CHAPTER 37

Configuring IP Unicast Routing

This chapter describes how to configure IP Version 4 (IPv4) unicast routing on the switch. Unless otherwise noted, the term *switch* refers to a standalone switch and to a switch stack. A switch stack operates and appears as a single router to the rest of the routers in the network.

Basic routing functions, including static routing and the Routing Information Protocol (RIP), are available with both the IP base feature set and the IP services feature set. To use advanced routing features and other routing protocols, you must have the IP services feature set enabled on the standalone switch or on the stack master.

Note If the switch or switch stack is running the advanced IP services feature set, you can also enable IP Version 6 (IPv6) unicast routing and configure interfaces to forward IPv6 traffic in addition to IPv4 traffic. For information about configuring IPv6 on the switch, see Chapter 38, "Configuring IPv6 Unicast Routing."

For more detailed IP unicast configuration information, see the *Cisco IOS IP Configuration Guide, Release 12.2* from the Cisco.com page under **Documentation** > **Cisco IOS Software** > **12.2 Mainline** > **Configuration Guides**. For complete syntax and usage information for the commands used in this chapter, see these command references from the Cisco.com page under **Documentation** > **Cisco IOS Software** > **12.2 Mainline** > **Command References**:

- *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services, Release 12.2*
- *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.2*
- *Cisco IOS IP Command Reference, Volume 3 of 3: Multicast, Release 12.2*

This chapter consists of these sections:

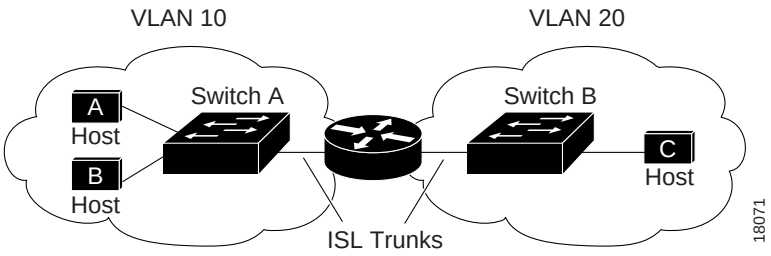
Note When configuring routing parameters on the switch and to allocate system resources to maximize the number of unicast routes allowed, you can use the **sdm prefer routing** global configuration command to set the Switch Database Management (SDM) feature to the routing template. For more information on the SDM templates, see Chapter 8, "Configuring SDM Templates," or see the **sdm prefer** command in the command reference for this release.

Understanding IP Routing

In some network environments, VLANs are associated with individual networks or subnetworks. In an IP network, each subnetwork is mapped to an individual VLAN. Configuring VLANs helps control the size of the broadcast domain and keeps local traffic local. However, network devices in different VLANs cannot communicate with one another without a Layer 3 device (router) to route traffic between the VLAN, referred to as inter-VLAN routing. You configure one or more routers to route traffic to the appropriate destination VLAN.

Figure 37-1 shows a basic routing topology. Switch A is in VLAN 10, and Switch B is in VLAN 20. The router has an interface in each VLAN.

Figure 37-1 Routing Topology Example



When Host A in VLAN 10 needs to communicate with Host B in VLAN 10, it sends a packet addressed to that host. Switch A forwards the packet directly to Host B without sending it to the router.

When Host A sends a packet to Host C in VLAN 20, Switch A forwards the packet to the router, which receives the traffic on the VLAN 10 interface. The router surveys the routing table, finds the correct outgoing interface, and forwards the packet on the VLAN 20 interface to Switch B. Switch B receives the packet and forwards it to Host C.

This section contains information on these routing topics:

Types of Routing

Routers and Layer 3 switches can route packets in these ways:

- By using default routing
- By using preprogrammed static routes for the traffic
- By dynamically calculating routes by using a routing protocol

Default routing refers to sending traffic with a destination unknown to the router to a default outlet or destination.

Static unicast routing forwards packets from predetermined ports through a single path into and out of a network. Static routing is secure and uses little bandwidth, but it does not automatically respond to changes in the network, such as link failures. Therefore, network changes might result in unreachable destinations. As networks grow, static routing becomes a labor-intensive liability.

Routers use these dynamic routing protocols to dynamically calculate the best route for forwarding traffic:

- Routers that use distance-vector protocols maintain routing tables with distance values of networked resources and periodically pass these tables to their neighbors. Distance-vector protocols use one or a series of metrics for calculating the best routes.
- Routers using link-state protocols maintain a complex database of network topology based on the exchange of link-state advertisements (LSAs) between routers. LSAs are triggered by an event in the network, which speeds up the convergence time or time required to respond to these changes. Link-state protocols respond quickly to topology changes but require greater bandwidth and more resources than distance-vector protocols.

Distance-vector protocols supported by the switch use Routing Information Protocol (RIP), a single-distance metric (cost) that determines the best path, and Border Gateway Protocol (BGP), which adds a path vector mechanism. The switch also supports the Open Shortest Path First (OSPF) link-state protocol and Enhanced IGRP (EIGRP), which adds some link-state routing features to traditional Interior Gateway Routing Protocol (IGRP) to improve efficiency.

Note On a switch or switch stack, the supported protocols are determined by the software running on the switch or stack master. If the switch or stack master is running the IP base feature set, only default routing, static routing and RIP are supported. All other routing protocols require the IP services feature set.

IP Routing and Switch Stacks

A switch stack appears to the network as a single router, regardless of which switch in the stack is connected to a routing peer. For additional information about switch stack operation, see Chapter 5, “Managing Switch Stacks.”

The stack master performs these functions:

- It initializes and configures the routing protocols.
- It sends routing protocol messages and updates to other routers.
- It processes routing protocol messages and updates received from peer routers.
- It generates, maintains, and distributes the distributed Cisco Express Forwarding (dCEF) database to all stack members. The routes are programmed on all switches in the stack bases on this database.

- The MAC address of the stack master is used as the router MAC address for the whole stack, and all outside devices use this address to send IP packets to the stack.
- All IP packets that require software forwarding or processing go through the CPU of the stack master.

Stack members perform these functions:

- They act as routing standby switches, ready to take over in case they are elected as the new stack master if the stack master fails.
- They program the routes into hardware. The routes programmed by the stack members are the same that are downloaded by the stack master as part of the dCEF database.

If a stack master fails, the stack detects that the stack master is down and elects one of the stack members to be the new stack master. During this period, except for a momentary interruption, the hardware continues to forward packets with no active protocols.

However, even though the switch stack maintains the hardware identification after a failure, the routing protocols on the router neighbors might flap during the brief interruption before the stack master restarts. Routing protocols such as OSPF and EIGRP need to recognize neighbor transitions. The router uses two levels of nonstop forwarding (NSF) to detect a change-over, to continue forwarding network traffic, and to recover route information from peer devices:

- NSF-aware routers tolerate neighboring router failures. After the neighbor router restarts, an NSF-aware router supplies information about its state and route adjacencies on request.
- NSF-capable routers support NSF. When they detect a stack master change, they rebuild routing information from NSF-aware or NSF-capable neighbors and do not wait for a restart.

The switch stack supports NSF-capable routing for OSPF and EIGRP. For more information, see the "OSPF NSF Capability" section on page 37-29 and the "EIGRP NSF Capability" section on page 37-39.

Upon election, the new stack master performs these functions:

- It starts generating, receiving, and processing routing updates.
- It builds routing tables, generates the CEF database, and distributes it to stack members.
- It uses its MAC address as the router MAC address. To notify its network peers of the new MAC address, it periodically (every few seconds for 5 minutes) sends a gratuitous ARP reply with the new router MAC address.

Note If you configure the persistent MAC address feature on the stack and the stack master changes, the stack MAC address does not change for the configured time period. If the previous stack master rejoins the stack as a member switch during that time period, the stack MAC address remains the MAC address of the previous stack master. See the "Enabling Persistent MAC Address" section on page 5-21.

- It attempts to determine the reachability of every proxy ARP entry by sending an ARP request to the proxy ARP IP address and receiving an ARP reply. For each reachable proxy ARP IP address, it generates a gratuitous ARP reply with the new router MAC address. This process is repeated for 5 minutes after a new stack master election.

Note When a stack master is running the IP services feature set, the stack can run all supported protocols, including Open Shortest Path First (OSPF), Enhanced IGRP (EIGRP), and Border Gateway Protocol (BGP). If the stack master fails and the new elected stack master is running the IP base feature set, these protocols no longer run on the stack.

Caution Partitioning on the switch stack into two or more stacks might lead to undesirable behavior in the network.

Steps for Configuring Routing

By default, IP routing is disabled on the switch, and you must enable it before routing can take place. For detailed IP routing configuration information, see the *Cisco IOS IP Configuration Guide, Release 12.2* from the Cisco.com page under **Documentation** > **Cisco IOS Software** > **12.2 Mainline** > **Configuration Guides**.

In the following procedures, the specified interface must be one of these Layer 3 interfaces:

- A routed port: a physical port configured as a Layer 3 port by using the **no switchport** interface configuration command.
- A switch virtual interface (SVI): a VLAN interface created by using the **interface vlan** *vlan_id* global configuration command and by default a Layer 3 interface.
- An EtherChannel port channel in Layer 3 mode: a port-channel logical interface created by using the **interface port-channel** *port-channel-number* global configuration command and binding the Ethernet interface into the channel group. For more information, see the "Configuring Layer 3 EtherChannels" section on page 36-14.

Note The switch does not support tunnel interfaces for unicast routed traffic.

All Layer 3 interfaces on which routing will occur must have IP addresses assigned to them. See the "Assigning IP Addresses to Network Interfaces" section on page 37-7.

A Layer 3 switch can have an IP address assigned to each routed port and SVI. The number of routed ports and SVIs that you can configure is not limited by software. However, the interrelationship between this number and the number and volume of features being implemented might have an impact on CPU utilization because of hardware limitations. To optimize system memory for routing, use the **sdm prefer routing** global configuration command.

Configuring routing consists of several main procedures:

- To support VLAN interfaces, create and configure VLANs on the switch or switch stack, and assign VLAN membership to Layer 2 interfaces. For more information, see Chapter 12, "Configuring VLANs."
- Configure Layer 3 interfaces.
- Enable IP routing on the switch.
- Assign IP addresses to the Layer 3 interfaces.
- Enable selected routing protocols on the switch.
- Configure routing protocol parameters (optional).

Configuring IP Addressing

A required task for configuring IP routing is to assign IP addresses to Layer 3 network interfaces to enable the interfaces and to allow communication with the hosts on those interfaces that use IP. These sections describe how to configure various IP addressing features. Assigning IP addresses to the interface is required; the other procedures are optional.

Default Addressing Configuration

Table 37-1 shows the default addressing configuration.

Table 37-1 Default Addressing Configuration

Feature	Default Setting
IP address	None defined.
ARP	No permanent entries in the Address Resolution Protocol (ARP) cache. Encapsulation: Standard Ethernet-style ARP. Timeout: 14400 seconds (4 hours).
IP broadcast address	255.255.255.255 (all ones).
IP classless routing	Enabled.
IP default gateway	Disabled.
IP directed broadcast	Disabled (all IP directed broadcasts are dropped).
IP domain	Domain list: No domain names defined. Domain lookup: Enabled. Domain name: Enabled.
IP forward-protocol	If a helper address is defined or User Datagram Protocol (UDP) flooding is configured, UDP forwarding is enabled on default ports. Any-local-broadcast: Disabled. Spanning Tree Protocol (STP): Disabled. Turbo-flood: Disabled.
IP helper address	Disabled.
IP host	Disabled.

Table 37-1 Default Addressing Configuration (continued)

Feature	Default Setting
IRDP	Disabled. Defaults when enabled: • Broadcast IRDP advertisements. • Maximum interval between advertisements: 600 seconds. • Minimum interval between advertisements: 0.75 times maximum interval • Preference: 0.
IP proxy ARP	Enabled.
IP routing	Disabled.
IP subnet-zero	Disabled.

Assigning IP Addresses to Network Interfaces

An IP address identifies a location to which IP packets can be sent. Some IP addresses are reserved for special uses and cannot be used for host, subnet, or network addresses. RFC 1166, “Internet Numbers,” contains the official description of IP addresses.

