-startup* option, the router advertises a maximum metric at system startup only.
Step 4	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to enable the stub router advertisements and advertise a maximum metric of 750 seconds at system startup only.

```
router# configure terminal
router(config)# router ospf 201
router(config-router)# max-metric router-lsa on-startup 750
router(config-router)# copy running-config startup-config
```

Modifying the Default Timers

OSPFv2 includes a number of timers that control the behavior of protocol messages and shortest path first (SPF) calculations. OSPFv2 includes the following optional timer parameters:

- LSA arrival time—Sets the minimum interval allowed between LSAs that arrive from a neighbor. LSAs that arrive faster than this time are dropped.
- Pacing LSAs—Sets the interval at which LSAs are collected into a group and refreshed, checksummed, or aged. This timer controls how frequently LSA updates occur and optimizes how many are sent in an LSA update message (see Flooding and LSA Group Pacing, page 5-6).
- Throttle LSAs—Sets the rate limits for generating LSAs. This timer controls how frequently LSAs are generated after a topology change occurs.
- Throttle SPF calculation—Controls how frequently the SPF calculation is run.

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

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

See Configuring Networks in OSPFv2, page 5-15 for information about the hello interval and dead timer.

BEFORE YOU BEGIN

Ensure that you have enabled OSPFv2 (see Enabling OSPFv2, page 5-13).

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag and enters the router configuration mode.
Step 3	**timers lsa-arrival** *milliseconds*	Sets the minimum interval in which the software accepts the same link-state advertisement (LSA) from Open Shortest Path First version 2 (OSPFv2) neighbors in milliseconds. The range is from 10 to 600000. The default is 1000 milliseconds.

	Command	Purpose
Step 4	**timers lsa-group-pacing** *seconds*	Sets the interval (in seconds) at which the router collects OSPFv2 LSAs into a group and refreshes, checksums or ages them. The range is from 1 to 1800. The default is 240 seconds.
Step 5	**timers throttle lsa** *start-time hold-interval max-time*	Sets the rate limit in milliseconds for generating LSAs with the following timers: *start-time*—The range is from 50 to 5000 milliseconds. The default value is 50 milliseconds. *hold-interva*l—The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds. *max-time*—The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds.
Step 6	**timers throttle spf** *delay-time hold-time max-wait*	Sets the SPF best path schedule initial delay time and the minimum hold time in seconds between SPF best path calculations. The range is from 1 to 600000. The default is no delay time and a 5000-millisecond hold time.
Step 7	**interface** *type slot/port*	Enters interface configuration mode.
Step 8	**ip ospf hello-interval** *seconds*	Sets the hello interval for this interface. The range is from 1 to 65535. The default is 10.
Step 9	**ip ospf dead-interval** *seconds*	Sets the dead interval for this interface. The range is from 1 to 65535.
Step 10	**ip ospf retransmit-interval** *seconds*	Sets the estimated time in seconds between LSAs transmitted from this interface. The range is from 1 to 65535. The default is 5.
Step 11	**ip ospf transmit-delay** *seconds*	Sets the estimated time in seconds to transmit an LSA to a neighbor. The range is from 1 to 450. The default is 1.
Step 12	**show ip ospf**	(Optional) Displays information about OSPFv2.
Step 13	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to modify system defaults to control LSA flooding with the lsa-group-pacing and timers throttle lsa parameters.

```
router# configure terminal
router(config)# router ospf 201
router(config-router)# timers lsa-arrival 300
router(config-router)# timers lsa-group-pacing 2000
router(config-router)# timers throttle lsa 3000
router(config-router)# copy running-config startup-config
```

Configuring Graceful Restart

Graceful restart is enabled by default on the router. You can configure the following optional parameters for graceful restart in an OSPFv2 instance:

- Grace period—Configures how long neighbors must wait after a graceful restart before tearing down adjacencies.
- Helper mode disabled—Disables helper mode on the local OSPFv2 instance. OSPFv2 does not participate in the graceful restart of a neighbor.
- Planned graceful restart only—Configures OSPFv2 to support graceful restart only in the event of a planned restart.

BEFORE YOU BEGIN

Ensure that you have enabled OSPFv2 (see Enabling OSPFv2, page 5-13).

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

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	**graceful-restart**	Enables graceful restart after it has been disabled. Graceful restart is enabled by default on the router.
Step 4	**graceful-restart grace-period** *seconds*	(Optional) Sets the grace period, in seconds. The range is from 5 to 1800. The default is 60 seconds.
Step 5	**graceful-restart helper-disable**	(Optional) Disables helper mode. This feature is enabled by default.
Step 6	**graceful-restart planned-only**	(Optional) Configures a graceful restart for planned (controlled) restarts only.
Step 7	**show ip ospf**	(Optional) Displays OSPFv2 information.
Step 8	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to re-enable graceful restart on the router (after it was disabled on the router) and then set the grace period to 120 seconds.

```
router# configure terminal
router(config)# router ospf 201
router(config-router)# graceful-restart
router(config-router)# graceful-restart grace-period 120
router(config-router)# copy running-config startup-config
```

To disable graceful restart on the router, enter the **no graceful-restart** command in the router configuration mode.

Restarting an OSPFv2 Instance

You can restart an OSPFv2 instance. This action clears all neighbors for the instance.

To restart an OSPFv2 instance and remove all associated neighbors, use the following command.

Command	Purpose
restart ospf *instance-tag*	Restarts the OSPFv2 instance and removes all neighbors.

Verifying the OSPFv2 Configuration

To display the OSPFv2 configuration, enter any or all of the following commands.

Note The Cisco CG-OS software does not support the [**vrf** {*vrf-name* | **all** | **default** | **management**}] parameter in the commands listed below.

Command	Purpose
show ip ospf	Displays the OSPFv2 configuration.
show ip ospf border-routers	Displays the OSPFv2 border router configuration.
show ip ospf database	Displays the OSPFv2 link-state database summary.
show ip ospf interface *number*	Displays the OSPFv2 interface configuration.
show ip ospf lsa-content-changed-list *neighbor-id interface-type number*	Displays the OSPFv2 LSAs that have changed.
show ip ospf neighbors [*neighbor-id*] [**detail**] [*interface-type number*]	Displays the list of OSPFv2 neighbors.
show ip ospf request-list *neighbor-id interface-type number*	Displays the list of OSPFv2 link-state requests.
show ip ospf retransmission-list *neighbor-id interface-type number*	Displays the list of OSPFv2 link-state retransmissions.
show ip ospf route [*ospf-route*] [**summary**]	Displays the internal OSPFv2 routes.
show ip ospf summary-address	Displays information about the OSPFv2 summary addresses.
show ip ospf virtual-links [brief]	Displays information about OSPFv2 virtual links.
show running-configuration ospf	Displays the current running OSPFv2 configuration.

Monitoring OSPFv2 Statistics

To display OSPFv2 statistics, enter the following commands.

Command	Purpose
show ip ospf policy statistics area *area-id* **filter-list** {**in** | **out**}	Displays the OSPFv2 route policy statistics for an area.
show ip ospf policy statistics redistribute {**direct** | **ospf** *id* | **static**}	Displays the OSPFv2 route policy statistics.
show ip ospf statistics	Displays the OSPFv2 event counters.
show ip ospf traffic [*interface-type number*]	Displays the OSPFv2 packet counters.

Configuration Example for OSPFv2

The following example shows how to configure OSPFv2.

```
feature ospf
router ospf 201
 router-id 290.0.2.1

interface ethernet 2/1
 ip router ospf 201 area 0.0.0.10
 ip ospf authentication
 ip ospf authentication-key 0 mypass
```

CHAPTER 6

Configuring OSPFv3

This chapter describes how to configure Open Shortest Path First version 3 (OSPFv3) for IPv6 networks on the Cisco 1000 Series Connected Grid Routers (*hereafter* referred to as the Cisco CG-OS router). The system software for the router is identified as the Cisco CG-OS software.

This chapter includes the following sections:

Information About OSPFv3

OSPFv3 is an IETF link-state protocol. An OSPFv3 router sends a special message, called a hello packet, out each OSPF-enabled interface to discover other OSPFv3 neighbor routers. Once a neighbor is discovered, the two routers compare information in the Hello packet to determine if the routers have compatible configurations. The neighbor routers attempt to establish adjacency, which means that the routers synchronize their link-state databases to ensure that they have identical OSPFv3 routing information. Adjacent routers share link-state advertisements (LSAs) that include information about the operational state of each link, the cost of the link, and any other neighbor information. The routers then flood these received LSAs out every OSPF-enabled interface so that all OSPFv3 routers eventually have identical link-state databases. When all OSPFv3 routers have identical link-state databases, the network is converged (see Convergence, page 1-6). Each router then uses Dijkstra's Shortest Path First (SPF) algorithm to build its route table.

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

Note OSPFv3 only supports IPv6 on the router. IPv4 is not supported.

For information about OSPFv2 for IPv4, see Configuring OSPFv2.

This section includes the following topics:

- Comparison of OSPFv3 and OSPFv2, page 6-2
- Hello Packet, page 6-2
- Neighbors, page 6-3
- Adjacency, page 6-3
- Designated Routers, page 6-4
- Areas, page 6-5
- Link-State Advertisement, page 6-5
- OSPFv3 and the IPv6 Unicast RIB, page 6-7
- Address Family Support, page 6-8
- Advanced Features, page 6-8

Comparison of OSPFv3 and OSPFv2

Much of the OSPFv3 protocol is the same as in OSPFv2. OSPFv3 is described in RFC 5340.

The key differences between the OSPFv3 and OSPFv2 protocols are as follows:

- OSPFv3 expands on OSPFv2 to provide support for IPv6 routing prefixes and the larger size of IPv6 addresses.
- LSAs in OSPFv3 are expressed as prefix and prefix length instead of address and mask.
- The router ID and area ID are 32-bit numbers with no relationship to IPv6 addresses.
- OSPFv3 uses link-local IPv6 addresses for neighbor discovery and other features.
- OSPFv3 uses IPv6 IPSec for authentication.
- OSPFv3 redefines LSA types.

Hello Packet

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

- Neighbor discovery
- Keepalives
- Bidirectional communications
- Designated router election (see Designated Routers, page 6-4)

The Hello packet contains information about the originating OSPFv3 interface and router, including the assigned OSPFv3 cost of the link, the hello interval, and optional capabilities of the originating router. An OSPFv3 interface that receives these Hello packets determines if the settings are compatible with the receiving interface settings. Compatible interfaces are considered neighbors and are added to the neighbor table (see Neighbors, page 6-3).

Hello packets also include a list of router IDs for the routers that communicate with the originating interface. When the receiving interface sees its own router ID in this list, that state confirms that bidirectional communication between the two interfaces exists.

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

Neighbors

An OSPFv3 interface must have a compatible configuration with a remote interface before the two are considered neighbors. The two OSPFv3 interfaces must match the following criteria:

- Hello interval
- Dead interval
- Area ID (see Areas, page 6-5)
- Authentication
- Optional capabilities

When there is a match, the router enters the following information into the neighbor table:

- Neighbor ID—The router ID of the neighbor router.
- Priority—Priority of the neighbor router. The priority is used for designated router election (see Designated Routers, page 6-4).
- State—Indication of whether the neighbor has just been heard from, is in the process of setting up bidirectional communications, is sharing the link-state information, or has achieved full adjacency.
- Dead time—Indication of how long since the last Hello packet was received from this neighbor.
- Link-local IPv6 Address—The link-local IPv6 address of the neighbor.
- Designated Router—Indication of whether the neighbor has been declared the designated router or backup designated router (see Designated Routers, page 6-4).
- Local interface—The local interface that received the Hello packet for this neighbor.