An interface can have one primary IP 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 referred to as a subnet mask. To receive an assigned network number, contact your Internet service provider.

Beginning in privileged EXEC mode, follow these steps to assign an IP address and a network mask to a Layer 3 interface:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode, and specify the Layer 3 interface to configure.
Step 3	**no switchport**	Remove the interface from Layer 2 configuration mode (if it is a physical interface).
Step 4	**ip address** *ip-address subnet-mask*	Configure the IP address and IP subnet mask.
Step 5	**no shutdown**	Enable the interface.
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show interfaces** [*interface-id*] **show ip interface** [*interface-id*] **show running-config interface** [*interface-id*]	Verify your entries.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use of Subnet Zero

We strongly discourage subnetting with a subnet address of zero because of the problems that can arise if a network and a subnet have the same addresses. For example, if network 131.108.0.0 is subnetted as 255.255.255.0, subnet zero would be written as 131.108.0.0, which is the same as the network address.

You can use the all ones subnet (131.108.255.0) and, even though we discourage this practice, you can enable the subnet zero if you need the entire subnet space for your IP address.

Beginning in privileged EXEC mode, follow these steps to enable subnet zero:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip subnet-zero**	Enable the use of subnet zero for interface addresses and routing updates.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show running-config**	Verify your entry.
Step 5	**copy running-config startup-config**	(Optional) Save your entry in the configuration file.

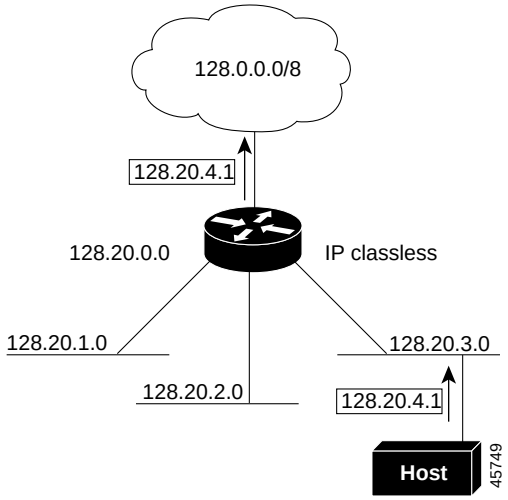
Use the **no ip subnet-zero** global configuration command to restore the default and to disable the use of subnet zero.

Classless Routing

By default, classless routing is enabled when the switch is configured to route. With classless routing, if a router receives packets for a subnet of a network with no default route, the router forwards the packet to the best supernet route. A *supernet* is contiguous blocks of Class C address spaces simulating a single, larger address space and is designed to relieve the pressure on the rapidly depleting Class B address space.

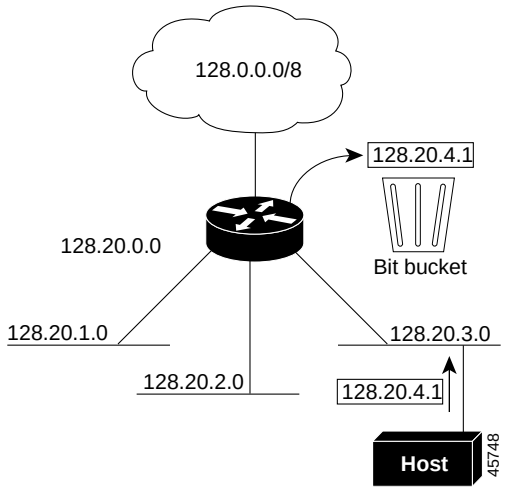
In Figure 37-2, classless routing is enabled. When the host sends a packet to 120.20.4.1, instead of discarding the packet, the router forwards it to the best supernet route. If you disable classless routing and a router receives packets destined for a subnet of a network with no network default route, the router discards the packet.

Figure 37-2 IP Classless Routing



In Figure 37-3, the router in network 128.20.0.0 is connected to subnets 128.20.1.0, 128.20.2.0, and 128.20.3.0. If the host sends a packet to 120.20.4.1, because there is no network default route, the router discards the packet.

Figure 37-3 No IP Classless Routing



To prevent the switch from forwarding packets destined for unrecognized subnets to the best supernet route possible, you can disable classless routing behavior.

Beginning in privileged EXEC mode, follow these steps to disable classless routing:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**no ip classless**	Disable classless routing behavior.
Step 3	**end**	Return to privileged EXEC mode.

	Command	Purpose
Step 4	**show running-config**	Verify your entry.
Step 5	**copy running-config startup-config**	(Optional) Save your entry in the configuration file.

To restore the default so that the switch forwards packets destined for a subnet of a network without a network default route to the best possible supernet route, use the **ip classless** global configuration command.

Configuring Address Resolution Methods

You can control interface-specific handling of IP by using address resolution. A device using IP can have both a local address or a MAC address, which uniquely defines the device on its local segment or LAN, and a network address, which identifies the network to which the device belongs.

Note In a switch stack, network communication uses a single MAC address and the IP address of the stack.

The local address or the MAC address is known as a data-link address because it is contained in the data-link layer (Layer 2) section of the packet header and is read by data-link (Layer 2) devices. To communicate with an Ethernet device, the software must learn the MAC address of the device. The process of learning the MAC address from an IP address is called *address resolution*. The process of learning the IP address from the MAC address is called *reverse address resolution*.

The switch uses these types of address resolution:

- Address Resolution Protocol (ARP) associates IP address with MAC addresses. Using an IP address as input, ARP learns the associated MAC address and then stores the IP address/MAC address association in an ARP cache for rapid retrieval. Then the IP datagram is encapsulated in a link-layer frame and sent over the network. Encapsulation of IP datagrams and ARP requests or replies on IEEE 802 networks other than Ethernet is specified by the Subnetwork Access Protocol (SNAP).
- Proxy ARP helps hosts with no routing tables learn the MAC addresses of hosts on other networks or subnets. If the switch (router) receives an ARP request for a host that is not on the same interface as the ARP request sender, and if the router has all of its routes to the host through other interfaces, it generates a proxy ARP packet giving its own local data-link address. The host that sent the ARP request then sends its packets to the router, which forwards them to the intended host.

The switch also uses the Reverse Address Resolution Protocol (RARP), which functions the same as ARP does, except that the RARP packets request an IP address instead of a local MAC address. Using RARP requires a RARP server on the same network segment as the router interface. Use the **ip rarp-server** *address* interface configuration command to identify the server.

For more information on RARP, see the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* under **Documentation** > **Cisco IOS Software** > **12.2 Mainline** > **Configuration Guides** from the Cisco.com page.

You can perform these tasks to configure address resolution:

- Define a Static ARP Cache, page 37-11
- Set ARP Encapsulation, page 37-11
- Enable Proxy ARP, page 37-12

Define a Static ARP Cache

ARP and other address resolution protocols provide dynamic mapping between IP addresses and MAC addresses. Because most hosts support dynamic address resolution, you usually do not need to specify static ARP cache entries. If you must define a static ARP cache entry, you can do so globally, which installs a permanent entry in the ARP cache that the switch uses to translate IP addresses into MAC addresses. You can also specify that the switch respond to ARP requests as if it were the owner of the specified IP address. If you do not want the ARP entry to be permanent, you can specify a timeout period for the ARP entry.

Beginning in privileged EXEC mode, follow these steps to provide static mapping between IP addresses and MAC addresses:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**arp** *ip-address hardware-address type*	Globally associate an IP address with a MAC (hardware) address in the ARP cache, and specify encapsulation type as one of these: • **arpa**—ARP encapsulation for Ethernet interfaces • **snap**—Subnetwork Address Protocol encapsulation for Token Ring and FDDI interfaces • **sap**—HP's ARP type
Step 3	**arp** *ip-address hardware-address type* [*alias*]	(Optional) Specify that the switch respond to ARP requests as if it were the owner of the specified IP address.
Step 4	**interface** *interface-id*	Enter interface configuration mode, and specify the interface to configure.
Step 5	**arp timeout** *seconds*	(Optional) Set the length of time that an ARP cache entry stays in the cache. The range is 0 to 2147483 seconds. The default is 14400 seconds (4 hours).
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show interfaces** [*interface-id*]	Verify the type of ARP and the timeout value used on all interfaces or on a specific interface.
Step 8	**show arp** or **show ip arp**	View the contents of the ARP cache.
Step 9	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To remove an entry from the ARP cache, use the **no arp** *ip-address hardware-address type* global configuration command. To remove all nonstatic entries from the ARP cache, use the **clear arp-cache** privileged EXEC command**.**

Set ARP Encapsulation

By default, Ethernet ARP encapsulation (represented by the **arpa** keyword) is enabled on an IP interface. You can change the encapsulation method to SNAP if required by your network.

Beginning in privileged EXEC mode, follow these steps to specify the ARP encapsulation type:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode, and specify the Layer 3 interface to configure.
Step 3	**arp** {**arpa** | **snap**}	Specify the ARP encapsulation method: • **arpa**—Address Resolution Protocol • **snap**—Subnetwork Address Protocol
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show interfaces** [*interface-id*]	Verify ARP encapsulation configuration on all interfaces or the specified interface.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable an encapsulation type, use the **no arp arpa** or **no arp snap** interface configuration command.

Enable Proxy ARP

By default, the switch uses proxy ARP to help hosts learn MAC addresses of hosts on other networks or subnets.

Beginning in privileged EXEC mode, follow these steps to enable proxy ARP if it has been disabled:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode, and specify the Layer 3 interface to configure.
Step 3	**ip proxy-arp**	Enable proxy ARP on the interface.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show ip interface** [*interface-id*]	Verify the configuration on the interface or all interfaces.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable proxy ARP on the interface, use the **no ip proxy-arp** interface configuration command.

Routing Assistance When IP Routing is Disabled

These mechanisms allow the switch to learn about routes to other networks when it does not have IP routing enabled:

- Proxy ARP, page 37-13
- Default Gateway, page 37-13
- ICMP Router Discovery Protocol (IRDP), page 37-13

Proxy ARP

Proxy ARP, the most common method for learning about other routes, enables an Ethernet host with no routing information to communicate with hosts on other networks or subnets. The host assumes that all hosts are on the same local Ethernet and that they can use ARP to learn their MAC addresses. If a switch receives an ARP request for a host that is not on the same network as the sender, the switch evaluates whether it has the best route to that host. If it does, it sends an ARP reply packet with its own Ethernet MAC address. The host that sent the request then sends the packet to the switch, which forwards it to the intended host. Proxy ARP treats all networks as if they are local and performs ARP requests for every IP address.

Proxy ARP is enabled by default. To enable it after it has been disabled, see the "Enable Proxy ARP" section on page 37-12. Proxy ARP works as long as other routers support it.

Default Gateway

Another method for locating routes is to define a default router or default gateway. All nonlocal packets are sent to this router, which either routes them appropriately or returns an IP Control Message Protocol (ICMP) redirect message, identifying the local router that the host should use. The switch caches the redirect messages and forwards each packet as efficiently as possible. This method cannot detect when the default router has failed or is unavailable.

Beginning in privileged EXEC mode, follow these steps to define a default gateway (router) when IP routing is disabled:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip default-gateway** *ip-address*	Set up a default gateway (router).
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show ip redirects**	Display the address of the default gateway router to verify the setting.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no ip default-gateway** global configuration command to disable this function.

ICMP Router Discovery Protocol (IRDP)

Router discovery allows the switch to dynamically learn about routes to other networks by using ICMP Router Discovery Protocol (IRDP). Hosts use IRDP to locate routers. When operating as a client, the switch generates router discovery packets. When operating as a host, the switch receives router discovery packets. The switch can also receive Routing Information Protocol (RIP) routing updates and use this information to infer locations of routers. The switch does not actually store the routing tables sent by routing devices; it merely keeps track of which systems are sending the data. The advantage of using IRDP is that it allows each router to specify both a priority and the time after which a device is assumed to be down if no more packets are received.

Each device discovered becomes a candidate for the default router, and a new highest-priority router is selected when a higher priority router is discovered, when the current default router is declared down, or when a TCP connection is about to time out because of excessive retransmissions.