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

Adjacency

Adjacency is the path from each router to its local designated router (DR). Not all neighbors establish adjacency. Depending on the network type and designated router establishment, some neighbors become fully adjacent and share LSAs with all their neighbors, while other neighbors do not. For more information, see Designated Routers, page 6-4.

Adjacency is established using Database Description packets, Link State Request packets, and Link State Update packets in OSPFv3. The Database Description packet includes the LSA headers from the link-state database of the neighbor (see Link-State Database, page 6-7). The local router compares these headers with its own link-state database and determines which LSAs are new or updated. The local router sends a Link State Request packet for each LSA for which it needs new or updated information. The neighbor responds with a Link State Update packet. This exchange continues until both routers have the same link-state information.

Designated Routers

Networks with multiple routers present a unique situation for OSPFv3. When every router floods the network with LSAs, multiple sources send the same link-state information. Depending on the type of network, OSPFv3 might use a single router, the designated router (DR), to control the LSA floods and represent the network to the rest of the OSPFv3 area (see Areas, page 6-5). When the DR fails, OSPFv3 selects a backup designated router (BDR).

Network types are as follows:

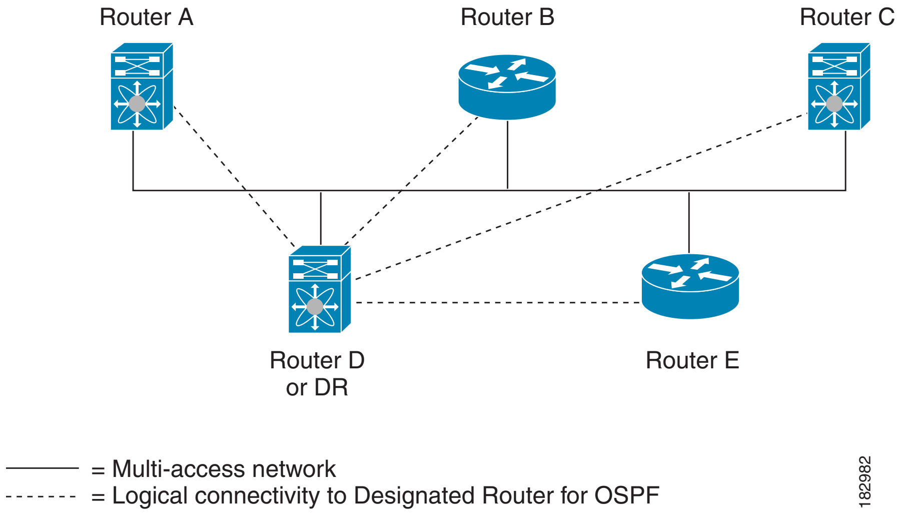
- Point-to-point—A network that exists only between two routers. All neighbors on a point-to-point network establish adjacency and there is no DR.
- Broadcast—A network with multiple routers that can communicate over a shared medium that allows broadcast traffic, such as Ethernet. OSPFv3 routers establish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

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

All other routers establish adjacency with the DR and the BDR and use the IPv6 multicast address FF02::6 to send LSA updates to the DR and BDR. Figure 6-1 shows this adjacency relationship between all routers and the DR.

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

Figure 6-1 Designated Router in Multi-Access Network



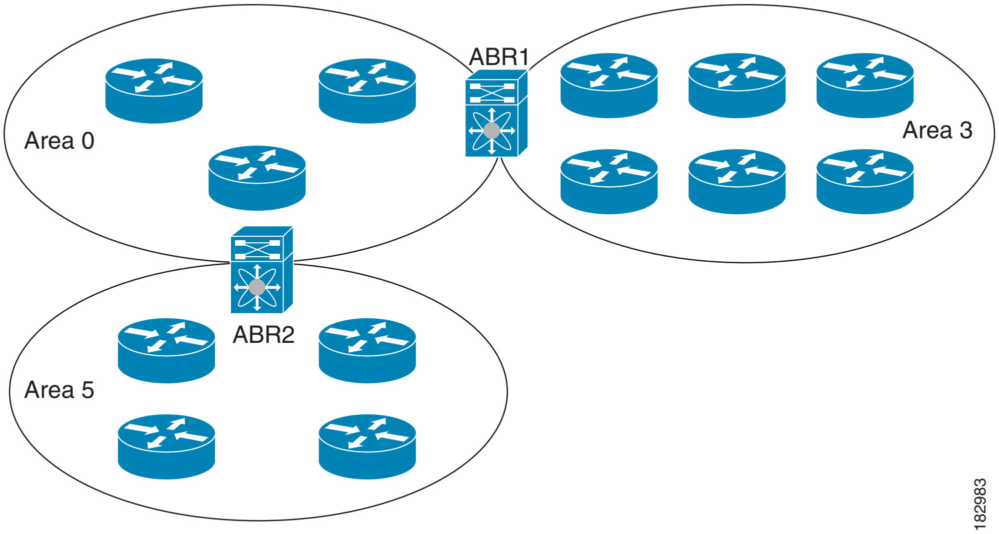
Areas

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

The Cisco CG-OS software always displays the area in dotted decimal notation.

When you define more than one area in an OSPFv3 network, you must also define the backbone area. Backbone areas have a reserved area ID of zero (0). If you have more than one area, then one or more routers become area border routers (ABRs). An ABR connects to both the backbone area and at least one other defined area (see Figure 6-2).

Figure 6-2 OSPFv3 Areas



The ABR has a separate link-state database for each area to which it connects. The ABR sends Inter-Area Prefix (type 3) LSAs (see Route Summarization, page 6-10) from one connected area to the backbone area. The backbone area sends summarized information about one area to another area. In Figure 6-2, Area 0 sends summarized information about Area 5 to Area 3.

OSPFv3 defines an additional router type: the autonomous system boundary router (ASBR). This router connects an OSPFv3 area to another autonomous system.

Link-State Advertisement

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

This section includes the following topics:

- LSA Types, page 6-6
- Link Cost, page 6-6

- Flooding and LSA Group Pacing, page 6-6
- Link-State Database, page 6-7

LSA Types

Table 6-1 shows the LSA types supported by the Cisco CG-OS software.

Table 6-1 LSA Types

Type	Name	Description
1	Router LSA	LSA sent by every router. This LSA includes the state and cost of all links but does not include prefix information. Router LSAs trigger an SPF recalculation. Router LSAs are flooded to the local OSPFv3 area.
2	Network LSA	LSA sent by the DR. This LSA lists all routers in the multi-access network but does not include prefix information. Network LSAs trigger an SPF recalculation. See Designated Routers, page 6-4.
3	Inter-Area Prefix LSA	LSA sent by the area border router to an external area for each destination in local area. This LSA includes the link cost from the border router to the local destination. See Areas, page 6-5.
4	Inter-Area Router LSA	LSA sent by the area border router to an external area. This LSA advertises the link cost to the ASBR only. See Areas, page 6-5.
5	AS External LSA	LSA generated by the ASBR. This LSA includes the link cost to an external autonomous system destination. AS External LSAs are flooded throughout the autonomous system. See Areas, page 6-5.
7	Type-7 LSA	LSA generated by the ASBR within an NSSA. This LSA includes the link cost to an external autonomous system destination. Type-7 LSAs are flooded only within the local NSSA. See Areas, page 6-5.
8	Link LSA	LSA sent by every router, using a link-local flooding scope (see Flooding and LSA Group Pacing, page 6-6. This LSA includes the link-local address and IPv6 prefixes for this link.
9	Intra-Area Prefix LSA	LSA sent by every router. This LSA includes any prefix or link state changes. Intra-Area Prefix LSAs are flooded to the local OSPFv3 area. This LSA does not trigger an SPF recalculation.
11	Grace LSAs	LSA sent by a restarting router, using a link-local flooding scope. This LSA is used for a graceful restart of OSPFv3. See Graceful Restart, page 6-10.

Link Cost

Each OSPFv3 interface has a link cost. The cost is an arbitrary number. By default, the Cisco CG-OS software assigns a cost that is the configured reference bandwidth divided by the interface bandwidth. By default, the reference bandwidth is 40 Gb/s. The link cost is carried in the LSA updates for each link.

Flooding and LSA Group Pacing

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

- Link-local—LSA is flooded only on the local link. Used for Link LSAs and Grace LSAs.

- Area-local—LSA is flooded throughout a single OSPFv3 area only. Used for Router LSAs, Network LSAs, Inter-Area-Prefix LSAs, Inter-Area-Router LSAs, and Intra-Area-Prefix LSAs.
- AS scope—LSA is flooded throughout the routing domain. An AS scope is used for AS External LSAs.

LSA flooding guarantees that all routers in the network have identical routing information. LSA flooding depends on the OSPFv3 area configuration (see Areas, page 6-5). The LSAs are flooded based on the link-state refresh time (every 30 minutes by default). Each LSA has its own link-state refresh time.

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

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

Link-State Database

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

The router removes LSAs from the link-state database when it does not receive an LSA update within a set interval (MaxAge). Routers flood a repeat of the LSA every 30 minutes to prevent aging out of accurate link-state information. The Cisco CG-OS software supports the LSA grouping feature to prevent all LSAs from refreshing at the same time. For more information, see Flooding and LSA Group Pacing, page 6-6.

Multi-Area Adjacency

OSPFv3 multi-area adjacency allows you to configure a link on the primary interface that is in more than one area. This link becomes the preferred intra-area link in those areas. Multi-area adjacency establishes a point-to-point unnumbered link in an OSPFv3 area that provides a topological path for that area. The primary adjacency uses the link to advertise an unnumbered point-to-point link in the Router LSA for the corresponding area when the neighbor state is full.

The multi-area interface exists as a logical construct over an existing primary interface for OSPFv3; however, the neighbor state on the primary interface is independent of the multi-area interface. The multi-area interface establishes a neighbor relationship with the corresponding multi-area interface on the neighboring router. See Configuring Multi-Area Adjacency, page 6-20 for more information.

OSPFv3 and the IPv6 Unicast RIB

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

- Add or remove routes

- Provide convergence updates to remove stale OSPFv3 routes and for stub router advertisements (see Multiple OSPFv3 Instances, page 6-11)

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

Address Family Support

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

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

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

Advanced Features

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

This section includes the following topics:

- Stub Area, page 6-8
- Not-So-Stubby Area, page 6-9
- Virtual Links, page 6-9
- Route Summarization, page 6-10
- Graceful Restart, page 6-10
- Multiple OSPFv3 Instances, page 6-11
- SPF Optimization, page 6-11

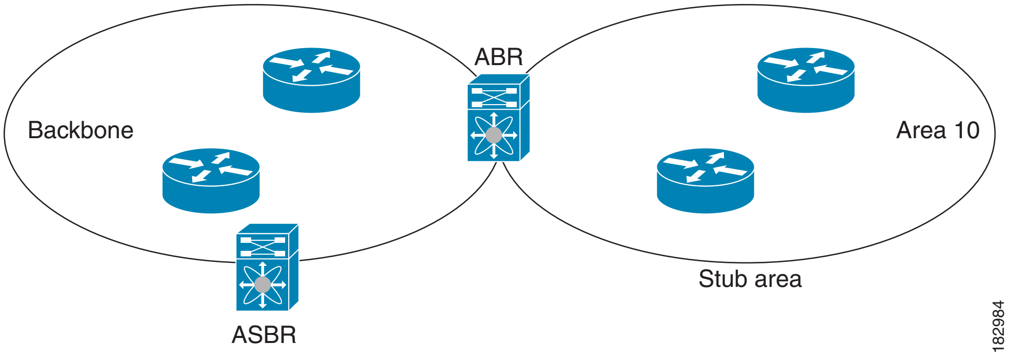
Stub Area

You can limit the amount of external routing information that floods an area by making it a stub area. A stub area is an area that does not allow AS External (type 5) LSAs (see Link-State Advertisement, page 6-5). These LSAs usually flood the local autonomous system to propagate external route information. Stub areas have the following requirements:

- All routers in the stub area must be stub routers. See Stub Routing, page 1-6.
- No ASBR routers exist in the stub area.
- You cannot configure virtual links in the stub area.

Figure 6-3 shows an example of an OSPFv3 autonomous system where all routers in area 0.0.0.10 have to go through the ABR to reach external autonomous systems. Area 0.0.0.10 can be configured as a stub area.

Figure 6-3 Stub Area



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

Not-So-Stubby Area

A Not-So-Stubby Area (NSSA) is similar to the stub area, except that an NSSA allows you to import autonomous system external routes within an NSSA using redistribution. The NSSA ASBR redistributes these routes and generates type-7 LSAs that it floods throughout the NSSA. You can optionally configure the ABR that connects the NSSA to other areas to translate this type-7 LSA to AS External (type 5) LSAs. The ABR then floods these AS External LSAs throughout the OSPFv3 autonomous system. The router supports summarization and filtering during the translation. See Link-State Advertisement, page 6-5 for details on type-7 LSAs.

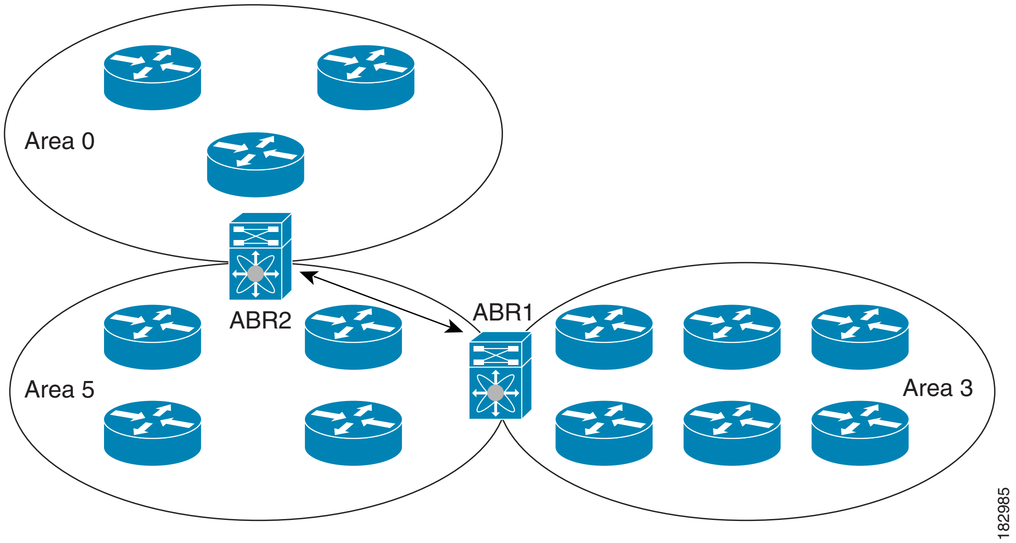
You can, for example, use NSSA to simplify administration when you are connecting a central site using OSPFv3 to a remote site that is using a different routing protocol. Before NSSA, the connection between the corporate site border router and a remote router could not be run as an OSPFv3 stub area because routes for the remote site could not be redistributed into a stub area. With NSSA, you can extend OSPFv3 to cover the remote connection by defining the area between the corporate site router and remote router as an NSSA (see Configuring NSSA, page 6-19).

The backbone area 0 cannot be an NSSA.

Virtual Links

Virtual links allow you to connect an OSPFv3 area ABR to a backbone area ABR when a direct physical connection is not available. Figure 6-4 shows a virtual link that connects Area 3 to the backbone area through Area 5.

Figure 6-4 Virtual Links



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

Route Summarization

Because OSPFv3 shares all learned routes with every OSPF-enabled router, you might want to use route summarization to reduce the number of unique routes that are flooded to every OSPF-enabled router. Route summarization simplifies route tables by replacing more-specific addresses with an address that represents all the specific addresses. For example, you can replace 2010:11:22:0:1000::1 and 2010:11:22:0:2000:679:1 with one summary address, 2010:11:22::/48.

Typically, you would summarize at the boundaries of area border routers (ABRs). Although you could configure summarization between any two areas, it is better to summarize in the direction of the backbone so that the backbone receives all the aggregate addresses and injects them, already summarized, into other areas.

Inter-area route summarization

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

Graceful Restart

OSPFv3 automatically restarts if the process experiences problems. After the restart, OSPFv3 initiates a graceful restart so that the platform is not taken out of the network topology. When you manually restart OSPFv3, it performs a graceful restart, which is similar to a stateful switchover. The Cisco CG-OS software applies the running configuration in both cases.

A graceful restart, also known as nonstop forwarding (NSF), allows OSPFv3 to remain in the data forwarding path through a process restart. When OSPFv3 needs to restart, it first sends a link-local Grace (type 11) LSA.

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

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

Note When the restarting OSPFv3 interface does not come back up before the end of the grace period, or if the network experiences a topology change, then the OSPFv3 neighbors tear down adjacency with the restarting OSPFv3 and treat it as a normal OSPFv3 restart.

Multiple OSPFv3 Instances

The Cisco CG-OS software supports multiple instances of the OSPFv3 protocol. By default, every instance uses the same system router ID. You must manually configure the router ID for each instance if the instances are in the same OSPFv3 autonomous system.

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

The Cisco CG-OS software allows only one OSPFv3 instance on an interface.

SPF Optimization

The Cisco CG-OS software optimizes the SPF algorithm in the following ways:

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

Prerequisites for OSPFv3

You must be familiar with routing fundamentals to configure OSPFv3.

You have completed the OSPFv3 network strategy and planning for your network. For example, you must decide whether to implement multiple areas.

You are familiar with IPv6 addressing and basic configuration. See Chapter 3, “Configuring IPv6,” for information on IPv6 routing and addressing.

You must be logged on to the router.

You have enabled OSPFv3 (see Enabling OSPFv3, page 6-12).

You have configured at least one interface for IPv6 that is capable of communicating with a remote OSPFv3 neighbor.

Guidelines and Limitations for OSPFv3

The Cisco CG-OS software displays areas in dotted decimal notation regardless of whether you enter the area in decimal or dotted decimal notation.

Default Settings

Table 6-2 lists the default settings for OSPFv3 parameters.

Table 6-2 Default OSPFv3 Parameters

Parameter	**Default**
Hello interval	10 seconds
Dead interval	40 seconds
Graceful restart grace period	60 seconds
Graceful restart notify period	15 seconds
OSPFv3 feature	Disabled
Stub router advertisement announce time	600 seconds
Reference bandwidth for link cost calculation	40 Gb/s
LSA minimal arrival time	1000 milliseconds
LSA group pacing	240 seconds
SPF calculation initial delay time	0 milliseconds
SPF calculation hold time	5000 milliseconds
SPF calculation initial delay time	0 milliseconds

Configuring Basic OSPFv3

Be sure to review the Prerequisites for OSPFv3 and Guidelines and Limitations for OSPFv3 sections before configuring OSPFv3.

This section includes the following topics:

- Enabling OSPFv3, page 6-12
- Creating an OSPFv3 Instance, page 6-13
- Configuring Networks in OSPFv3, page 6-14

Enabling OSPFv3

You must enable OSPFv3 before you can configure OSPFv3.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**feature ospfv3**	Enables OSPFv3.
Step 3	**show feature**	(Optional) Displays enabled and disabled features.
Step 4	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to enable OSPFv3 on the router.

```
router# configure terminal
router(config)#feature ospfv3
router(config)# copy running-config startup-config
```

To disable the OSPFv3 feature and remove all associated configurations, use the **no feature ospfv3** command in configuration mode.

Creating an OSPFv3 Instance

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

- Router ID—Configures the router ID for this OSPFv3 instance. If you do not use this parameter, the router ID selection algorithm is used. For more information, see the Router IDs, page 1-5.
- Administrative distance—Rates the trustworthiness of a routing information source. For more information, see the Administrative Distance, page 1-6.
- Log adjacency changes—Creates a system message whenever an OSPFv3 neighbor changes its state.
- Maximum paths—Sets the maximum number of equal paths that OSPFv3 installs in the route table for a particular destination. Use this parameter for load balancing between multiple paths.
- Reference bandwidth—Controls the calculated OSPFv3 cost metric for a network. The calculated cost is the reference bandwidth divided by the interface bandwidth. You can override the calculated cost by assigning a link cost when a network is added to the OSPFv3 instance. For more information, see Configuring Networks in OSPFv3, page 6-14.

For more information about OSPFv3 instance parameters, see Configuring Advanced OSPFv3, page 6-16.

BEFORE YOU BEGIN

You must enable OSPFv3 (see Enabling OSPFv3, page 6-12).

Enter the **show ospfv3** *instance-tag* command to verify that the instance tag is not in use on this router.

OSPFv3 must be able to obtain a router identifier (for example, a configured loopback address) or you must configure the router ID option (see Step 3 below).

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	**router-id** *128-bit number*	Configures the OSPFv3 router ID. This ID is a 128-bit number and uses the dotted decimal notation. It identifies this OSPFv3 instance and must exist on a configured interface in the system.
Step 4	**log-adjacency-changes [detail]**	(Optional) Generates a system message whenever a neighbor changes state (Router config mode).
Step 5	**distance** number	(Optional) Configures the administrative distance for this OSPFv3 instance. The range is from 1 to 255. The default is 110. (Address family mode)
Step 6	**maximum-paths** paths	(Optional) Configures the maximum number of equal OSPFv3 paths to a destination in the route table. The range is from 1 to 16. The default is 8. Use this command for load balancing. (Address family mode)
Step 7	**show ipv6 ospfv3**	(Optional) Displays OSPFv3 configuration information for the router.
Step 8	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to create an OSPFv3 instance on the router.

```
router# configure terminal
router(config)# router ospfv3 201
router(config-router)# log-adjacency-changes
router(config-router)# address-family
router(config-router-af)# distance 200
router(config-router-af)# exit
router(config-router)# copy running-config startup-config
```

To remove the OSPFv3 instance and all associated configuration, use the **no router ospfv3** *instance-tag* command in configuration mode.

Note When operating in the interface mode, the **no router ospfv3** *instance-tag* command does not remove all associated OSPFv3 configurations. You must manually remove any associated OSPFv3 commands for the interface.

Configuring Networks in OSPFv3

You can configure a network to OSPFv3 by associating it through the interface that the router uses to connect to that network (see Neighbors, page 6-3). You can add all networks to the default backbone area (Area 0), or you can create new areas using any decimal number or an IP address.

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

Note When configuring OSPFv3 on an interface, you must assign a valid IPv6 address to that interface before you can enable OSPFv3 on the interface.

BEFORE YOU BEGIN

You must enable OSPFv3 on the router (see Enabling OSPFv3, page 6-12).