The only required task for IRDP routing on an interface is to enable IRDP processing on that interface. When enabled, the default parameters apply. You can change any of these parameters.

Beginning in privileged EXEC mode, follow these steps to enable and configure IRDP on an interface:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode, and specify the Layer 3 interface to configure.
Step 3	**ip irdp**	Enable IRDP processing on the interface.
Step 4	**ip irdp multicast**	(Optional) Send IRDP advertisements to the multicast address (224.0.0.1) instead of IP broadcasts. **Note** This command allows for compatibility with Sun Microsystems Solaris, which requires IRDP packets to be sent out as multicasts. Many implementations cannot receive these multicasts; ensure end-host ability before using this command.
Step 5	**ip irdp holdtime** *seconds*	(Optional) Set the IRDP period for which advertisements are valid. The default is three times the **maxadvertinterval** value. It must be greater than **maxadvertinterval** and cannot be greater than 9000 seconds. If you change the **maxadvertinterval** value, this value also changes.
Step 6	**ip irdp maxadvertinterval** *seconds*	(Optional) Set the IRDP maximum interval between advertisements. The default is 600 seconds.
Step 7	**ip irdp minadvertinterval** *seconds*	(Optional) Set the IRDP minimum interval between advertisements. The default is 0.75 times the **maxadvertinterval**. If you change the **maxadvertinterval**, this value changes to the new default (0.75 of **maxadvertinterval**).
Step 8	**ip irdp preference** *number*	(Optional) Set a device IRDP preference level. The allowed range is -2^{31} to 2^{31}. The default is 0. A higher value increases the router preference level.
Step 9	**ip irdp address** *address* [*number*]	(Optional) Specify an IRDP address and preference to proxy-advertise.
Step 10	**end**	Return to privileged EXEC mode.
Step 11	**show ip irdp**	Verify settings by displaying IRDP values.
Step 12	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

If you change the **maxadvertinterval** value, the **holdtime** and **minadvertinterval** values also change. It is important that you first change the **maxadvertinterval** value, before manually changing either the **holdtime** or **minadvertinterval** values.

Use the **no ip irdp** interface configuration command to disable IRDP routing.

Configuring Broadcast Packet Handling

After configuring an IP interface address, you can enable routing and configure one or more routing protocols, or you can configure the way the switch responds to network broadcasts. A broadcast is a data packet destined for all hosts on a physical network. The switch supports these kinds of broadcasting:

- A directed broadcast packet sent to a specific network or series of networks. A directed broadcast address includes the network or subnet fields.
- A flooded broadcast packet sent to every network.

Note You can also limit broadcast, unicast, and multicast traffic on Layer 2 interfaces by using the **storm-control** interface configuration command to set traffic suppression levels. For more information, see Chapter 25, "Configuring Port-Based Traffic Control."

Routers provide some protection from broadcast storms by limiting the extent to the local cable. Bridges (including intelligent bridges), because they are Layer 2 devices, forward broadcasts to all network segments, thus propagating broadcast storms. The best solution to the broadcast storm problem is to use a single broadcast address scheme on a network. In most IP implementations, you can set the broadcast address. Many implementations, including the one in the switch, support several addressing schemes for forwarding broadcast messages.

Perform the tasks in these sections to enable these schemes:

- Enabling Directed Broadcast-to-Physical Broadcast Translation, page 37-15
- Forwarding UDP Broadcast Packets and Protocols, page 37-16
- Establishing an IP Broadcast Address, page 37-17
- Flooding IP Broadcasts, page 37-18

Enabling Directed Broadcast-to-Physical Broadcast Translation

By default, IP directed broadcasts are dropped; they are not forwarded. Dropping IP-directed broadcasts makes routers less susceptible to denial-of-service attacks.

You can enable forwarding of IP-directed broadcasts on an interface when the broadcast becomes a physical (MAC-layer) broadcast. Only those protocols configured by using the **ip forward-protocol** global configuration command are forwarded.

You can specify an access control list (ACL) to control which broadcasts are forwarded. When an ACL is specified, only those IP packets permitted by the ACL can be translated from directed broadcasts to physical broadcasts. For more information on access lists, see Chapter 33, "Configuring Network Security with ACLs."

Beginning in privileged EXEC mode, follow these steps to enable forwarding of IP-directed broadcasts on an interface:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode, and specify the interface to configure.

	Command	Purpose
Step 3	**ip directed-broadcast** [*access-list-number*]	Enable directed broadcast-to-physical broadcast translation on the interface. You can include an ACL to control which broadcasts are forwarded. When an access list, only IP packets permitted by the access list can be translated **Note** The **ip directed-broadcast** interface configuration command can be configured on a VPN routing/forwarding(VRF) interface and is VRF-aware. Directed broadcast traffic is routed only within the VRF.
Step 4	**exit**	Return to global configuration mode.
Step 5	**ip forward-protocol** {**udp** [*port*] | **nd** | **sdns**}	Specify the protocols and ports the router uses when forwarding broadcast packets. • **udp**—Forward UPD datagrams. *port*: (Optional) Destination port that controls which UDP services are forwarded. • **nd**—Forward Network Disk datagrams. • **sdns**—Forward Secure Data Network Service (SDNS) datagrams
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show ip interface** [*interface-id*] or **show running-config**	Verify the configuration on the interface or all interfaces.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no ip directed-broadcast** interface configuration command to disable translation of directed broadcasts to physical broadcasts. Use the **no ip forward-protocol** global configuration command to remove a protocol or a port.

Forwarding UDP Broadcast Packets and Protocols

User Datagram Protocol (UDP) is an IP host-to-host layer protocol. UDP provides a connectionless session between two end systems and does not acknowledge received datagrams. Network hosts can use UDP broadcasts to find address, configuration, and name information. If such a host is on a network segment that does not include a server, UDP broadcasts might not be forwarded. However, you can configure an interface on a router to forward certain broadcast classes to a helper address. You can use more than one helper address per interface.

You can specify a UDP destination port to control which UDP services are forwarded. You can specify multiple UDP protocols. You can also specify the Network Disk Protocol (NDP), which is used by older diskless Sun workstations and the network security protocol SDNS.

By default, both UDP and NDP forwarding are enabled if a helper address has been defined for an interface. The description for the **ip forward-protocol** interface configuration command in the *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services, Release 12.2* lists the ports that are forwarded by default if you do not specify any UDP ports.

If you do not specify any UDP ports when you configure the forwarding of UDP broadcasts, you are configuring the router to act as a BOOTP forwarding agent. BOOTP packets carry DHCP information.

Beginning in privileged EXEC mode, follow these steps to enable forwarding of UDP broadcast packets on an interface and to specify the destination address:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode, and specify the Layer 3 interface to configure.
Step 3	**ip helper-address** *address*	Enable forwarding and specify the destination address for forwarding UDP broadcast packets, including BOOTP.
Step 4	**exit**	Return to global configuration mode.
Step 5	**ip forward-protocol** {**udp** [*port*] | **nd** | **sdns**}	Specify which protocols the router forwards when forwarding broadcast packets.
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show ip interface** [*interface-id*] or **show running-config**	Verify the configuration on the interface or all interfaces.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no ip helper-address** interface configuration command to disable the forwarding of broadcast packets to specific addresses. Use the **no ip forward-protocol** global configuration command to remove a protocol or port.

Establishing an IP Broadcast Address

The IP broadcast address (and the default) is an address consisting of all ones (255.255.255.255). However, the switch can be configured to generate any form of IP broadcast address.

Beginning in privileged EXEC mode, follow these steps to set the IP broadcast address on an interface:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode, and specify the interface to configure.
Step 3	**ip broadcast-address** *ip-address*	Enter a broadcast address different from the default, for example 128.1.255.255.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show ip interface** [*interface-id*]	Verify the broadcast address on the interface or all interfaces.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To restore the default IP broadcast address, use the **no ip broadcast-address** interface configuration command.

Flooding IP Broadcasts

You can allow IP broadcasts to be flooded throughout your internetwork in a controlled fashion by using the database created by the bridging STP. Using this feature also prevents loops. To support this capability, you must configure bridging on each interface that is to participate in the flooding. If bridging is not configured on an interface, it still can receive broadcasts. However, the interface never forwards broadcasts it receives, and the router never uses that interface to send broadcasts received on a different interface.

Packets that are forwarded to a single network address using the IP helper-address mechanism can be flooded. Only one copy of the packet is sent on each network segment.

To be considered for flooding, packets must meet these criteria. (Note that these are the same conditions to be met for packet forwarding when using IP helper addresses.)

- The packet must be a MAC-level broadcast.
- The packet must be an IP-level broadcast.
- The packet must be a TFTP, DNS, Time, NetBIOS, Network Disk, or BOOTP packet, or a UDP specified by the **ip forward-protocol udp** global configuration command.
- The time-to-live (TTL) value of the packet must be at least 2.

A flooded UDP datagram is given the destination address specified with the **ip broadcast-address** interface configuration command on the output interface. The destination address can be set to any address. Thus, the destination address might change as the datagram propagates through the network. The source address never changed. The TTL value decrements.

When a flooded UDP datagram is sent on an interface (and the destination address is possibly changed), the datagram is processed by the normal IP output routines and is, therefore, subject to ACLs, if they are present on the output interface.

Beginning in privileged EXEC mode, follow these steps to use the bridging spanning-tree database to flood UDP datagrams:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip forward-protocol spanning-tree**	Use the bridging spanning-tree database to flood UDP datagrams.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show running-config**	Verify your entry.
Step 5	**copy running-config startup-config**	(Optional) Save your entry in the configuration file.

Use the **no ip forward-protocol spanning-tree** global configuration command to disable the flooding of IP broadcasts.

In the switch, the majority of packets are forwarded in hardware; most packets do not go through the switch CPU. For those packets that do go to the CPU, you can speed up spanning tree-based UDP flooding by a factor of about four to five times by using turbo-flooding. This feature is supported over Ethernet interfaces configured for ARP encapsulation.

Beginning in privileged EXEC mode, follow these steps to increase spanning-tree-based flooding:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode
Step 2	**ip forward-protocol turbo-flood**	Use the spanning-tree database to speed up flooding of UDP datagrams.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show running-config**	Verify your entry.
Step 5	**copy running-config startup-config**	(Optional) Save your entry in the configuration file.

To disable this feature, use the **no ip forward-protocol turbo-flood** global configuration command.

Monitoring and Maintaining IP Addressing

When the contents of a particular cache, table, or database have become or are suspected to be invalid, you can remove all its contents by using the **clear** privileged EXEC commands. Table 37-2 lists the commands for clearing contents.

Table 37-2 Commands to Clear Caches, Tables, and Databases

Command	Purpose
clear arp-cache	Clear the IP ARP cache and the fast-switching cache.
clear host {*name* | *}}	Remove one or all entries from the hostname and the address cache.
clear ip route {*network* [*mask*] |*}	Remove one or more routes from the IP routing table.

You can display specific statistics, such as the contents of IP routing tables, caches, and databases; the reachability of nodes; and the routing path that packets are taking through the network. Table 37-3 lists the privileged EXEC commands for displaying IP statistics.

Table 37-3 Commands to Display Caches, Tables, and Databases

Command	Purpose
show arp	Display the entries in the ARP table.
show hosts	Display the default domain name, style of lookup service, name-server hosts, and the cached list of hostnames and addresses.
show ip aliases	Display IP addresses mapped to TCP ports (aliases).
show ip arp	Display the IP ARP cache.
show ip interface [*interface-id*]	Display the IP status of interfaces.
show ip irdp	Display IRDP values.
show ip masks *address*	Display the masks used for network addresses and the number of subnets using each mask.
show ip redirects	Display the address of a default gateway.
show ip route [*address* [*mask*]] | [*protocol*]	Display the current state of the routing table.
show ip route summary	Display the current state of the routing table in summary form.

Enabling IP Unicast Routing

By default, the switch is in Layer 2 switching mode, and IP routing is disabled. To use the Layer 3 capabilities of the switch, you must enable IP routing.