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	**ipv6 address** *ipv6-prefix/length*	Assigns an IPv6 address to this interface.
Step 4	**ipv6 router ospfv3** *instance-tag* **area** *area-id* [**secondaries none**]	Adds the interface to the OSPFv3 instance and area.
Step 5	**show ipv6 ospfv3 interface** *interface-type slot/port*	(Optional) Displays OSPFv3 information for the interface.
Step 6	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to add an interface to a OSPFv3 instance 201 and area a network area 0.0.0.10.

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

OPTIONAL PARAMETERS

You can configure the following optional parameters for OSPFv3 in interface configuration mode.

Command	Purpose
ospfv3 cost number	Configures the OSPFv3 cost metric for this interface. The default is to calculate a cost metric, based on the reference bandwidth and interface bandwidth. The range is from 1 to 65535.
ospfv3 dead-interval seconds	Configures the OSPFv3 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.
ospfv3 hello-interval seconds	Configures the OSPFv3 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.
ospfv3 mtu-ignore	Configures OSPFv3 to ignore any IP maximum transmission unit (MTU) mismatch with a neighbor. The default is to not establish adjacency if the neighbor MTU does not match the local interface MTU.
ospfv3 network {**broadcast** | **point-point**}	Sets the OSPFv3 network type.
ospfv3 passive-interface	Suppresses routing updates on the interface.
ospfv3 priority *number*	Configures the OSPFv3 priority, used to determine the DR for an area. The range is from 0 to 255. The default is 1. See Designated Routers, page 6-4.
ospfv3 shutdown	Shuts down the OSPFv3 instance on the interface.

Configuring Advanced OSPFv3

Configure OSPFv3 after you have designed your OSPFv3 network.

This section includes the following topics:

- Configuring Filter Lists for Border Routers, page 6-17
- Configuring Stub Areas, page 6-17
- Configuring a Totally Stubby Area, page 6-18
- Configuring NSSA, page 6-19
- Configuring Multi-Area Adjacency, page 6-20
- Configuring Virtual Links, page 6-21
- Configuring Route Summarization, page 6-23
- Modifying the Default Timers, page 6-24
- Configuring Graceful Restart, page 6-25
- Restarting an OSPFv3 Instance, page 6-26
- Monitoring OSPFv3, page 6-27

Configuring Filter Lists for Border Routers

You can separate your OSPFv3 domain into a series of areas that contain related networks. All areas must connect to the backbone area through an area border router (ABR). OSPFv3 domains can connect to external domains as well through an autonomous system border router (ASBR). See Areas, page 6-5.

ABRs have the following optional configuration parameters:

- Area range—Configures route summarization between areas. For more information, see Configuring Route Summarization, page 6-23.
- Filter list—Filters the Inter-Area Prefix (type 3) LSAs on an ABR that are allowed in from an external area.

ASBRs also support filter lists.

BEFORE YOU BEGIN

You must enable OSPFv3 (see Enabling OSPFv3, page 6-12).

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	**address-family ipv6 unicast**	Enters IPv6 unicast into address family mode.
Step 4	**area** *area-id* **filter-list route-map** *map-name* {**in** | **out**}	Filters incoming or outgoing Inter-Area Prefix (type 3) LSAs on an ABR.
Step 5	**show ipv6 ospfv3 policy statistics area** *id* **filter-list** {**in** | **out**}	(Optional) Displays OSPFv3 policy information.
Step 6	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to define a filter-list for OSPFv3 area 0.0.0.10:

```
router# configure terminal
router(config)# router ospfv3 201
router(config-router)# address-family ipv6 unicast
router(config-router-af)# area 0.0.0.10 filter-list route-map FilterLSAs in
router(config-router-af)# show ipv6 ospfv3 policy statistics area 0.0.0.10 filter-list in
router(config-router-af)# copy running-config startup-config
```

To terminate an OSPFv3 routing process, use the **no router ospfv3** *instance-tag* command in configuration mode.

Configuring Stub Areas

You can configure a stub area for part of an OSPFv3 domain where external traffic is not necessary. Stub areas block AS External (type 5) LSAs, limiting unnecessary routing to and from selected networks. See Stub Area, page 6-8. You can optionally block all summary routes from going into the stub area.

BEFORE YOU BEGIN

You must enable OSPFv3 (see Enabling OSPFv3, page 6-12).

Ensure that there are no virtual links or ASBRs in the proposed stub area.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	**area** *area-id* **stub [no-summary]**	Creates this area as a stub area. When you specify the optional **no-summary** parameter in the command, an area border router (ABR) does not forward summary link advertisements into the stub area.
Step 4	**address-family ipv6 unicast**	(Optional) Enters IPv6 unicast address family mode.
Step 5	**area** *area-id* **default-cost** *cost*	(Optional) Sets the cost metric for the default summary route sent into this stub area. The range is from 0 to 16777215.
Step 6	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

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

```
router# configure terminal
router(config)# router ospfv3 201
router(config-router)# area 0.0.0.10 stub no-summary
router(config-router)# copy running-config startup-config
```

To terminate an OSPFv3 routing process, use the **no router ospfv3** *instance-tag* command in the configuration mode.

Configuring a Totally Stubby Area

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

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

Command	Purpose
area *area-id* **stub no-summary**	Creates an area as a totally stubby area.

EXAMPLE

This example shows how to create a totally stub area that blocks all summary route updates from entering the stub area.

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

To remove an OSPFv3 stub area, use the **no area** *area-id* **stub [no-summary]** command in router mode.

Configuring NSSA

You can configure an NSSA for part of an OSPFv3 domain where limited external traffic is required. See Not-So-Stubby Area, page 6-9. You can optionally translate this external traffic to an AS External (type 5) LSA and flood the OSPFv3 domain with this routing information.

An NSSA can be configured with the following optional parameters:

- Default information originate—Generates a Type-7 LSA for a default route to the external autonomous system. Use this option on an NSSA ASBR if the ASBR contains the default route in the routing table. This option can be used on an NSSA ABR whether or not the ABR contains the default route in the routing table.
- No redistribution—Redistributes routes that bypass the NSSA to other areas in the OSPFv3 autonomous system. Use this option when the NSSA ASBR is also an ABR.
- No summary—Blocks all summary routes from flooding the NSSA. Use this option on the NSSA ABR.
- Route map—Filters the external routes to limit those routes that the router floods throughout the NSSA and other areas.
- Suppress—Suppresses the forwarding address in translated LSAs. The ABR uses 0.0.0.0 as the forwarding IPV4 address.
- Translate—Translates Type-7 LSAs to AS External (type 5) LSAs for areas outside the NSSA. Use this command on an NSSA ABR to flood the redistributed routes throughout the OSPFv3 autonomous system. You can optionally suppress the forwarding address in these AS External LSAs.

BEFORE YOU BEGIN

You must enable OSPF (see Enabling OSPFv3, page 6-12).

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

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	**area** *area-id* **nssa [no-redistribution] [default-information-originate][route-map map-name][no-summary] [translate type7 {always | never} [suppress-fa]]**	Creates this area as an NSSA.
Step 4	**address-family ipv6 unicast**	(Optional) Enters IPv6 unicast address family mode.

	Command	Purpose
Step 5	**area** *area-id* **default-cost** *cost*	(Optional) Sets the cost metric for the default summary route sent into this NSSA. The range is from 0 to 16777215.
Step 6	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

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

```
router# configure terminal
router(config)# router ospfv3 201
router(config-router)# area 0.0.0.10 nssa no-summary
router(config-router)# copy running-config startup-config
```

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

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

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

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

This example shows how to create an NSSA that always translates Type-7 LSAs to AS External (type 5) LSAs.

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

This example shows how to create an NSSA that always translates Type-7 LSAs to AS External (type 5) LSAs but does not place the Type-7 forwarding addresses into the AS External (type 5) LSAs.

```
router# configure terminal
router(config)# router ospfv3 201
router(config-router)# area 0.0.0.10 nssa translate type 7 suppress-fa
router(config-router)# copy running-config startup-config
```

To remove the NSSA, use the **no area** command in the router mode.

Configuring Multi-Area Adjacency

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

BEFORE YOU BEGIN

You must enable OSPFv3 on the router (see Enabling OSPFv3, page 6-12).

Ensure that you have configured a primary area for the interface (see Configuring Networks in OSPFv3, page 6-14.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	**ipv6 router ospfv3 multi-area** *area-id*	Adds additional areas to an OSPFv3 interface. The *area-id* can be either an IP address or a positive integer.
Step 4	**show ipv6 ospfv3 interface** *interface-type slot/port*	(Optional) Displays OSPFv3 information.
Step 5	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

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

```
router# configure terminal
router(config)# interface ethernet 2/1
router(config-if)# ipv6 router ospfv3 multi-area 20
router(config-if)# copy running-config startup-config
```

To remove the multi-area adjacency on the OSPFv3 interface, use the **no ipv6 router ospfv3 multi-area** *area-id* command.

Configuring Virtual Links

A virtual link connects an isolated area to the backbone area through an intermediate area. See Virtual Links, page 6-9. You can configure the following optional parameters for a virtual link:

- Authentication—Sets simple password or MD5 message digest authentication and associated keys.
- Dead interval—Sets the time that a neighbor waits for a Hello packet before declaring the local router as dead and tearing down adjacencies.
- Hello interval—Sets the time between successive Hello packets.
- Retransmit interval—Sets the estimated time between successive LSAs.
- Transmit delay—Sets the estimated time to transmit an LSA to a neighbor.

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

BEFORE YOU BEGIN

You must enable OSPFv3 (see Enabling OSPFv3, page 6-12).

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	**area** *area-id* **virtual-link** *router-id*	Creates one end of a virtual link to a remote router. You must also create a virtual link on the remote router to complete the link.
Step 4	**show ipv6 ospfv3 virtual-link** [**brief**]	(Optional) Displays OSPFv3 virtual link information.
Step 5	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

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

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

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

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

```
router# configure terminal
router(config)# router ospfv3 101
router(config-router)# area 0.0.0.10 virtual-link 2001:0DB8::1
router(config-router-vlink)# copy running-config startup-config
```

To remove a virtual link on the OSPFv3 interface, use the **no area** *area-id* **virtual link router** *router-id* command.

OPTIONAL COMMANDS

You can configure the following optional commands in virtual link configuration mode.

Command	Purpose
dead-interval seconds	(Optional) Configures the OSPFv3 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.
hello-interval *seconds*	(Optional) Configures the OSPFv3 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.
retransmit-interval *seconds*	(Optional) Configures the OSPFv3 retransmit interval, in seconds. The range is from 1 to 65535. The default is 5.
transmit-delay *seconds*	(Optional) Configures the OSPFv3 transmit delay, in seconds. The range is from 1 to 450. The default is 1.

Configuring Route Summarization

You can configure route summarization for inter-area routes by configuring an address range that is summarized. For more information, see Route Summarization, page 6-10.

BEFORE YOU BEGIN

You must enable OSPF (see Enabling OSPFv3, page 6-12).

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	**address-family ipv6 unicast**	Enters IPv6 unicast address family mode.
Step 4	**area** *area-id* **range** *ipv6-prefix/length* [**not-advertise**]	Creates a summary address on an ABR for a range of addresses and advertises this summary address in a Inter-Area Prefix (type 3) LSA. To not advertise the summary address, append the command with **not-advertise**.
Step 5	**summary-address** *ipv6-prefix/length* **[not-advertise][tag** *tag***]**	Creates a summary address on an ASBR for a range of addresses. Optionally, you can suppress (**not-advertise**) addresses that match the *ipv6-prefix/length*. **Note** The router does not support the [**tag** *tag*] option of this command because it does not support redistribution of routes.
Step 6	**show ipv6 ospfv3 summary-address**	(Optional) Displays information about OSPFv3 summary addresses.
Step 7	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

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

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

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

```
router# configure terminal
router(config)# router ospfv3 201
router(config-router)# address-family ipv6 unicast
router(config-router-af)# summary-address 2001:0DB8::/48
router(config-router)# copy running-config startup-config
```

To terminate an OSPFv3 routing process, use the **no router ospfv3** *instance-tag* command in the configuration mode.

Modifying the Default Timers

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

Each of the following parameters comes with a default setting which you can modify.

- OSPFv3 includes the following timer parameters at the router configuration mode:
  - Group pacing LSAs—Sets the interval at which LSAs are collected into a group and refreshed, checksummed, or aged. This timer controls how frequently LSA updates occur and optimizes how many are sent in an LSA update message (see Flooding and LSA Group Pacing, page 6-6).
  - LSA arrival time—Sets the minimum interval allowed between the same LSA arriving from an OSPFv3 neighbor. The Cisco CG-OS software drops all LSAs that arrive faster than the set minimal interval time.
  - Throttle LSAs—Sets rate limits for generating LSAs. This timer controls how frequently LSAs are generated after a topology change occurs.
  - Throttle SPF calculation—Controls how frequently the SPF calculation is run.
- OSPFv3 includes the following timer parameters at the interface configuration mode:
  - Retransmit interval—Sets the estimated time between successive LSAs.
  - Transmit delay—Sets the estimated time to transmit an LSA to a neighbor.

For information on the hello interval and dead timer, refer to Configuring Networks in OSPFv3, page 6-14

BEFORE YOU BEGIN

You must enable OSPFv3 (see Enabling OSPFv3, page 6-12).

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	**timers lsa-arrival** *milliseconds*	(Optional) Sets the minimum LSA arrival time in milliseconds. The range is from 10 to 600000. The default is 1000 milliseconds.
Step 4	**timers lsa-group-pacing** *seconds*	(Optional) Sets the interval in seconds for grouping LSAs. The range is from 1 to 1800. The default is 240 seconds.

	Command	Purpose
Step 5	**timers throttle lsa** *start-time hold-interval max-time*	(Optional) Sets the rate limit in milliseconds for generating LSAs. You can configure the following timers: *start-time*—The range is from 50 to 5000 milliseconds. The default value is 50 milliseconds. *hold-interva*l—The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds. *max-time*—The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds.
Step 6	**address-family ipv6 unicast**	Enters IPv6 unicast address family mode.
Step 7	**timers throttle spf** *delay-time hold-time*	(Optional) Sets the SPF best path schedule initial delay time and the minimum hold time in seconds between SPF best path calculations. The range is from 1 to 600000. The default is no delay time and 5000 millisecond hold time.
Step 8	**interface** *type slot/port*	Enters interface configuration mode.
Step 9	**ospfv3 retransmit-interval** *seconds*	(Optional) Sets the estimated time in seconds between LSAs transmitted from this interface. The range is from 1 to 65535. The default is 5.
Step 10	**ospfv3 transmit-delay** *seconds*	Sets the estimated time in seconds to transmit an LSA to a neighbor. The range is from 1 to 450. The default is 1.
Step 11	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

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

```
router# configure terminal
router(config)# router ospfv3 201
router(config-router)# timers lsa-group-pacing 300
router(config-router)# copy running-config startup-config
```

To return to the default settings for a timer, use the **no** option for each command as shown in the example below.

```
router(config-router)# no timers throttle lsa
```

Configuring Graceful Restart

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

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

- Planned graceful restart only—Configures OSPFv3 to support graceful restart only in the event of a planned restart.

BEFORE YOU BEGIN

You must enable OSPFv3 (See Enabling OSPFv3, page 6-12).

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

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	**graceful-restart**	Enables graceful restart after an admin disables it. A graceful restart is enabled by default.
Step 4	**graceful-restart grace-period** *seconds*	Sets the grace period, in seconds. The range is from 5 to 1800. The default is 60 seconds.
Step 5	**graceful-restart helper-disable**	Disables helper mode. Enabled by default.
Step 6	**graceful-restart planned-only**	Configures graceful restart for planned restarts only.
Step 7	**show ipv6 ospfv3**	(Optional) Displays OSPFv3 information.
Step 8	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to enable graceful restart on the router after it was disabled and then set the grace period to 120 seconds.

```
router# configure terminal
router(config)# router ospfv3 201
router(config-router)# graceful-restart
router(config-router)# graceful-restart grace-period 120
router(config-router)# copy running-config startup-config
```

Restarting an OSPFv3 Instance

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

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

Command	Purpose
restart ospfv3 *instance-tag*	Restarts the OSPFv3 instance and removes all neighbors.

Verifying the OSPFv3 Configuration

To display the OSPFv3 configuration, enter any or all of the following commands.

Command	Purpose
show ipv6 ospfv3	Displays the OSPFv3 configuration.
show ipv6 ospfv3 border-routers	Displays the internal OSPF routing table entries to an ABR and ASBR.
show ipv6 ospfv3 database	Displays lists of information related to the OSPFv3 database for a specific router.
show ipv6 ospfv3 interface *type number*	Displays the OSPFv3 interface configuration.
show ipv6 ospfv3 neighbors	Displays the neighbor information. Use the **clear ospfv3 neighbors** command to remove adjacency with all neighbors.
show ipv6 ospfv3 request-list	Displays a list of LSAs requested by a router.
show ipv6 ospfv3 retransmission-list	Displays a list of LSAs waiting to be retransmitted.
show ipv6 ospfv3 summary-address	Displays a list of all summary address redistribution information configured under an OSPFv3 instance.
show running-configuration ospfv3	Displays the current running OSPFv3 configuration.

Monitoring OSPFv3

To display OSPFv3 statistics, use the following commands.