Beginning in privileged EXEC mode, follow these steps to enable IP routing:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip routing**	Enable IP routing.
Step 3	**router** *ip_routing_protocol*	Specify an IP routing protocol. This step might include other commands, such as specifying the networks to route with the **network** (RIP) router configuration command. For information on specific protocols, see sections later in this chapter and to the *Cisco IOS IP Configuration Guide, Release 12.2*. **Note** The IP base feature set supports only RIP as a routing protocol.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show running-config**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no ip routing** global configuration command to disable routing.

This example shows how to enable IP routing by using RIP as the routing protocol:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# router rip
Switch(config-router)# network 10.0.0.0
Switch(config-router)# end
```

You can now set parameters for the selected routing protocols as described in these sections:

- Configuring RIP, page 37-20
- Configuring OSPF, page 37-26
- Configuring EIGRP, page 37-36
- Configuring BGP, page 37-44
- Configuring Unicast Reverse Path Forwarding, page 37-80
- Configuring Protocol-Independent Features, page 37-80 (optional)

Configuring RIP

The Routing Information Protocol (RIP) is an interior gateway protocol (IGP) for use in small, homogeneous networks. It is a distance-vector routing protocol that uses broadcast User Datagram Protocol (UDP) data packets to exchange routing information. You can find detailed information about RIP in *IP Routing Fundamentals*, published by Cisco Press.

Note RIP is the only routing protocol supported by the IP base feature set; other routing protocols require the switch or the stack master to be running the IP services feature set.

Using RIP, the switch sends routing information updates (advertisements) every 30 seconds. If a router does not receive an update from another router for 180 seconds or more, it marks the routes served by that router as unusable. If there is still no update after 240 seconds, the router removes all routing table entries for that router.

RIP uses hop counts to rate the value of different routes. The hop count is the number of routers that can be traversed in a route. The hop count range is 0 to 15. A directly connected network has a hop count of zero; a network with a hop count of 16 is unreachable. The small range makes RIP unsuitable for large networks.

If the router has a default network path, RIP advertises a route that links the router to the pseudonetwork 0.0.0.0. The 0.0.0.0 network does not exist: It is treated by RIP as a network to implement the default routing feature. The switch advertises the default network if a default was learned by RIP or if the router has a gateway of last resort and RIP is configured with a default metric. RIP sends updates to the interfaces in specified networks. If an interface's network is not specified, it is not advertised in any RIP update.

These sections contain this configuration information:

- Default RIP Configuration, page 37-21
- Configuring Basic RIP Parameters, page 37-22
- Configuring RIP Authentication, page 37-23
- Configuring Summary Addresses and Split Horizon, page 37-24

Default RIP Configuration

Table 37-4 shows the default RIP configuration.

Table 37-4 Default RIP Configuration

Feature	Default Setting
Auto summary	Enabled.
Default-information originate	Disabled.
Default metric	Built-in; automatic metric translations.
IP RIP authentication key-chain	No authentication. Authentication mode: clear text.
IP RIP receive version	According to the **version** router configuration command.
IP RIP send version	According to the **version** router configuration command.
IP RIP triggered	According to the **version** router configuration command.
IP split horizon	Varies with media.
Neighbor	None defined.
Network	None specified.
Offset list	Disabled.

Table 37-4 Default RIP Configuration (continued)

Feature	Default Setting
Output delay	0 milliseconds.
Timers basic	• Update: 30 seconds. • Invalid: 180 seconds. • Hold-down: 180 seconds. • Flush: 240 seconds.
Validate-update-source	Enabled.
Version	Receives RIP Version 1 and 2 packets; sends Version 1 packets.

Configuring Basic RIP Parameters

To configure RIP, you enable RIP routing for a network and optionally configure other parameters. RIP configuration commands are ignored on the switch until you configure the network number.

Beginning in privileged EXEC mode, follow these steps to enable and configure RIP:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip routing**	Enable IP routing. (Required only if IP routing is disabled.)
Step 3	**router rip**	Enable a RIP routing process, and enter router configuration mode.
Step 4	**network** *network number*	Associate a network with a RIP routing process. You can specify multiple **network** commands. RIP routing updates are sent and received through interfaces only on these networks. **Note** You must configure a network number for RIP commands to take effect.
Step 5	**neighbor** *ip-address*	(Optional) Define a neighboring router with which to exchange routing information. This step allows routing updates from RIP (normally a broadcast protocol) to reach nonbroadcast networks.
Step 6	**offset list** [*access-list number* | *name*] {**in** | **out**} *offset* [*type number*]	(Optional) Apply an offset list to routing metrics to increase incoming and outgoing metrics to routes learned through RIP. You can limit the offset list with an access list or an interface.
Step 7	**timers basic** *update invalid holddown flush*	(Optional) Adjust routing protocol timers. Valid ranges for all timers are 0 to 4294967295 seconds. • *update*—The time between sending routing updates. The default is 30 seconds. • *invalid*—The time after which a route is declared invalid. The default is 180 seconds. • *holddown*—The time until a route is removed from the routing table. The default is 180 seconds. • *flush*—The amount of time for which routing updates are postponed. The default is 240 seconds.

	Command	Purpose
Step 8	**version** {**1** | **2**}	(Optional) Configure the switch to receive and send only RIP Version 1 or RIP Version 2 packets. By default, the switch receives Version 1 and Version 2 but sends only Version 1. You can also use the interface commands **ip rip** {**send** | **receive**} **version** **1** | **2** | **1 2**} to control what versions are used for sending and receiving on interfaces.
Step 9	**no auto summary**	(Optional) Disable automatic summarization. By default, the switch summarizes subprefixes when crossing classful network boundaries. Disable summarization (only RIP Version 2) to advertise subnet and host routing information to classful network boundaries.
Step 10	**no validate-update-source**	(Optional) Disable validation of the source IP address of incoming RIP routing updates. By default, the switch validates the source IP address of incoming RIP routing updates and discards the update if the source address is not valid. Under normal circumstances, we do not recommend that you disable this feature. However, if you have a router that is off-network and you want to receive its updates, you can use this command.
Step 11	**output-delay** *delay*	(Optional) Add interpacket delay for RIP updates sent. By default, packets in a multiple-packet RIP update have no delay time added between packets. If you are sending packets to a lower-speed device, you can add an interpacket delay in the range of 8 to 50 milliseconds.
Step 12	**end**	Return to privileged EXEC mode.
Step 13	**show ip protocols**	Verify your entries.
Step 14	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To turn off the RIP routing process, use the **no router rip** global configuration command.

To display the parameters and current state of the active routing protocol process, use the **show ip protocols** privileged EXEC command. To display summary address entries in the RIP database, use the **show ip rip database** privileged EXEC command.

Configuring RIP Authentication

RIP Version 1 does not support authentication. If you are sending and receiving RIP Version 2 packets, you can enable RIP authentication on an interface. The key chain specifies the set of keys that can be used on the interface. If a key chain is not configured, no authentication is performed, not even the default. Therefore, you must also perform the tasks in the "Managing Authentication Keys" section on page 37-94.

The switch supports these modes of authentication on interfaces for which RIP authentication is enabled: plain text and MD5. The default is plain text.

Beginning in privileged EXEC mode, follow these steps to configure RIP authentication on an interface:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode, and specify the interface to configure.
Step 3	**ip rip authentication key-chain** *name-of-chain*	Enable RIP authentication.
Step 4	**ip rip authentication mode** [**text** | **md5**}	Configure the interface to use plain text authentication (the default) or MD5 digest authentication.
Step 5	**end**	Return to privileged EXEC mode.
Step 6	**show running-config interface** [*interface-id*]	Verify your entries.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To restore clear text authentication, use the **no ip rip authentication mode** interface configuration command. To prevent authentication, use the **no ip rip authentication key-chain** interface configuration command.

Configuring Summary Addresses and Split Horizon

Routers connected to broadcast-type IP networks and using distance-vector routing protocols normally use the split-horizon mechanism to reduce the possibility of routing loops. Split horizon blocks information about routes from being advertised by a router on any interface from which that information originated. This feature usually optimizes communication among multiple routers, especially when links are broken.

Note In general, we do not recommend disabling split horizon unless you are certain that your application requires it to properly advertise routes.

If you want to configure an interface running RIP to advertise a summarized local IP address pool on a network access server for dial-up clients, use the **ip summary-address rip** interface configuration command.

Beginning in privileged EXEC mode, follow these steps to set an interface to advertise a summarized local IP address and to disable split horizon on the interface:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode, and specify the Layer 3 interface to configure.
Step 3	**ip address** *ip-address subnet-mask*	Configure the IP address and the IP subnet.
Step 4	**ip summary-address rip** *ip address ip-network mask*	Configure the IP address to be summarized and the IP network mask.
Step 5	**no ip split horizon**	Disable split horizon on the interface.
Step 6	**end**	Return to privileged EXEC mode.

	Command	Purpose
Step 7	**show ip interface** *interface-id*	Verify your entries.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable IP summarization, use the **no ip summary-address rip** router configuration command.

In this example, the major net is 10.0.0.0. The summary address of 10.2.0.0 overrides the autosummary address of 10.0.0.0 so that 10.2.0.0 is advertised from interface Gigabit Ethernet port 2, and 10.0.0.0 is not advertised. In the example, if the interface is still in Layer 2 mode (the default), you must enter the **no switchport** interface configuration command before entering the **ip address** interface configuration command.

Note If split horizon is enabled, neither autosummary nor interface summary addresses (those configured with the **ip summary-address rip** router configuration command) are advertised.