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

Configuration Examples for OSPFv3

This example shows how to configure OSPFv3.

```
feature ospfv3
router ospfv3 201
 router-id 290.0.2.1

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

CHAPTER 7

Configuring WAN Backhaul Redundancy

This chapter describes how to configure WAN backhaul redundancy for cellular and WiMAX interfaces on the Cisco 1000 Series Connected Grid Routers (*hereafter* referred to as the Cisco CG-OS router). The system software for the router is identified as the Cisco CG-OS software.

This chapter includes the following sections:

Information About WAN Backhaul Redundancy

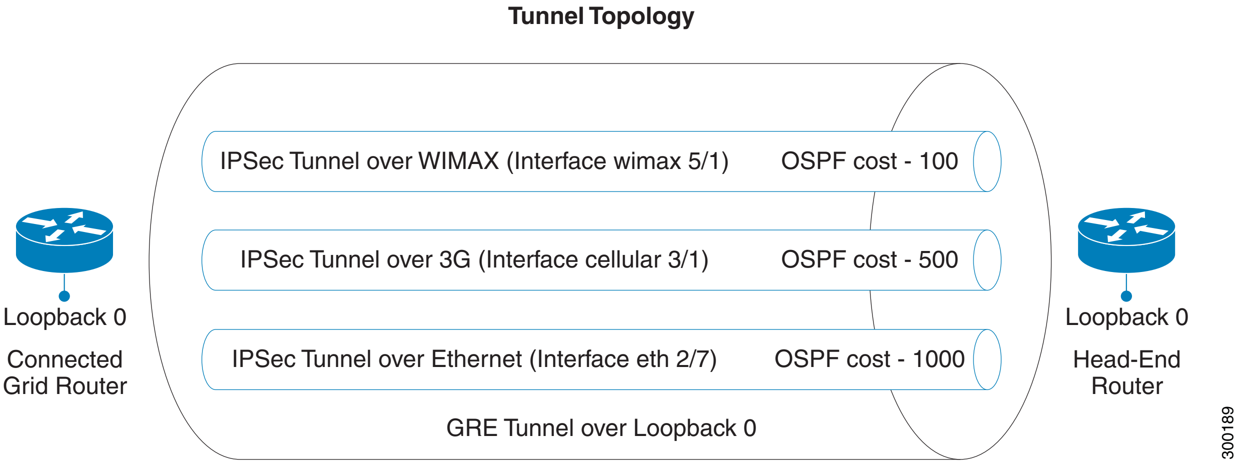
You can configure the Cisco CG-OS router to have redundant WAN backhauls within an Open Shortest Path First version 2 (OSPFv2) area by assigning link costs to cellular (3G) and WiMAX interfaces. The interface with the lower assigned link cost remains the primary link until that link goes down; and, then traffic automatically goes to the secondary link with the next lowest cost. In cases where the link with the higher cost fails, no redirect of traffic occurs because the Cisco CG-OS router by default routes all traffic to the link with the lowest cost.

IPSec virtual tunnels are configured between the Cisco CG-OS router and the head-end router (such as the Cisco ASR 1000 Series) to encapsulate the interface traffic (cellular and WiMAX). A tunnel for each of the interfaces is configured.

The router employs IKEv2 to authenticate the traffic running between the Cisco CG-OS router and the head-end router by using either pre-shared key (PSK) or RSA authentication.

A Generic Routing Encapsulation (GRE) tunnel (see RFC2784 and RFC2890) is built between the Cisco CG-OS router and the head-end router. The GRE tunnel allows IPv6 traffic within an OSPFv3 area to traverse the tunnel between the two routers without the packets being parsed or examined by either of the routers.

Figure 7-1 WAN Backhaul Redundancy Employing IPSec and GRE Tunnels


Tunnel Topology
IPSec Tunnel over WIMAX (Interface wimax 5/1)
OSPF cost - 100
IPSec Tunnel over 3G (Interface cellular 3/1)
OSPF cost - 500
IPSec Tunnel over Ethernet (Interface eth 2/7)
OSPF cost - 1000
GRE Tunnel over Loopback 0
Loopback 0
Connected Grid Router
Loopback 0
Head-End Router
300189

Prerequisites for WAN Backhaul Redundancy

When configuring for IPv4 traffic:

- Ensure that your OSPFv2 network strategy and planning for your network is complete. For example, you must decide whether your network requires multiple areas.
- You must be familiar with routing fundamentals to configure OSPFv2.
- Enable the OSPFv2 feature on the Cisco CG-OS router (see Enabling OSPFv2, page 5-13).
- Configure at least one interface for IPv4 that can communicate with a remote OSPFv2 neighbor.

When configuring for IPv6 traffic:

- Ensure that your OSPFv3 network strategy and planning for your network is complete. For example, you must decide whether your network requires multiple areas.
- You must be familiar with routing fundamentals to configure OSPFv3.
- Enable the OSPFv3 feature on the Cisco CG-OS router (see Enabling OSPFv3, page 6-12).
- Configure at least one interface for IPv6 that can communicate with a remote OSPFv3 neighbor.

Guidelines and Limitations

The Ethernet interface cannot currently be configured as a redundant WAN backhaul.

Configuring WAN Backhaul Redundancy

Tip Cisco recommends that you configure the feature on the Cisco CG-OS router in the order noted below.

Note An example configuration for the head-end router (and Cisco CG-OS router) is provided in the Configuration Example section. For details on configuring the Cisco ASR 1000 Series, refer to the following URL: http://www.cisco.com/en/US/partner/products/ps9343/index.html

BEFORE YOU BEGIN

Ensure that all of the requirements listed in the Prerequisites for WAN Backhaul Redundancy section are met before beginning to configure the Cisco CG-OS router.

DETAILED STEPS

To configure the loopback interface, follow these steps.

	Command	Purpose
Step 1	**interface loopback** *number*	Creates a path between the Cisco CG-OS router and the head-end router.
Step 2	**ip address** *ip address*	Creates an IP address for the loopback interface.
Step 3	**ip router ospf** *instance-tag* **area** *area-id* **[secondaries none]**	Creates an OSPFv2 area on the interface.

To configure the cellular interface, follow these steps.

	Command	Purpose
Step 1	**chat-script gsm1** *name*	Defines the ATDT modem commands when the dialer is initiated to automate the connect procedure.
Step 2	**interface dialer** *number*	Configures an external dialer interface.
Step 3	**dialer persistent**	Initiates a dial-out when the connection disconnects to ensure the connection remains active.
Step 4	**dialer pool** *number*	Creates a dialer pool.
Step 5	**dialer string gsm** *number*	Specifies the number or string to call. Reference the name of the chat script in this command for call setup.
Step 6	**interface cellular** *slot/port*	Configures the 3G cellular interface. Slot value range is 3 to 6. Port value is 1.
Step 7	**dialer pool-member** *number*	Assigns membership to a defined dialer pool.
Step 8	**no shutdown**	Brings up the port, administratively.
Step 9	**exit**	Exits to the global configuration mode.

To configure the WiMAX interface, follow these steps.

	Command	Purpose
Step 1	**wimax scan-list** *name*	Sets a scan-list and enters the scan list mode.
Step 2	**channel** {**index** *number* | **frequency** *number* | **bandwidth** *number*}	Configures the channel number, frequency, or bandwidth for the interface.
Step 3	**nap id** *nap-id* **priority** *value* **channel-index** *value*	Defines the Network Access Provider (NAP) ID, priority, and channel for the interface. *nap-id*–Must be in the following format: XX:XX:XX (maximum size of 64). **priority** *value*–Any value in the range of 1 to 250. **channel-index** v*alue*–Any value in the range of 0 to15.
Step 4	**nsp id** *nsp-id* **home**	Defines the Network Service Provider (NSP) ID for the interface. *nsp-id*–Must be in the following format: XX:XX:XX (maximum size of 64).
Step 5	**interface wimax** *slot/port*	Configures the WiMAX cellular interface. Slot value range is 3 to 6. Port value is 1.
Step 6	**scan-list** *name*	Attaches a scan list to the interface.
Step 7	**ip address** *ip address*	Assigns an IP address to the interface.
Step 8	**no shutdown**	Brings up the port, administratively.
Step 9	**exit**	Enter exit to return to the global configuration mode.

To enable IKEv2 on the Cisco CG-OS router, follow these steps.

	Command	Purpose
Step 1	**feature crypto ike**	Enables IKEv2 on the Cisco CG-OS router. **Note** To prevent loss of the IKEv2 configuration, do not disable IKEv2 when IPSec is enabled on the Cisco CG-OS router.
Step 2	**crypto ike domain ipsec**	Configures the IKEv2 domain and enters the IKEv2 configuration submode.
Step 3	**identity hostname**	(Optional) Configures the identity of the IKEv2 protocol. By default, the IP address of the Cisco CG-OS router is taken as the identity for IKEv2 protocol. **Note** This command is optional when using pre-share key (PSK) authentication.
Step 4	**keepalive** *value*	(Optional) Configures the frequency of keep alive messages sent between peers in the tunnel. Keep alive messages validate the ability of peers to send and receive traffic. Value can be any number between 120 and 86400 seconds. The default value is 3600 seconds.
Step 5	**policy** *value*	Defines IKEv2 priority policy and enters the policy configuration submode. The lower the number, the higher the priority.

	Command	Purpose
Step 6	**authentication** *method*	Specifies the IKEv2 authentication method. Method options are PSK (pre-share) and RSA signature (rsa-sig) authentication. **Note** RSA is the default setting.
Step 7	**encryption** *enc_algo*	(Optional) Specifies the encryption algorithm for the policy. Options are: 3des–168-bit DES (3DES) aes–AES-CBC Default setting for the Cisco CG-OS router is aes.
Step 8	**hash** *hash_algo*	(Optional) Configures the hash algorithm for the IKEv2 policy. Options are: md5–HMAC-MD5 sha–HMAC-SHA1 Default setting for the Cisco CG-OS router is sha.
Step 9	**group** *DH_group*	Configures the Diffie Hellman group for the policy. Options are: 1–768-bit Diffie Hellman group 2–1024-bit Diffie Hellman group 5–1536-bit Diffie Hellman group Default setting for the Cisco CG-OS router is 2.
Step 10	**lifetime seconds** *value*	(Optional) Specifies the IKEv2 SA lifetime for the policy. Value is a range from 600 to 86400 seconds. Default setting is 86400 seconds.
Step 11	**exit**	Exits to the global configuration mode.
Step 12	**key** *pre-share key* **{address** *ip address* | **hostname** *name***}**	Configures the key that IKEv2 communicates to the peer (head-end router) ip address. The peer must have the same key in its configuration. Maximum per-shared ley length is 128.

To enable IPSec tunnelling on the Cisco CG-OS router and configure its parameters, follow these steps.

	Command	Purpose
Step 1	**feature crypto ipsec virtual-tunnel**	Enables IPSec tunnelling on the Cisco CG-OS router and creates a virtual tunnel interface.
Step 2	**crypto ipsec transform-set** *tx-form-name* {*txform* | *encr_txform auth_txform*}	Configures a single transform set that is included within the IPSec protection profile. Options for *txform* are: • esp-gcm 128–128-bit AES-GCM authenticated encryption AES-CBC • esp-gcm 256–256-bit AES-GCM authenticated encryption Options for *encr_txform* are: • esp-aes 128 *or* esp-aes 256 AES-CBC Options for *auth_txform* are: • esp-sha1-hmac *or* esp-sha256-hmac HMAC-SHA **Note** The transform-set name (tx-form-name) defined here must match that transform-set name associated with the IPSec profile in Step 5 below.
Step 3	**crypto ip sec profile** *profile-name*	Configures an IPSec profile for attachment to the tunnel interface.
Step 4	**description** *text*	(Optional) Allows the user to provide a description for the profile. The character limit is 64 characters.
Step 5	**set transform-set** *txfrom-name*	Associates the transformation set to the currently configured IPSec profile (see Step 2 above).
Step 6	**set pfs** *group*	(Optional) Configures the Diffie-Hillman group for perfect forward secrecy for the IPSec tunnel. Options for group are as follows: group 1–768-bit mode Diffie-Hillman group 4–2048-bit mode Diffie-Hillman group 2–1024-bit mode Diffie-Hillman group 5–1536-bit mode Diffie-Hillman By default, PFS is disabled.
Step 7	**set security-association lifetime** [*seconds*] [*kilobytes*]	(Optional) Specifies the lifetime of the IPSec security association. When the configured lifetime value expires, a new security association is negotiated. Lifetime can be expressed in both time (seconds, 120 to 86400) and data volume (kilobytes, 2560 to 4292967295). The default time value is 3600 seconds. The default data volume is 4608000 kilobytes.
Step 8	**exit**	Exits to the global configuration mode.

To enable OSPFv2 processing on the Cisco CG-OS router, enter the following command in the global configuration mode.

	Command	Purpose
Step 1	**router ospf** *instance-tag*	Creates an OSPFv2 routing instance. *instance-tag*–Internal identifier for the routing instance and can be an alphanumeric word or positive integer.

To enable OSPFv3 processing on the Cisco CG-OS router, enter the following command in the global configuration mode.

	Command	Purpose
Step 1	**router ospfv3** *instance-tag*	Creates an OSPFv3 routing instance. *instance-tag*–Internal identifier for the routing instance and can be an alphanumeric word or positive integer.

To enable tunneling on the Cisco CG-OS router, enter the following command in the global configuration mode.

	Command	Purpose
Step 1	**feature tunnel**	Enables tunneling on the Cisco CG-OS router.

To create an IPSec tunnel on the Cisco CG-OS router and define an OSPFv2 area on each of the interfaces (cellular and WiMAX), follow these steps to configure each interface.

An individual tunnel must be configured for each of the interfaces, cellular and WiMAX (see Figure 7-1).

	Command	Purpose
Step 1	**interface tunnel** *number*	Creates a virtual tunnel. When configuring a 3G or WiMAX interface, number is any value from 1 to 4095.
Step 2	**ip address** *ip address*	Assigns an IP address for the tunnel.
Step 3	**ip ospf cost** *interface-cost*	Specifies the cost of sending a packet on an interface.
Step 4	**ip ospf dead-interval** *seconds*	Sets the interval during which a router must receive at least one hello packet before the router declares that neighbor as down. The range in seconds is from 1 to 65535.
Step 5	**ip ospf hello-interval** *seconds*	Sets the interval between hello packets that OSPFv2 sends on the interface. The range in seconds is from 1 to 65535.
Step 6	**ip ospf mtu-ignore**	Disables the OSPFv2 MTU mismatch detection on database descriptor (DBD) packets. **Note** This parameter allows systems with Cisco IOS (such as the head-end router) and the Cisco CG-OS router to communicate and must be configured.

	Command	Purpose
Step 7	**ip router ospf** *instance-tag* **area** *area-id* **[secondaries none]**	Creates an OSPFv2 area on the interface.
Step 8	**tunnel mode ipsec ipv4**	Configures the encapsulation mode for the tunnel.
Step 9	**tunnel source** {*ip-address* | *interface-type slot-port*}	Configures the source endpoint for the tunnel.
Step 10	**tunnel destination** {*ip-address* | *host-name*}	Configures the destination endpoint for the tunnel.
Step 11	**no keepalive** *value*	(Optional) Disables the keepalive value on the Cisco CG-OS router. The **keepalive** *value* command configures the frequency of keep alive messages sent between peers in the tunnel. Keep alive messages validate the ability of peers to send and receive traffic. Value can be any number between 120 and 86400 seconds. The default value is 3600 seconds.
Step 12	**tunnel protection ipsec profile** *profile-name*	Binds the IPSec protection profile to the tunnel interfaces. **Note** The *profile-name* defined in this step must match the profile name assigned to the virtual tunnel interface in Step 3 by using the **crypto ip sec profile** *profile-name* command.
Step 13	**no shutdown**	(Optional) Brings the port up, administratively.
Step 14	**exit**	Exits the interface configuration mode.

To configure a Generic Routing Encapsulation (GRE) tunnel on the Cisco CG-OS router, follow these steps.

	Command	Purpose
Step 1	**interface tunnel** *number*	Creates a GRE tunnel on an interface. *number*–Any value from 1 to 4095.
Step 2	**ip address** *ip address*	Assigns an IPv4 address to the tunnel interface.
Step 3	**ipv6 address** *addr*	Assigns an IPv6 address to the tunnel interface. The format is A:B::C:D/length. The length range is 1 to 128.
Step 4	**ipv6 router ospfv3** *instance-tag* **area** *area-id* **[secondaries none]**	Creates an OSPFv3 area on the tunnel interface.
Step 5	**tunnel source** {*ip-address* | *interface-type slot-port*}	Configures the source endpoint for the tunnel.
Step 6	**tunnel destination** {*ip-address* | *host-name*}	Configures the destination endpoint for the tunnel.
Step 7	**no shutdown**	(Optional) Brings the port up, administratively.

EXAMPLE

Cisco CG-OS Router Configuration

The following example shows how to configure the Cisco CG-OS router.

For details on the head-end router configuration, refer to the Head-End Router Configuration.

These commands show how to configure the loopback0 interface, the cellular interface, and the WiMAX interface and their parameters.

```
router(config)# interface loopback 0
router(config-if)# ip address 20.0.0.2/24
router(config-if)# ip router ospf 2 area 0.0.0.2
router(config-if)# exit
router(config)# chat-script gsm1 PROFILE1
router(config)# interface dialer 1
router(config-if)# dialer persistent
router(config-if)# dialer pool 1
router(config-if)# dialer string gsm 1
router(config-if)# exit
router(config)# interface cellular 3/1
router(config-if)# dialer pool-member 1
router(config-if)# no shutdown
router(config-if)# exit
router(config)# wimax scan-list aaa
router(config-if)# channel index 1 frequency 2550000 bandwidth 10Mhz
router(config-if)# nap id 00:00:01 priority 1 channel-index 1
router(config-if)# nsp id 00:00:01 home
router(config-if)# exit
router(config)# interface wimax 5/1
router(config-if)# scan-list aaa
router(config-if)# ip address 192.10.0.21/16
router(config-if)# no shutdown
router(config-if)# exit
router(config)#
```

These commands show how to enable IKEv2 on the Cisco CG-OS router.

```
router(config)# feature crypto ike
router(config)# crypto ike domain ipsec
router(config-ike-ipsec)# policy 10
router(config-ike-ipsec-policy)# authentication pre-share
router(config-ike-ipsec-policy)# group 5
router(config-ike-ipsec-policy)# exit
router(config-ike-ipsec)# key company123 address 11.0.0.1
router(config-ike-ipsec)# key company123 address 192.10.0.1
router(config-ike-ipsec)# key company123 address 192.168.168.1
router(config-ike-ipsec)# exit
router(config)#
```

These commands show how to enable IPSec tunnelling (and configure its parameters) on the Cisco CG-OS router.

```
router(config)# feature crypto ipsec virtual-tunnel
router(config)# crypto ipsec transform-set MyTransformSet esp-aes 128 esp-shal-hmac
router(config)# crypto ipsec profile MyProfile
router(config-ipsec-profile)# description VTI IPSec Profile
router(config-ipsec-profile)# set transform-set MyTransformSet
router(config-ipsec-profile)# exit
router(config)#
```

This command shows how to enable OSPFv2 processing on the Cisco CG-OS router.

```
router(config)# router ospf 2
```

This command shows how to enable OSPFv3 processing on the Cisco CG-OS router.

```
router(config)# router ospfv3 1
```

This command shows how to enable tunneling on the Cisco CG-OS router.

```
router(config)# feature tunnel
```

These commands show how to configure tunnels that transport WiMAX (tunnel 15) and cellular (tunnel 15) data within the IPSec virtual tunnel from the Cisco CG-OS router to the head-end router.

```
router(config)# interface tunnel 15
router(config-if)# ip address 23.0.5.2/30
router(config-if)# ip ospf cost 100
router(config-if)# ip ospf dead-interval 20
router(config-if)# ip ospf hello-interval 5
router(config-if)# ip ospf mtu-ignore
router(config-if)# ip router ospf 2 area 0.0.0.2
router(config-if)# tunnel mode ipsec ipv4
router(config-if)# tunnel source wimax 5/1
router(config-if)# tunnel destination 192.10.0.1
router(config-if)# no keepalive
router(config-if)# tunnel protection ipsec profile MyProfile
router(config-if)# no shutdown
router(config-if)# exit
router(config)# interface tunnel 16
router(config-if)# ip address 23.0.6.2/30
router(config-if)# ip ospf cost 500
router(config-if)# ip ospf dead-interval 20
router(config-if)# ip ospf hello-interval 5
router(config-if)# ip ospf mtu-ignore
router(config-if)# ip router ospf 2 area 0.0.0.2
router(config-if)# tunnel mode ipsec ipv4
router(config-if)# tunnel source cellular 3/1
router(config-if)# tunnel destination 192.168.168.1
router(config-if)# no keepalive
router(config-if)# tunnel protection ipsec profile MyProfile
router(config-if)# no shutdown
router(config-if)# exit
router(config)#
```

These commands show how to configure a Generic Routing Encapsulation (GRE) tunnel on the Cisco CG-OS router.

```
router(config)# interface tunnel 1
router(config-if)# ip address 25.0.0.2/24
router(config-if)# ip address 25::2/64
router(config-if)# ipv6 router ospfv3 1 area 0.0.0.1
router(config-if)# tunnel source loopback 0
router(config-if)# tunnel destination 20.0.0.1
router(config-if)# no shutdown
router(config)#
```

Head-End Router Configuration

The following example shows how to configure the head-end router to communicate with the Cisco CG-OS router defined above.

This sample configuration is for a Cisco ASR 1000 Series system operating with Cisco IOS. For more details on the Cisco ASR and its configuration, refer to the following URL: http://www.cisco.com/en/US/partner/products/ps9343/index.html

These commands show how to configure the loopback0 interface on the head-end router.

```
HeadEndRtr(config)# interface loopback 0
HeadEndRtr(config-if)# ip address 20.0.0.1 255.255.255.0
HeadEndRtr(config-if)# exit
HeadEndRtr(config)#
```

These commands show how to enable IKEv2 on the head-end router.

```
HeadEndRtr(config)# crypto ikev2 keyring MyIke2KeyRing
HeadEndRtr(config-ike)#  peer wimax-cgr
HeadEndRtr(config-ike)#  address 192.10.0.21
HeadEndRtr(config-ike)#  pre-shared-key Cisco123
HeadEndRtr(config-ike)#  peer cellular-cgr
HeadEndRtr(config-ike)#  address 192.168.168.21
HeadEndRtr(config-ike)#  pre-shared-key Cisco123
HeadEndRtr(config-ike)# exit
HeadEndRtr(config)# cryto ikev2 profile MyIke2Profile
HeadEndRtr(config-ike)# match fvrf any
HeadEndRtr(config-ike)# match identity remote address 192.10.0.21 255.255.255.255
HeadEndRtr(config-ike)# match identity remote address 192.168.168.19 255.255.255.255
HeadEndRtr(config-ike)# authentication local pre-share
HeadEndRtr(config-ike)# authentication remote pre-share
HeadEndRtr(config-ike)# keyring MyIke2KeyRing
HeadEndRtr(config-ike)# exit
HeadEndRtr(config)#
```

These commands show how to enable IPSec tunnelling and configure its parameters on the head-end router.

```
HeadEndRtr(config)# crypto ipsec transform-set AES128SHA1 esp-aes esp-sha-hmac
HeadEndRtr(config)# crypto ipsec profile IPSecProfile
HeadEndRtr(config-ipsec-profile)# set transform-set AES128SHA1
HeadEndRtr(config-ipsec-profile)# set ikev2-profile MyIke2Profile
HeadEndRtr(config-ipsec-profile)# responder-only
HeadEndRtr(config-ipsec-profile)# exit
HeadEndRtr(config)#
```

These commands show how to configure tunnels that transport WiMAX (tunnel 15) and cellular (tunnel 16) data from the head-end router to the Cisco CG-OS router.

```
HeadEndRtr(config)# interface tunnel 15
HeadEndRtr(config-if)# ip address 23.0.5.1 255.255.255.252
HeadEndRtr(config-if)# ip ospf cost 100
HeadEndRtr(config-if)# ip ospf hello-interval 5
HeadEndRtr(config-if)# ip ospf mtu-ignore
HeadEndRtr(config-if)# tunnel source GigabitEthernet 0/2
HeadEndRtr(config-if)# tunnel mode ipsec ipv4
HeadEndRtr(config-if)# tunnel destination 192.10.0.21
HeadEndRtr(config-if)# tunnel protection ipsec profile IPSecProfile
HeadEndRtr(config-if)# exit
HeadEndRtr(config)# interface tunnel 16
HeadEndRtr(config-if)# ip address 23.0.6.1 255.255.255.252
HeadEndRtr(config-if)# ip ospf cost 500
HeadEndRtr(config-if)# ip ospf hello-interval 5
HeadEndRtr(config-if)# ip ospf mtu-ignore
HeadEndRtr(config-if)# tunnel source GigabitEthernet 0/1.168
HeadEndRtr(config-if)# tunnel mode ipsec ipv4
HeadEndRtr(config-if)# tunnel destination 192.168.168.21
HeadEndRtr(config-if)# tunnel protection ipsec profile IPSecProfile
HeadEndRtr(config-if)# exit
HeadEndRtr(config)#
```

These command shows how to create a GRE tunnel on the head-end router.

```
HeadEndRtr(config)# interface tunnel 1
HeadEndRtr(config)# ip address 25.0.0.1 255.255.255.0
HeadEndRtr(config-if)# ip ospf mtu-ignore
HeadEndRtr(config-if)# ip address 25::1/64
HeadEndRtr(config-if)# ipv6 enable
HeadEndRtr(config-if)# ipv6 ospf mtu-ignore
HeadEndRtr(config-if)# ipv6 ospf 1 area 1
HeadEndRtr(config-if)# tunnel source loopback 0
HeadEndRtr(config-if)# tunnel destination 20.0.0.2
HeadEndRtr(config-if)# exit
HeadEndRtr(config)#
```

These command shows how to enable OSPFv3 processing, define OSPFv3 areas, and enable the OSPFv3 for IPv6 router configuration mode (which allows you to configure IPv6 parameters) on the head-end router.