```
Switch(config)# router rip
Switch(config-router)# interface gigabitethernet1/0/2
Switch(config-if)# ip address 10.1.5.1 255.255.255.0
Switch(config-if)# ip summary-address rip 10.2.0.0 255.255.0.0
Switch(config-if)# no ip split-horizon
Switch(config-if)# exit
Switch(config)# router rip
Switch(config-router)# network 10.0.0.0
Switch(config-router)# neighbor 2.2.2.2 peer-group mygroup
Switch(config-router)# end
```

Configuring Split Horizon

Routers connected to broadcast-type IP networks and using distance-vector routing protocols normally use the split-horizon mechanism to reduce the possibility of routing loops. Split horizon blocks information about routes from being advertised by a router on any interface from which that information originated. This feature can optimize communication among multiple routers, especially when links are broken.

Note In general, we do not recommend disabling split horizon unless you are certain that your application requires it to properly advertise routes.

Beginning in privileged EXEC mode, follow these steps to disable split horizon on the interface:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode, and specify the interface to configure.
Step 3	**ip address** *ip-address subnet-mask*	Configure the IP address and the IP subnet.
Step 4	**no ip split-horizon**	Disable split horizon on the interface.
Step 5	**end**	Return to privileged EXEC mode.

	Command	Purpose
Step 6	**show ip interface** *interface-id*	Verify your entries.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To enable the split horizon mechanism, use the **ip split-horizon** interface configuration command.

Configuring OSPF

This section briefly describes how to configure Open Shortest Path First (OSPF). For a complete description of the OSPF commands, see the "OSPF Commands" chapter of the *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.2*.

Note OSPF classifies different media into broadcast, nonbroadcast, and point-to-point networks. The switch supports broadcast (Ethernet, Token Ring, and FDDI) and point-to-point networks (Ethernet interfaces configured as point-to-point links).

OSPF is an Interior Gateway Protocol (IGP) designed expressly for IP networks, supporting IP subnetting and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending and receiving packets.

The Cisco implementation conforms to the OSPF Version 2 specifications with these key features:

- Definition of stub areas is supported.
- Routes learned through any IP routing protocol can be redistributed into another IP routing protocol. At the intradomain level, OSPF can import routes learned through EIGRP and RIP. OSPF routes can also be exported into RIP.
- Plain text and MD5 authentication among neighboring routers within an area is supported.
- Configurable routing interface parameters include interface output cost, retransmission interval, interface transmit delay, router priority, router dead and hello intervals, and authentication key.
- Virtual links are supported.
- Not-so-stubby-areas (NSSAs) per RFC 1587are supported.

OSPF typically requires coordination among many internal routers, *area border routers* (ABRs) connected to multiple areas, and *autonomous system boundary routers* (ASBRs). The minimum configuration would use all default parameter values, no authentication, and interfaces assigned to areas. If you customize your environment, you must ensure coordinated configuration of all routers.

These sections contain this configuration information:

Note To enable OSPF, the switch or stack master must be running the IP services feature set.

Default OSPF Configuration

Table 37-5 shows the default OSPF configuration.

Table 37-5 Default OSPF Configuration

Feature	Default Setting
Interface parameters	Cost: No default cost predefined. Resend interval: 5 seconds. Send delay: 1 second. Priority: 1. Hello interval: 10 seconds. Dead interval: 4 times the hello interval. No authentication. No password specified. MD5 authentication disabled.
Area	Authentication type: 0 (no authentication). Default cost: 1. Range: Disabled. Stub: No stub area defined. NSSA: No NSSA area defined.
Auto cost	100 Mb/s.
Default-information originate	Disabled. When enabled, the default metric setting is 10, and the external route type default is Type 2.
Default metric	Built-in, automatic metric translation, as appropriate for each routing protocol.
Distance OSPF	dist1 (all routes within an area): 110. dist2 (all routes from one area to another): 110. dist3 (routes from other routing domains): 110.
OSPF database filter	Disabled. All outgoing link-state advertisements (LSAs) are flooded to the interface.
IP OSPF name lookup	Disabled.
Log adjacency changes	Enabled.
Neighbor	None specified.
Neighbor database filter	Disabled. All outgoing LSAs are flooded to the neighbor.
Network area	Disabled.

Table 37-5 Default OSPF Configuration (continued)

Feature	Default Setting
NSF[1] awareness	Enabled[2]. Allows Layer 3 switches to continue forwarding packets from a neighboring NSF-capable router during hardware or software changes.
NSF capability	Disabled. **Note** The switch stack supports OSPF NSF-capable routing for IPv4.
Router ID	No OSPF routing process defined.
Summary address	Disabled.
Timers LSA group pacing	240 seconds.
Timers shortest path first (spf)	spf delay: 5 seconds. spf-holdtime: 10 seconds.
Virtual link	No area ID or router ID defined. Hello interval: 10 seconds. Resend interval: 5 seconds. Send delay: 1 second. Dead interval: 40 seconds. Authentication key: no key predefined. Message-digest key (MD5): no key predefined.

1. NSF = nonstop forwarding.
2. OSPF NSF awareness is enabled for IPv4 on switches running the IP services feature set.

OSPF Nonstop Forwarding

The switch or switch stack supports two levels of nonstop forwarding (NSF):

- OSPF NSF Awareness, page 37-28
- OSPF NSF Capability, page 37-29

OSPF NSF Awareness

The IP-services feature set supports OSPF NSF awareness for IPv4. When the neighboring router is NSF-capable, the Layer 3 switch continues to forward packets from the neighboring router during the interval between the primary route processor in a router failure and the take-over of the backup route processor, or while the primary route processor is manually reloaded for a nondisruptive software upgrade.

This feature cannot be disabled. For more information on this feature, see the “OSPF Nonstop Forwarding (NSF) Awareness” section of the *Cisco IOS IP Routing Protocols Configuration Guide, Release 12.4* at this URL:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a00804557a8.html

OSPF NSF Capability

The IP-services feature set also supports OSPF NSF-capable routing for IPv4 for better convergence and lower traffic loss following a stack-master change. When a stack-master change occurs in an OSPF NSF-capable stack, the new stack master must do two things to resynchronize its link-state database with its OSFP neighbors:

- Release the available OSPF neighbors on the network without resetting the neighbor relationship.
- Re-acquire the contents of the link-state database for the network.

After a stack-master change, the new master sends an OSPF NSF signal to neighboring NSF-aware devices. A device recognizes this signal to mean that it should not reset the neighbor relationship with the stack. As the NSF-capable stack master receives signals from other routes on the network, it begins to rebuild its neighbor list.

When the neighbor relationships are reestablished, the NSF-capable stack master resynchronizes its database with its NSF-aware neighbors, and routing information is exchanged between the OSPF neighbors. The new stack master uses this routing information to remove stale routes, to update the routing information database (RIB), and to update the forwarding information base (FIB) with the new information. The OSPF protocols then fully converge.

Note OSPF NSF requires that all neighbor networking devices be NSF-aware. If an NSF-capable router discovers non-NSF-aware neighbors on a network segment, it disables NSF capabilities for that segment. Other network segments where all devices are NSF-aware or NSF-capable continue to provide NSF capabilities.

Use the **nsf** OSPF routing configuration command to enable OSPF NSF routing. Use the **show ip ospf** privileged EXEC command to verify that it is enabled.

For more information about this feature, see the *Cisco Nonstop Forwarding Feature Overview* at this URL:
http://www.cisco.com/en/US/products/sw/iosswrel/ps1829/products_feature_guide09186a00800ab7fc.html

Note NSF is not supported on interfaces configured for Hot Standby Router Protocol (HSRP).

Configuring Basic OSPF Parameters

Enabling OSPF requires that you create an OSPF routing process, specify the range of IP addresses to be associated with the routing process, and assign area IDs to be associated with that range.

Beginning in privileged EXEC mode, follow these steps to enable OSPF:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router ospf** *process-id*	Enable OSPF routing, and enter router configuration mode. The process ID is an internal identification parameter that is locally assigned and can be any positive integer. Each OSPF routing process has a unique value.
Step 3	**nsf**	(Optional) Enable NSF operations for OSPF.

	Command	Purpose
Step 4	**network** *address wildcard-mask* **area** *area-id*	Define an interface on which OSPF runs and the area ID for that interface. You can use the wildcard-mask to use a single command to define more than one interface to be associated with a specific OSPF area. The area ID can be a decimal value or an IP address.
Step 5	**end**	Return to privileged EXEC mode.
Step 6	**show ip protocols**	Verify your entries.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To end an OSPF routing process, use the **no router ospf** *process-id* global configuration command.

This example shows how to configure an OSPF routing process and assign it a process ID of 109:

```
Switch(config)# router ospf 109
Switch(config-router)# network 131.108.0.0 255.255.255.0 area 24
```

Configuring OSPF Interfaces

You can use the **ip ospf** interface configuration commands to modify interface-specific OSPF parameters. You are not required to modify any of these parameters, but some interface parameters (hello interval, dead interval, and authentication key) must be consistent across all routers in an attached network. If you modify these parameters, be sure all routers in the network have compatible values.

Beginning in privileged EXEC mode, follow these steps to modify OSPF interface parameters:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode, and specify the Layer 3 interface to configure.
Step 3	**ip ospf** *cost*	(Optional) Specify the cost of sending a packet on the interface.
Step 4	**ip ospf retransmit-interval** *seconds*	(Optional) Specify the number of seconds between link-state advertisement transmissions. The range is 1 to 65535 seconds. The default is 5 seconds.
Step 5	**ip ospf transmit-delay** *seconds*	(Optional) Set the estimated number of seconds to wait before sending a link state update packet. The range is 1 to 65535 seconds. The default is 1 second.
Step 6	**ip ospf priority** *number*	(Optional) Set priority to help find the OSPF designated router for a network. The range is from 0 to 255. The default is 1.
Step 7	**ip ospf hello-interval** *seconds*	(Optional) Set the number of seconds between hello packets sent on an OSPF interface. The value must be the same for all nodes on a network. The range is 1 to 65535 seconds. The default is 10 seconds.
Step 8	**ip ospf dead-interval** *seconds*	(Optional) Set the number of seconds after the last device hello packet was seen before its neighbors declare the OSPF router to be down. The value must be the same for all nodes on a network. The range is 1 to 65535 seconds. The default is 4 times the hello interval.

	Command	Purpose
Step 9	**ip ospf authentication-key** *key*	(Optional) Assign a password to be used by neighboring OSPF routers. The password can be any string of keyboard-entered characters up to 8 bytes in length. All neighboring routers on the same network must have the same password to exchange OSPF information.
Step 10	**ip ospf message digest-key** *keyid* **md5** *key*	(Optional) Enable MDS authentication. • *keyid*—An identifier from 1 to 255. • *key*—An alphanumeric password of up to 16 bytes.
Step 11	**ip ospf database-filter all out**	(Optional) Block flooding of OSPF LSA packets to the interface. By default, OSPF floods new LSAs to all interfaces in the same area except to the interface on which the LSA arrives.
Step 12	**end**	Return to privileged EXEC mode.
Step 13	**show ip ospf interface** [*interface-name*]	Display OSPF-related interface information.
Step 14	**show ip ospf neighbor detail**	Display NSF awareness status of neighbor switch. The output matches one of these examples: • *Options is 0x52* *LLS Options is 0x1 (LR)* When both of these lines appear, the neighbor switch is NSF-aware. • *Options is 0x42*—This means the neighbor switch is not NSF-aware.
Step 15	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no** form of these commands to remove the configured parameter value or to return to the default value.

Configuring OSPF Area Parameters

You can optionally configure several OSPF area parameters. These parameters include authentication for password-based protection against unauthorized access to an area, stub areas, and not-so-stubby-areas (NSSAs). *Stub areas* are areas into which information on external routes is not sent. Instead, the area border router (ABR) generates a default external route into the stub area for destinations outside the autonomous system. An NSSA does not flood all LSAs from the core into the area, but can import autonomous-system external routes within the area by redistribution.

Route summarization consolidates advertised addresses into a single summary route to be advertised by other areas. If network numbers are contiguous, you can use the **area range** router configuration command to configure the ABR to advertise a summary route that covers all networks in the range.

Note The OSPF **area** router configuration commands are all optional.

Beginning in privileged EXEC mode, follow these steps to configure area parameters:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router ospf** *process-id*	Enable OSPF routing, and enter router configuration mode.
Step 3	**area** *area-id* **authentication**	(Optional) Allow password-based protection against unauthorized access to the identified area. The identifier can be either a decimal value or an IP address.
Step 4	**area** *area-id* **authentication message-digest**	(Optional) Enable MD5 authentication on the area.
Step 5	**area** *area-id* **stub** [**no-summary**]	(Optional) Define an area as a stub area. The **no-summary** keyword prevents an ABR from sending summary link advertisements into the stub area.
Step 6	**area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate**] [**no-summary**]	(Optional) Defines an area as a NSSA. Every router within the same area must agree that the area is an NSSA. Select one of these keywords: • **no-redistribution**—Select when the router is an NSSA ABR and you want the **redistribute** command to import routes into normal areas but not into the NSSA. • **default-information-originate**—Select an ABR to allow importing type 7 LSAs into the NSSA. • **no-redistribution**—Select to not send summary LSAs into the NSSA.
Step 7	**area** *area-id* **range** *address mask*	(Optional) Specify an address range for which a single route is advertised. Use this command only with area border routers.
Step 8	**end**	Return to privileged EXEC mode.
Step 9	**show ip ospf** [*process-id*]	Display information about the OSPF routing process in general or for a specific process ID to verify configuration.
	show ip ospf [*process-id* [*area-id*]] **database**	Display lists of information related to the OSPF database for a specific router.
Step 10	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no** form of these commands to remove the configured parameter value or to return to the default value.

Configuring Other OSPF Parameters

You can optionally configure other OSPF parameters from router configuration mode.

- Route summarization: When redistributing routes from other protocols, as described in the "Using Route Maps to Redistribute Routing Information" section on page 37-84, each route is advertised individually in an external LSA. To help decrease the size of the OSPF link state database, you can use the **summary-address** router configuration command to advertise a single router for all the redistributed routes included in a specified network address and mask.

- Virtual links: In OSPF, all areas must be connected to a backbone area. You can establish a virtual link in case of a backbone-continuity break by configuring two ABRs as endpoints of a virtual link. Configuration information includes the identity of the other virtual endpoint (the other ABR) and the nonbackbone link that the two routers have in common (the transit area). Virtual links cannot be configured through a stub area.
- Default route: When you specifically configure redistribution of routes into an OSPF routing domain, the route automatically becomes an autonomous system boundary router (ASBR). You can force the ASBR to generate a default route into the OSPF routing domain.
- Domain name server (DNS) names for use in all OSPF **show** privileged EXEC command displays makes it easier to identify a router than displaying it by router ID or neighbor ID.
- Default metrics: OSPF calculates the OSPF metric for an interface according to the bandwidth of the interface. The metric is calculated as *ref-bw* divided by bandwidth, where *ref* is 10 by default, and bandwidth (*bw*) is specified by the **bandwidth** interface configuration command. For multiple links with high bandwidth, you can specify a larger number to differentiate the cost on those links.
- Administrative distance is a rating of the trustworthiness of a routing information source, an integer between 0 and 255, with a higher value meaning a lower trust rating. An administrative distance of 255 means the routing information source cannot be trusted at all and should be ignored. OSPF uses three different administrative distances: routes within an area (intra-area), routes to another area (interarea), and routes from another routing domain learned through redistribution (external). You can change any of the distance values.
- Passive interfaces: Because interfaces between two devices on an Ethernet represent only one network segment, to prevent OSPF from sending hello packets for the sending interface, you must configure the sending device to be a passive interface. Both devices can identify each other through the hello packet for the receiving interface.
- Route calculation timers: You can configure the delay time between when OSPF receives a topology change and when it starts the shortest path first (SPF) calculation and the hold time between two SPF calculations.
- Log neighbor changes: You can configure the router to send a syslog message when an OSPF neighbor state changes, providing a high-level view of changes in the router.

Beginning in privileged EXEC mode, follow these steps to configure these OSPF parameters:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router ospf** *process-id*	Enable OSPF routing, and enter router configuration mode.
Step 3	**summary-address** *address mask*	(Optional) Specify an address and IP subnet mask for redistributed routes so that only one summary route is advertised.
Step 4	**area** *area-id* **virtual-link** router-id [**hello-interval** *seconds*] [**retransmit-interval** *seconds*] [**trans**] [[**authentication-key** *key*] | **message-digest-key** *keyid* **md5** *key*]]	(Optional) Establish a virtual link, and set its parameters. See the "Configuring OSPF Interfaces" section on page 37-30 for parameter definitions and Table 37-5 on page 37-27 for virtual link defaults.
Step 5	**default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type-value*] [**route-map** *map-name*]	(Optional) Force the ASBR to generate a default route into the OSPF routing domain. Parameters are all optional.
Step 6	**ip ospf name-lookup**	(Optional) Configure DNS name lookup. The default is disabled.

	Command	Purpose
Step 7	**ip auto-cost reference-bandwidth** *ref-bw*	(Optional) Specify an address range for which a single route is advertised. Use this command only with area border routers.
Step 8	**distance ospf** {[**inter-area** *dist1*] [**inter-area** *dist2*] [**external** *dist3*]}	(Optional) Change the OSPF distance values. The range is 1 to 255. The default distance for each type of route is 110.
Step 9	**passive-interface** *type number*	(Optional) Suppress the sending of hello packets through the specified interface.
Step 10	**timers throttle spf** *spf-delay spf-holdtime spf-wait*	(Optional) Configure route calculation timers. • *spf-delay*—Delay between receiving a change to SPF calculation. The range is from 1 to 600000 miliseconds. • *spf-holdtim*e—Delay between first and second SPF calculation. The range is from 1 to 600000 in milliseconds. • *spf-wait*—Maximum wait time in milliseconds for SPF calculations. The range is from 1 to 600000 in milliseconds.
Step 11	**ospf log-adj-changes**	(Optional) Send syslog message when a neighbor state changes.
Step 12	**end**	Return to privileged EXEC mode.
Step 13	**show ip ospf** [*process-id* [*area-id*]] **database**	Display lists of information related to the OSPF database for a specific router. For some of the keyword options, see the "Monitoring OSPF" section on page 37-35.
Step 14	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Changing LSA Group Pacing

The OSPF LSA group pacing feature allows the router to group OSPF LSAs and pace the refreshing, check-summing, and aging functions for more efficient router use. This feature is enabled by default with a 4-minute default pacing interval, and you will not usually need to modify this parameter. The optimum group pacing interval is inversely proportional to the number of LSAs that the router is refreshing, check-summing, and aging. For example, if you have approximately 10,000 LSAs in the database, decreasing the pacing interval would benefit you. If you have a very small database (40 to 100 LSAs), increasing the pacing interval to 10 to 20 minutes might benefit you slightly.

Beginning in privileged EXEC mode, follow these steps to configure OSPF LSA pacing:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router ospf** *process-id*	Enable OSPF routing, and enter router configuration mode.
Step 3	**timers lsa-group-pacing** *seconds*	Change the group pacing of LSAs.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show running-config**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default value, use the **no timers lsa-group-pacing** router configuration command.

Configuring a Loopback Interface

OSPF uses the highest IP address configured on the interfaces as its router ID. If this interface is down or removed, the OSPF process must recalculate a new router ID and resend all its routing information through its interfaces. If a loopback interface is configured with an IP address, OSPF uses this IP address as its router ID, even if other interfaces have higher IP addresses. Loopback interfaces never fail, which provides greater stability. OSPF automatically prefers a loopback interface over other interfaces, and it chooses the highest IP address among all loopback interfaces.

Beginning in privileged EXEC mode, follow these steps to configure a loopback interface:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface loopback 0**	Create a loopback interface, and enter interface configuration mode.
Step 3	**ip address** *address mask*	Assign an IP address to this interface.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show ip interface**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no interface loopback 0** global configuration command to disable the loopback interface.

Monitoring OSPF

You can display specific statistics such as the contents of IP routing tables, caches, and databases.

Table 37-6 lists some of the privileged EXEC commands for displaying statistics. For more **show ip ospf database** privileged EXEC command options and for explanations of fields in the resulting display, see the *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.2*.

Table 37-6 Show IP OSPF Statistics Commands

Command	Purpose
show ip ospf [*process-id*]	Display general information about OSPF routing processes.
show ip ospf [*process-id*] **database** [**router**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**router**] [**self-originate**] **show ip ospf** [*process-id*] **database** [**router**] [**adv-router** [*ip-address*]] **show ip ospf** [*process-id*] **database** [**network**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**summary**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**asbr-summary**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**external**] [*link-state-id*] **show ip ospf** [*process-id area-id*] **database** [**database-summary**]	Display lists of information related to the OSPF database.
show ip ospf border-routes	Display the internal OSPF routing ABR and ASBR table entries.
show ip ospf interface [*interface-name*]	Display OSPF-related interface information.

Table 37-6 Show IP OSPF Statistics Commands

Command	Purpose
show ip ospf neighbor [*interface-name*] [*neighbor-id*] **detail**	Display OSPF interface neighbor information.
show ip ospf virtual-links	Display OSPF-related virtual links information.

Configuring EIGRP

Note If the switch is running the IP base image, you can configure *complete* EIGRP routing. However, the configuration is not implemented because the IP base image supports only EIGRP *stub routing*.

After you have entered the **eigrp stub** router configuration command, only the **eigrp stub connected summary** command takes effect. Although the CLI help might show the **receive-only** and **static** keywords, and you can enter these keywords, the switch running the IP base image always behaves as if the **connected** and **summary** keywords were configured.