```
HeadEndRtr(config)# router ospf 2
HeadEndRtr(config-router)# network 20.0.0.0 0.0.0.255 area 2
HeadEndRtr(config-router)# network 23.0.5.0 0.0.0.255 area 2
HeadEndRtr(config-router)# network 23.0.6.0 0.0.0.255 area 2
HeadEndRtr(config-router)# exit
HeadEndRtr(config)# ipv6 router ospf 1
```

Verifying the WAN Backhaul Configuration

To display the WAN backhaul configuration, enter any or all of the following commands on the Cisco CG-OS router:

Note The Cisco CG-OS software does not support the [**vrf** {*vrf-name* | **all** | **default** | **management**}] parameter in the commands listed below.

Command	Purpose
show crypto ike domain ipsec	Displays the IKEv2 configuration.
show crypto ike domain ipsec policy	Displays the configured IKEv2 policies.
show crypto ipsec profile *profile_name*	Displays all defined IPSec profiles.
show crypto ipsec transform set *tx-form name*	Displays all defined transform-sets.
show file volatile:///bhmgr.tcl	Confirms installation status of the backhaul manager script (loaded during manufacture) and location of the script.
show {ip ospf | ospfv3}	Displays the configuration of the selected option.

Command	Purpose
show ipv6 ospfv3 interface	Displays configured tunnels and their state and configuration.
show running-config {ospf | ospfv3}	Displays the currently running OSPFv2 or OSPFv3 configuration.
show running-config interface {wimax | cellular} *slot/port*	Displays the currently running WiMAX or cellular configuration.
show tunnel internal database reachability	Displays the configured tunnel names and IP addresses.

Monitoring Statistics

To display WAN backhaul statistics, enter the following commands:

Command	Purpose
show {ip ospf | ospfv3} policy statistics area *area-id* **filter-list** {**in | out**}	Displays the route policy statistics for an area for the selected option.
show {ip ospf | ospfv3 | ipv6 ospfv3} statistics	Displays event counters for the selected option.
show {ip ospf | opsfv3 | ipv6 ospfv3} traffic [*interface-type slot/port*]	Displays the packet counters for the selected option.

Configuration Example

See EXAMPLE, page 7-9.

CHAPTER 8

Configuring Static Routing

This chapter describes how to configure static routing on the Cisco 1000 Series Connected Grid Routers (*hereafter* referred to as the Cisco CG-OS router). The system software for the router is identified as the Cisco CG-OS software.

This chapter includes the following sections:

Information About Static Routing

Routers forward packets using either route information from route table entries that you manually configure or the route information that is calculated using dynamic routing algorithms.

Static routes, which define explicit paths between two routers, cannot be automatically updated. Static routes must be manually configured when network changes occur. Static routes use less bandwidth than dynamic routes. Additionally, the router uses no CPU cycles when calculating and analyzing routing updates.

You can supplement dynamic routes with static routes where appropriate. You can redistribute static routes into dynamic routing algorithms but you cannot redistribute routing information calculated by dynamic routing algorithms into the static routing table.

Cisco recommends that you use static routes in environments where network traffic is predictable and where the network design is simple. Large, constantly changing networks are not compatible with static routes because they cannot react to network changes. Most networks use dynamic routes to communicate between routers but might have one or two static routes configured for special cases. Static routes are also useful for specifying a gateway of last resort (a default router to which all unroutable packets are sent).

Static routing applies to both IPv4 and IPv6 environments.

This section includes the following topics:

- Administrative Distance, page 8-2

Administrative Distance

An administrative distance is the metric used by routers to choose the best path when there are two or more routes to the same destination from two different routing protocols. An administrative distance guides the selection of one routing protocol (or static route) over another, when more than one protocol adds the same route to the unicast routing table. Each routing protocol is prioritized in order of most to least reliable using an administrative distance value.

Static routes have a default administrative distance of 1. A router prefers a static route to a dynamic route because the router considers a route with a low number to be the shortest. When you want a dynamic route to override a static route, you can specify an administrative distance for the static route. For example, when you have two dynamic routes with an administrative distance of 120, you would specify an administrative distance that is greater than 120 for the static route when you want the dynamic route to override the static route.

Directly Connected Static Routes

You must specify only the output interface (the interface on which all packets are sent to the destination network) in a directly connected static route. The router assumes the destination is directly attached to the output interface and the packet destination is used as the next-hop address. The next hop can be an interface, only for point-to-point interfaces. For broadcast interfaces, the next hop must be an IPv4/IPv6 address.

Fully Specified Static Routes

You must specify either the output interface (the interface on which all packets are sent to the destination network) or the next-hop address in a fully specified static route. You can use a fully specified static route when the output interface is a multi-access interface and you need to identify the next-hop address. The next-hop address must be directly attached to the specified output interface.

Floating Static Routes

A floating static route is a static route that the router uses to back up a dynamic route. You must configure a floating static route with a higher administrative distance than the dynamic route that it backs up. In this instance, the router prefers a dynamic route to a floating static route. You can use a floating static route as a replacement if the dynamic route is lost.

Note By default, a router prefers a static route to a dynamic route because a static route has a smaller administrative distance than a dynamic route.

Remote Next Hops for Static Routes

You can specify the next-hop address of a neighboring router that is not directly connected to the router for static routes by using remote (not directly attached) next-hops. When a static route has remote next hops during data forwarding, the routing table uses the next hops entries to identify the corresponding directly attached next hops that have access to the remote next hops.

Prerequisites for Static Routing

Identify the IP addresses for use as static routes

Guidelines and Limitations for Static Routing

You can specify an interface as the next-hop address for a static route only for point-to-point interfaces such as generic routing encapsulation (GRE) tunnels.

When the next-hop address for a static route is unreachable, the router does not add the static route to the unicast routing table.

Default Settings

Table 8-1 lists the default setting for the static routing administrative distance parameter.

Table 8-1 Default Static Routing Parameters

Parameters	Default
administrative distance	1

Configuring Static Routing

This section includes the following topics:

- Configuring a Static Route, page 8-3
- Verifying the Static Routing Configuration, page 8-4

Configuring a Static Route

You can configure a static route on the router.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**ip route** {*ip-prefix* | *ip-addr ip-mask*} {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**tag** *tag-value* [*pref*]	Configures a static route and the interface for this static route. You can optionally configure the next-hop address. The *preference* value sets the administrative distance. The range is from 1 to 255. The default is 1.
	ipv6 route *ipv6-prefix* {*nh-prefix*|*link-local-nh-prefix*} | (*nexthop* [*interface*] | *link-local-nexthop* [*interface*]} [**tag** *tag-value* [*pref*]	Configures a static route and the interface for this static route. You can optionally configure the next-hop address. The *preference* value sets the administrative distance. The range is from 1 to 255. The default is 1.
Step 3	**show** {**ip** | **ipv6**} **static-route**	(Optional) Displays information about static routes.
Step 4	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to configure a IPv4 static route.

```
router# configure terminal
router(config)# ip route 192.0.2.0/8 ethernet 2/1 192.0.2.4
router(config)# copy running-config startup-config
```

This example shows how to configure a IPv6 static route.

```
router# configure terminal
router(config)# ipv6 route 2001:0DB8::/48 6::6 ethernet 2/1
router(config)# copy running-config startup-config
```

To remove the static route, use the **no** {**ip** | **ipv6**} **static-route** command.

Verifying the Static Routing Configuration

To display the static routing configuration, perform one of the following tasks.

Command	Purpose
show ip static-route	Displays the configured static routes.
show ipv6 static-route	Displays the configured static routes.

Configuration Example for Static Routing

This example shows how to configure a IPv4 static route.

```
configure terminal
 ip route 192.0.2.0/8 192.0.2.10
 copy running-config startup-config
```

GLOSSARY

A

ABR See area border router.

address family A specific type of network addressing supported by a routing protocol. Examples include IPv4 unicast and IPv4 multicast.

adjacency Two OSPF routers that have compatible configurations and have synchronized their link-state databases.

administrative distance A rating of the trustworthiness of a routing information source. In general, the higher the value, the lower the trust rating.

area A logical division of routers and links within an OSPF domain that creates separate subdomains. LSA flooding is contained within an area.

area border router A router that connects one OSPF area to another OSPF area.

ARP Address resolution protocol. ARP discovers the MAC address for a known IPv4 address.

AS See autonomous system.

ASBR See autonomous system border router.

autonomous system A network controlled by a single technical administration entity.

autonomous system border router A router that connect a an OSPF autonomous system to an external autonomous system.

B

backup designated router See BDR.

bandwidth The available traffic capacity of a link.

BDR Backup designated router. An elected router in a multi-access OSPF network that acts as the backup if the designated router fails. All neighbors form adjacencies with the backup designated router (BDR) as well as the designated router.

C

communication cost Measure of the operating cost to route over a link.

converged The point at which all routers in a network have identical routing information.

convergence See converged.

D

dead interval The time within which an OSPF router must receive a Hello packet from an OSPF neighbor. The dead interval is usually a multiple of the hello interval. If no Hello packet is received, the neighbor adjacency is removed.

default gateway A router to which all unroutable packets are sent. Also called the router of last resort.

delay The length of time required to move a packet from the source to the destination through the internetwork.

designated router See DR.

DHCP Dynamic Host Control Protocol.

distance vector Defines routes by distance (for example, the number of hops to the destination) and direction (for example, the next-hop router) and then broadcasts to the directly connected neighbor routers.

DNS client Domain Name System client. Communicates with DNS server to translate a hostname to an IP address.

DR Designated router. An elected router in a multi-access OSPF network that sends LSAs on behalf of all its adjacent neighbors. All neighbors establish adjacency with only the designated router and the backup designated router.

E

F

G

gateway A switch or router that forwards Layer 3 traffic from a LAN to the rest of the network.

graceful restart A feature that allows a router to remain in the data forwarding path while a routing protocol reboots.

H

hello packet A special message used by OSPF to discover neighbors. Also acts as a keep alive messages between established neighbors.

high availability The ability of a system or component to limit or avoid network disruption when a component fails.

I

ICMP Internet Control Message Protocol (ICMP)

IETF RFCs Internet Engineering Task Force Request for Comments.

instance An independent, configurable entity, typically a protocol.

IP tunnels A method of encapsulating packets within various Internet Protocols (IP) to interconnect communications between different networks.

IPv4 Internet Protocol version 4.

IPv6 Internet Protocol version 6.

K

keepalive A special message sent between routing peers to verify and maintain communications between the pair.

L

link cost An arbitrary number configured on an OSPF interface which is in shortest path first calculations.

link-state Shares information about a link and link cost to neighboring routers.

link-state advertisement See LSA.

LSA Link-state advertisement. An OSPF message to share information on the operational state of a link, link cost, and other OSPF neighbor information.

link-state database OSPF database of all LSAs received. OSPF uses this database to calculate the best path to each destination in the network.

link-state refresh The time that OSPF floods the network with LSAs to ensure all OSPF routers have the same information.

load The degree to which a network resource, such as a router, is busy.

load balancing The distribution of network traffic across multiple paths to a given destination.

M

message digest A one-way hash applied to a message using a shared password and appended to the message to authenticate the message and ensure the message has not been altered in transit.

metric A standard of measurement, such as the path bandwidth, that is used by routing algorithms to determine the optimal path to a destination.

MD5 authentication digest A cryptographic construction that is calculated based on an authentication key and the original message and sent along with the message to the destination. Allows the destination to determine the authenticity of the sender and guarantees that the message has not been tampered with during transmission.

MTU Maximum transmission unit. The largest packet size that a network link transmits without fragmentation.

N

NDP Neighbor Discovery Protocol. The protocol used by IPv6 to find the MAC address associated with an IPv6 address.

next hop The next router that a packet is sent to on its way to the destination address.

NSSA Not-So-Stubby-Area. Limits AS external LSAs in an OSPF area.

O

OSPF Open Shortest Path First. An IETF link-state protocol. OSPFv2 supports IPv4 and OSPFv3 supports IPv6.

P

path length Sum of all link costs or the hop count that a packet experiences when routed from the source to the destination.

policy-based routing The method of using route maps to alter the route selected for a packet.

R

redistribution One routing protocol accepts route information from another routing protocol and advertises it in the local autonomous system.

reliability The dependability (usually described in terms of the bit-error rate) of each network link.

RIB Routing Information Base. Maintains the routing table with directly connected routes, static routes, and routes learned from dynamic unicast routing protocols.

Routing Information Base See RIB.

route map A construct used to map a route or packet based on match criteria and optionally alter the route or packet based on set criteria. Used in route redistribution and policy-based routing.

route summarization A process that replaces a series of related, specific routes in a route table with a more generic route.

router ID A unique identifier used by routing protocols. If not manually configured, the routing protocol selects the highest IP address configured on the system.

S

SPF algorithm Shortest Path First algorithm. Dijkstra's algorithm used by OSPF to determine the shortest route through a network to a particular destination.

split horizon Routes learned from an interface are not advertised back along the interface they were learned on, preventing the router from seeing its own route updates.

split horizon with poison reverse Routes learned from an interface are set as unreachable and advertised back along the interface they were learned on, preventing the router from seeing its own route updates.

static route A manually configured route.

stub area An OSPF area that does not allow AS External (type 5) LSAs.

stub router A router that has no direct connection to the main network and which routes to that network using a known remote router.

SVI switched virtual interface.

U

U6FIB Unicast IPv6 Forwarding Information Base.

UFIB Unicast Forwarding Information Base for IPv4.

U6RIB Unicast IPv6 Routing Information Base. The unicast routing table that gathers information from all routing protocols and updates the forwarding information base for each module.

URIB Unicast Routing Information Base for IPv4. The unicast routing table that gathers information from all routing protocols and updates the forwarding information base for each module.

V

VDC virtual device context. Used to split a physical system into secure, independent, logical systems.

virtualization A method of making a physical entity act as multiple, independent logical entities.