Enhanced IGRP (EIGRP) is a Cisco-proprietary enhanced version of the IGRP. EIGRP uses the same distance vector algorithm and distance information as IGRP; however, the convergence properties and the operating efficiency of EIGRP are significantly improved.

The convergence technology employs an algorithm referred to as the diffusing update algorithm (DUAL), which guarantees loop-free operation at every instant throughout a route computation. All devices involved in a topology change can synchronize at the same time. Routers that are not affected by topology changes are not involved in recomputations.

IP EIGRP provides increased network width. With RIP, the largest possible width of your network is 15 hops. Because the EIGRP metric is large enough to support thousands of hops, the only barrier to network expansion is the transport-layer hop counter. EIGRP increments the transport control field only when an IP packet has traversed 15 routers and the next hop to the destination was learned through EIGRP. When a RIP route is used as the next hop to the destination, the transport control field increments as usual.

EIGRP offers these features:

- Fast convergence.
- Incremental updates when the state of a destination changes rather than sending the entire contents of the routing table, minimizing the bandwidth required for EIGRP packets.
- Less CPU usage because full update packets need not be processed each time they are received.
- Protocol-independent neighbor discovery mechanism to learn about neighboring routers.
- Variable-length subnet masks (VLSMs).
- Arbitrary route summarization.
- Scalable for large networks.

EIGRP has these four basic components:

- *Neighbor discovery and recovery* is the process that routers use to dynamically learn of other routers on their directly attached networks. Routers must also discover when their neighbors become unreachable or inoperative. Neighbor discovery and recovery is achieved by periodically sending small hello packets. As long as hello packets are received, the Cisco IOS software learns that a neighbor is alive and functioning. When this status is determined, the neighboring routers can exchange routing information.
- *The reliable transport protocol* is responsible for guaranteed, ordered delivery of EIGRP packets to all neighbors. It supports intermixed transmission of multicast and unicast packets. Some EIGRP packets must be sent reliably, and others need not be. For efficiency, reliability is provided only when necessary. For example, on a multi-access network with multicast capabilities (such as Ethernet), it is not necessary to send hellos reliably to all neighbors individually. Therefore, EIGRP sends a single multicast hello with an indication in the packet informing the receivers that the packet need not be acknowledged. Other types of packets (such as updates) require acknowledgment, which is shown in the packet. The reliable transport has a provision to send multicast packets quickly when there are unacknowledged packets pending. Doing so helps ensure that convergence time remains low in the presence of varying speed links.
- *The DUAL finite state machine* embodies the decision process for all route computations. It tracks all routes advertised by all neighbors. DUAL uses the distance information (known as a metric) to select efficient, loop-free paths. DUAL selects routes to be inserted into a routing table based on feasible successors. A successor is a neighboring router used for packet forwarding that has a least-cost path to a destination that is guaranteed not to be part of a routing loop. When there are no feasible successors, but there are neighbors advertising the destination, a recomputation occurs. This is the process whereby a new successor is determined. The amount of time it takes to recompute the route affects the convergence time. Recomputation is processor-intensive; it is advantageous to avoid recomputation if it is not necessary. When a topology change occurs, DUAL tests for feasible successors. If there are feasible successors, it uses any it finds to avoid unnecessary recomputation.
- The *protocol-dependent modules* are responsible for network layer protocol-specific tasks. An example is the IP EIGRP module, which is responsible for sending and receiving EIGRP packets that are encapsulated in IP. It is also responsible for parsing EIGRP packets and informing DUAL of the new information received. EIGRP uses DUAL for routing decisions, but the results are stored in the IP routing table. EIGRP is also responsible for redistributing routes learned by other IP routing protocols.

These sections contain this configuration information:

- Default EIGRP Configuration, page 37-38
- Configuring Basic EIGRP Parameters, page 37-40
- Configuring EIGRP Interfaces, page 37-41
- Configuring EIGRP Route Authentication, page 37-42
- EIGRP Stub Routing, page 37-43
- Monitoring and Maintaining EIGRP, page 37-44

Note To enable EIGRP, the switch or stack master must be running the IP services feature set.

Default EIGRP Configuration

Table 37-7 shows the default EIGRP configuration.

Table 37-7 Default EIGRP Configuration

Feature	Default Setting
Auto summary	Enabled. Subprefixes are summarized to the classful network boundary when crossing classful network boundaries.
Default-information	Exterior routes are accepted, and default information is passed between EIGRP processes during redistribution.
Default metric	Only connected routes and interface static routes can be redistributed without a default metric. The metric includes: • Bandwidth: 0 or greater kb/s. • Delay (tens of microseconds): 0 or any positive number that is a multiple of 39.1 nanoseconds. • Reliability: any number between 0 and 255 (255 means 100 percent reliability). • Loading: effective bandwidth as a number between 0 and 255 (255 is 100 percent loading). • MTU: maximum transmission unit size of the route in bytes. 0 or any positive integer.
Distance	Internal distance: 90. External distance: 170.
EIGRP log-neighbor changes	Disabled. No adjacency changes logged.
IP authentication key-chain	No authentication provided.
IP authentication mode	No authentication provided.
IP bandwidth-percent	50 percent.
IP hello interval	For low-speed nonbroadcast multiaccess (NBMA) networks: 60 seconds; all other networks: 5 seconds.
IP hold-time	For low-speed NBMA networks: 180 seconds; all other networks: 15 seconds.
IP split-horizon	Enabled.
IP summary address	No summary aggregate addresses are predefined.
Metric weights	tos: 0; k1 and k3: 1; k2, k4, and k5: 0
Network	None specified.
NSF[1] awareness	Enabled[2]. Allows Layer 3 switches to continue forwarding packets from a neighboring NSF-capable router during hardware or software changes.
NSF capability	Disabled. **Note** The switch supports EIGRP NSF-capable routing for IPv4.
Offset-list	Disabled.
Router EIGRP	Disabled.

Table 37-7 Default EIGRP Configuration (continued)

Feature	Default Setting
Set metric	No metric set in the route map.
Traffic-share	Distributed proportionately to the ratios of the metrics.
Variance	1 (equal-cost load-balancing).

1. NSF = nonstop forwarding
2. EIGRP NSF awareness is enabled for IPv4 on switches running the IP services feature set.

To create an EIGRP routing process, you must enable EIGRP and associate networks. EIGRP sends updates to the interfaces in the specified networks. If you do not specify an interface network, it is not advertised in any EIGRP update.

If you have routers on your network that are configured for IGRP, and you want to change to EIGRP, you must designate transition routers configured with *both* IGRP and EIGRP. In these cases, perform Steps 1 through 3 in the next section and also see the "Configuring Split Horizon" section on page 37-25. You must use the same autonomous-system number for routes so that they are automatically redistributed.

EIGRP Nonstop Forwarding

The switch stack supports two levels of EIGRP nonstop forwarding:

- EIGRP NSF Awareness, page 37-39
- EIGRP NSF Capability, page 37-39

EIGRP NSF Awareness

The IP-services feature set supports EIGRP NSF awareness for IPv4. When the neighboring router is NSF-capable, the Layer 3 switch continues to forward packets from the neighboring router during the interval between the primary route processor in a router failure and the backup RP take-over, or while you manually reload the primary route processor for a nondisruptive software upgrade.

This feature cannot be disabled. For more information on this feature, see the "EIGRP Nonstop Forwarding (NSF) Awareness" section of the *Cisco IOS IP Routing Protocols Configuration Guide, Release 12.4* at this URL:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a008045292972.html

EIGRP NSF Capability

The IP-services feature set also supports EIGRP NSF-capable routing for IPv4 for better convergence and lower traffic loss following a stack-master change. When an EIGRP NSF-capable stack master restarts or a new stack master starts and NSF restarts, the switch has no neighbors, and the topology table is empty. The switch must bring up the interfaces, re-acquire neighbors, and rebuild the topology and routing tables without interrupting the traffic directed to the switch stack. EIGRP peer routers maintain the routes learned from the new stack master and continue to forward traffic through the NSF restart process.

To prevent an adjacency reset by the neighbors, the new stack master uses a new restart bit in the EIGRP packet header. When the neighbor receives it, it synchronizes the stack in its peer list and maintains the adjacency with the stack. The neighbor then sends its topology table to the stack master with the restart bit set to show that it is NSF-aware and is aiding the new stack master.

If at least one of the stack peer neighbors is NSF-aware, the stack master receives updates and rebuilds its database. Each NSF-aware neighbor sends an end-of-table (EOT) marker in the last update packet to mark the end of the table content. The stack master recognizes the convergence when it receives the EOT marker and begins sending updates. When the stack master has received all EOT markers from its neighbors or when the NSF converge timer expires, EIGRP notifies the routing information database (RIB) of convergence and floods its topology table to all NSF-aware peers.

Note NSF is not supported on interfaces configured for Hot Standby Router Protocol (HSRP).

Use the **nsf** EIGRP routing configuration command to enable EIGRP NSF routing. Use the **show ip protocols** privileged EXEC command to verify that NSF is enabled. See the command reference for this release for information about the **nsf** command.

Configuring Basic EIGRP Parameters

Beginning in privileged EXEC mode, follow these steps to configure EIGRP. Configuring the routing process is required; other steps are optional:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router eigrp** *autonomous-system*	Enable an EIGRP routing process, and enter router configuration mode. The autonomous-system number identifies the routes to other EIGRP routers and tags routing information.
Step 3	**nsf**	(Optional) Enable EIGRP NSF. Enter this command on the stack master and on all of its peers.
Step 4	**network** *network-number*	Associate networks with an EIGRP routing process. EIGRP sends updates to the interfaces in the specified networks.
Step 5	**eigrp log-neighbor-changes**	(Optional) Enable logging of EIGRP neighbor changes to monitor routing system stability.
Step 6	**metric weights** *tos k1 k2 k3 k4 k5*	(Optional) Adjust the EIGRP metric. Although the defaults have been carefully set to provide excellent operation in most networks, you can adjust them. **Caution** Setting metrics is complex, and we do not recommend doing without guidance from an experienced network designer.
Step 7	**offset list [***access-list number* | *name*] {**in** | **out**} *offset* [*type number*]	(Optional) Apply an offset list to routing metrics to increase incoming and outgoing metrics to routes learned through EIGRP. You can limit the offset list with an access list or an interface.
Step 8	**no auto-summary**	(Optional) Disable automatic summarization of subnet routes into network-level routes.
Step 9	**ip summary-address eigrp** *autonomous-system-number address mask*	(Optional) Configure a summary aggregate.
Step 10	**end**	Return to privileged EXEC mode.

	Command	Purpose
Step 11	**show ip protocols**	Verify your entries.

Step 12	**show ip protocols**	Verify your entries.
		For NSF awareness, the output shows:
		`*** IP Routing is NSF aware ***`
		`EIGRP NSF enabled`

Step 13	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no** forms of these commands to disable the feature or to return the setting to the default value.

Configuring EIGRP Interfaces

Other optional EIGRP parameters can be configured on an interface basis.

Beginning in privileged EXEC mode, follow these steps to configure EIGRP interfaces:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode, and specify the Layer 3 interface to configure.
Step 3	**ip bandwidth-percent eigrp** *percent*	(Optional) Configure the percentage of bandwidth that can be used by EIGRP on an interface. The default is 50 percent.
Step 4	**ip summary-address eigrp** *autonomous-system-number address mask*	(Optional) Configure a summary aggregate address for a specified interface (not usually necessary if auto-summary is enabled).
Step 5	**ip hello-interval eigrp** *autonomous-system-number seconds*	(Optional) Change the hello-time interval for an EIGRP routing process. The range is 1 to 65535 seconds. The default is 60 seconds for low-speed NBMA networks and 5 seconds for all other networks.
Step 6	**ip hold-time eigrp** *autonomous-system-number seconds*	(Optional) Change the hold-time interval for an EIGRP routing process. The range is 1 to 65535 seconds. The default is 180 seconds for low-speed NBMA networks and 15 seconds for all other networks. **Caution** Do not adjust the hold time without consulting Cisco technical support.
Step 7	**no ip split-horizon eigrp** *autonomous-system-number*	(Optional) Disable split horizon to allow route information to be advertised by a router from any interface that originated that information.
Step 8	**end**	Return to privileged EXEC mode.
Step 9	**show ip eigrp interface**	Display which interfaces EIGRP is active on and information about EIGRP relating to those interfaces.
Step 10	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no** forms of these commands to disable the feature or to return the setting to the default value.

Configuring EIGRP Route Authentication

EIGRP route authentication provides MD5 authentication of routing updates from the EIGRP routing protocol to prevent the introduction of unauthorized or false routing messages from unapproved sources.

Beginning in privileged EXEC mode, follow these steps to enable authentication:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode, and specify the Layer 3 interface to configure.
Step 3	**ip authentication mode eigrp** *autonomous-system* **md5**	Enable MD5 authentication in IP EIGRP packets.
Step 4	**ip authentication key-chain eigrp** *autonomous-system key-chain*	Enable authentication of IP EIGRP packets.
Step 5	**exit**	Return to global configuration mode.
Step 6	**key chain** *name-of-chain*	Identify a key chain, and enter key-chain configuration mode. Match the name configured in Step 4.
Step 7	**key** *number*	In key-chain configuration mode, identify the key number.
Step 8	**key-string** *text*	In key-chain key configuration mode, identify the key string.
Step 9	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}	(Optional) Specify the time period during which the key can be received. The *start-time* and *end-time* syntax can be either *hh*:*mm*:*ss Month date year* or *hh*:*mm*:*ss date Month year*. The default is *forever* with the default *start-time* and the earliest acceptable date as January 1, 1993. The default *end-time* and **duration** is **infinite**.
Step 10	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}	(Optional) Specify the time period during which the key can be sent. The *start-time* and *end-time* syntax can be either *hh*:*mm*:*ss Month date year* or *hh*:*mm*:*ss date Month year*. The default is *forever* with the default *start-time* and the earliest acceptable date as January 1, 1993. The default *end-time* and **duration** is **infinite**.
Step 11	**end**	Return to privileged EXEC mode.
Step 12	**show key chain**	Display authentication key information.
Step 13	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no** forms of these commands to disable the feature or to return the setting to the default value.

EIGRP Stub Routing

Note The IP base feature set contains EIGRP stub routing capability, which only advertises connected or summary routes from the routing tables to other switches in the network. The switch uses EIGRP stub routing at the access layer to eliminate the need for other types of routing advertisements. For enhanced capability and complete EIGRP routing, the switch must be running the IP services feature set.
On a switch running the IP base feature set, if you try to configure multi-VRF-CE and EIGRP stub routing at the same time, the configuration is not allowed.

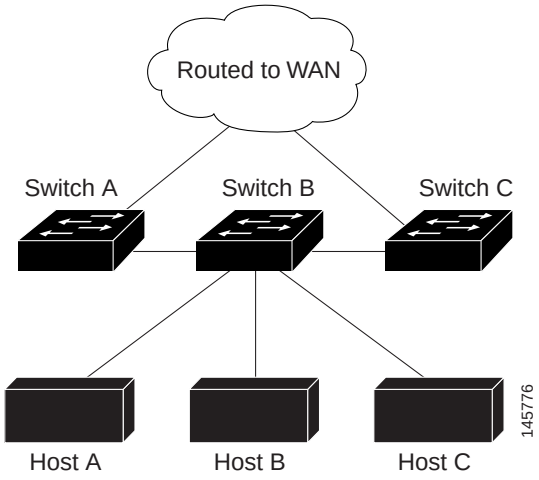
In a network using EIGRP stub routing, the only allowable route for IP traffic to the user is through a switch that is configured with EIGRP stub routing. The switch sends the routed traffic to interfaces that are configured as user interfaces or are connected to other devices.

When using EIGRP stub routing, you need to configure the distribution and remote routers to use EIGRP and to configure only the switch as a stub. Only specified routes are propagated from the switch. The switch responds to all queries for summaries, connected routes, and routing updates.

Any neighbor that receives a packet informing 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.

In Figure 37-4, Switch B is configured as an EIGRP stub router. Switches A and C are connected to the rest of the WAN. Switch B advertises connected, static, redistribution, and summary routes to Switches A and C. Switch B does not advertise any routes learned from switch A (and the reverse).

Figure 37-4 EIGRP Stub Router Configuration



For more information about EIGRP stub routing, see “Configuring EIGRP Stub Routing” section of the *Cisco IOS IP Configuration Guide, Volume 2 of 3: Routing Protocols, Release 12.2*.

Monitoring and Maintaining EIGRP

You can delete neighbors from the neighbor table. You can also display various EIGRP routing statistics. Table 37-8 lists the privileged EXEC commands for deleting neighbors and displaying statistics. For explanations of fields in the resulting display, see the *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.2*.

Table 37-8 IP EIGRP Clear and Show Commands

Command	Purpose
clear ip eigrp neighbors [*if-address* | *interface*]	Delete neighbors from the neighbor table.
show ip eigrp interface [*interface*] [*as number*]	Display information about interfaces configured for EIGRP.
show ip eigrp neighbors [*type-number*]	Display EIGRP discovered neighbors.
show ip eigrp topology [*autonomous-system-number*] | [[*ip-address*] *mask*]]	Display the EIGRP topology table for a given process.
show ip eigrp traffic [*autonomous-system-number*]	Display the number of packets sent and received for all or a specified EIGRP process.

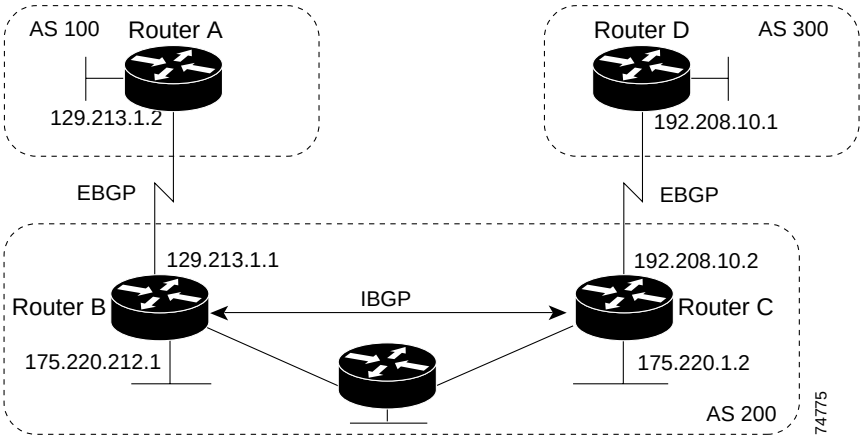
Configuring BGP

The Border Gateway Protocol (BGP) is an exterior gateway protocol for an interdomain routing system that guarantees the loop-free exchange of routing information between autonomous systems. Autonomous systems are made up of routers operating under the same administration and run Interior Gateway Protocols (IGPs), such as RIP or OSPF, within their boundaries and interconnecting by using an Exterior Gateway Protocol (EGP). BGP Version 4 is the standard EGP for interdomain routing in the Internet. You can find detailed information about BGP in *Internet Routing Architectures*, published by Cisco Press, and in the "Configuring BGP" chapter in the *Cisco IP and IP Routing Configuration Guide*.

For details about BGP commands and keywords, see the "IP Routing Protocols" part of the *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.2*. For a list of BGP commands that are visible but not supported by the switch, see Appendix C, "Unsupported Commands in Cisco IOS Release 12.2(46)SE."

Routers belonging to the same autonomous system and exchanging BGP updates run *internal BGP* (IBGP). Routers belonging to different autonomous systems and exchanging BGP updates run *external BGP* (EBGP). Most configuration commands are the same for configuring EBGP and IBGP. The difference is that the routing updates are exchanged either between autonomous systems (EBGP) or within an autonomous system (IBGP). Figure 37-5 shows a network that is running both EBGP and IBGP.

Figure 37-5 EBGP, IBGP, and Multiple Autonomous Systems



Before exchanging information with an external autonomous system, BGP ensures that networks in the autonomous system can be reached by defining internal BGP peering among routers and by redistributing BGP routing information to IGPs that run in the autonomous system, such as IGRP and OSPF.

Routers that run a BGP routing process are often referred to as BGP *speakers*. BGP uses the Transmission Control Protocol (TCP) as its transport protocol (specifically port 179). Two BGP speakers that have a TCP connection to each other are known as *peers* or *neighbors*. In Figure 37-5, Routers A and B are BGP peers, as are Routers B and C and Routers C and D. The routing information is a series of autonomous-system numbers that describe the full path to the destination network. BGP uses this information to construct a loop-free map of autonomous systems.

The network has these characteristics:

- Routers A and B are running EBGP, and Routers B and C are running IBGP. Note that the EBGP peers are directly connected and that the IBGP peers are not. As long as an IGP allows the two neighbors to reach one another, IBGP peers do not have to be directly connected.
- All BGP speakers within an autonomous system must establish a peer relationship. That is, the BGP speakers within an autonomous system must have a logical full mesh. However, BGP4 provides techniques to reduce the requirement for a logical full mesh: *confederations* and *route reflectors*.
- Autonomous system AS 200 is a *transit autonomous system* for AS 100 and AS 300—that is, AS 200 transfers packets between AS 100 and AS 300.

BGP peers first exchange their full BGP routing tables and then send only incremental updates. BGP peers also exchange keepalive messages (to ensure that the connection is up) and notification messages (in response to errors or special conditions).

In BGP, each route consists of a network number, a list of autonomous systems that information has passed through (the *autonomous system path*), and a list of other *path attributes*. The primary function of a BGP system is to exchange network reachability information, including information about the list of autonomous-system paths, with other BGP systems. This information determines autonomous-system connectivity, to prune routing loops, and to enforce autonomous-system-level policy decisions.

A router or switch running Cisco IOS does not select or use an IBGP route unless it has a route available to the next-hop router and it has received synchronization from an IGP (unless IGP synchronization is disabled). When multiple routes are available, BGP bases its path selection on *attribute* values. See the "Configuring BGP Decision Attributes" section on page 37-53 for information about BGP attributes.

BGP Version 4 supports classless interdomain routing (CIDR), so you can reduce the size of your routing tables by creating aggregate routes, resulting in *supernets*. CIDR eliminates the network classes within BGP and supports the advertising of IP prefixes.

These sections contain this configuration information:

For detailed descriptions of BGP configuration, see the "Configuring BGP" chapter in the "IP Routing Protocols" part of the Cisco IOS IP Configuration Guide, Release 12.2. For details about specific commands, see the *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.2*.

For a list of BGP commands that are visible but not supported by the switch, see Appendix C, "Unsupported Commands in Cisco IOS Release 12.2(46)SE."

Default BGP Configuration

Table 37-9 shows the default BGP configuration. For the defaults for all characteristics, see the specific commands in the *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.2*.

Table 37-9 Default BGP Configuration

Feature	Default Setting
Aggregate address	Disabled: None defined.
Autonomous-system path access list	None defined.
Auto summary	Enabled.
Best path	• The router considers *as-path* in choosing a route and does not compare similar routes from external BGP peers. • Compare router ID: Disabled.
BGP community list	• Number: None defined. When you permit a value for the community number, the list defaults to an implicit deny for everything else that has not been permitted. • Format: Cisco IOS default format (32-bit number).

Table 37-9 Default BGP Configuration (continued)

Feature	Default Setting
BGP confederation identifier/peers	• Identifier: None configured. • Peers: None identified.
BGP fast external fallover	Enabled.
BGP local preference	100. The range is 0 to 4294967295 with the higher value preferred.
BGP network	None specified; no backdoor route advertised.
BGP route dampening	Disabled by default. When enabled: • Half-life is 15 minutes. • Re-use is 750 (10-second increments). • Suppress is 2000 (10-second increments). • Max-suppress-time is 4 times half-life; 60 minutes.
BGP router ID	The IP address of a loopback interface if one is configured or the highest IP address configured for a physical interface on the router.
Default information originate (protocol or network redistribution)	Disabled.
Default metric	Built-in, automatic metric translations.
Distance	• External route administrative distance: 20 (acceptable values are from 1 to 255). • Internal route administrative distance: 200 (acceptable values are from 1 to 255). • Local route administrative distance: 200 (acceptable values are from 1 to 255).
Distribute list	• In (filter networks received in updates): Disabled. • Out (suppress networks from being advertised in updates): Disabled.
Internal route redistribution	Disabled.
IP prefix list	None defined.
Multi-exit discriminator (MED)	• Always compare: Disabled. Does not compare MEDs for paths from neighbors in different autonomous systems. • Best-path compare: Disabled. • MED missing as worst path: Disabled. • Deterministic MED comparison is disabled.

Table 37-9 Default BGP Configuration (continued)

Feature	Default Setting
Neighbor	• Advertisement interval: 30 seconds for external peers; 5 seconds for internal peers. • Change logging: Enabled. • Conditional advertisement: Disabled. • Default originate: No default route is sent to the neighbor. • Description: None. • Distribute list: None defined. • External BGP multihop: Only directly connected neighbors are allowed. • Filter list: None used. • Maximum number of prefixes received: No limit. • Next hop (router as next hop for BGP neighbor): Disabled. • Password: Disabled. • Peer group: None defined; no members assigned. • Prefix list: None specified. • Remote autonomous system (add entry to neighbor BGP table): No peers defined. • Private autonomous system number removal: Disabled. • Route maps: None applied to a peer. • Send community attributes: None sent to neighbors. • Shutdown or soft reconfiguration: Not enabled. • Timers: keepalive: 60 seconds; holdtime: 180 seconds. • Update source: Best local address. • Version: BGP Version 4. • Weight: Routes learned through BGP peer: 0; routes sourced by the local router: 32768.
NSF[1] awareness	Disabled[2]. Allows Layer 3 switches to continue forwarding packets from a neighboring NSF-capable router during hardware or software changes.
Route reflector	None configured.
Synchronization (BGP and IGP)	Enabled.
Table map update	Disabled.
Timers	Keepalive: 60 seconds; holdtime: 180 seconds.

1. NSF = nonstop forwarding.
2. NSF awareness can be enabled for IPv4 on switches with the IP services feature set by enabling graceful restart.

Nonstop Forwarding Awareness

The BGP NSF awareness feature is supported for IPv4 in the IP services feature set. To enable this feature with BGP routing, you need to enable graceful restart. When the neighboring router is NSF-capable, and this feature is enabled, the Layer 3 switch continues to forward packets from the

neighboring router during the interval between the primary route processor in a router failure and the backup route processor take-over, or while you manually reload the primary route processor for a nondisruptive software upgrade.

For more information, see the "BGP Nonstop Forwarding (NSF) Awareness" section of the *Cisco IOS IP Routing Protocols Configuration Guide, Release 12.4* at this URL:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a008045568e.html

Enabling BGP Routing

To enable BGP routing, you establish a BGP routing process and define the local network. Because BGP must fully recognize the relationships with its neighbors, you must also specify a BGP neighbor.

BGP supports two kinds of neighbors: internal and external. *Internal neighbors* are in the same autonomous system; *external neighbors* are in different autonomous systems. External neighbors are usually adjacent to each other and share a subnet, but internal neighbors can be anywhere in the same autonomous system.

The switch supports the use of private autonomous-system numbers, usually assigned by service providers and given to systems whose routes are not advertised to external neighbors. The private autonomous-system numbers are from 64512 to 65535. You can configure external neighbors to remove private autonomous-system numbers from the autonomous-system path by using the **neighbor remove-private-as** router configuration command. Then when an update is passed to an external neighbor, if the autonomous-system path includes any private autonomous-system numbers, these numbers are dropped.

If your autonomous system will be passing traffic through it from another autonomous system to a third autonomous system, it is important to be consistent about the routes it advertises. If BGP advertised a route before all routers in the network learned about the route through the IGP, the autonomous system might receive traffic that some routers could not yet route. Therefore, BGP must wait until the IGP has propagated information across the autonomous system so that BGP *synchronizes* with the IGP. Synchronization is enabled by default. If your autonomous system does not pass traffic from one autonomous system to another autonomous system, or if all routers in your autonomous systems are running BGP, you can disable synchronization, which allows your network to carry fewer routes in the IGP and allows BGP to converge more quickly.

Note To enable BGP, the switch or stack master must be running the IP services feature set.

Beginning in privileged EXEC mode, follow these steps to enable BGP routing, establish a BGP routing process, and specify a neighbor:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip routing**	Enable IP routing (required only if IP routing is disabled).
Step 3	**router bgp** *autonomous-system*	Enable a BGP routing process, assign it an autonomous-system number, and enter router configuration mode. The autonomous-system number can be from 1 to 65535, with 64512 to 65535 designated as private autonomous numbers.

	Command	Purpose
Step 4	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]	Configure a network as local to this autonomous system, and enter it in the BGP table.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *number*	Add an entry to the BGP neighbor table specifying that the neighbor identified by the IP address belongs to the specified autonomous system. EBGP neighbors are usually directly connected, and the IP address is the interface address at the other end of the connection. For IBGP, the IP address can be the address of any of the router interfaces.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **remove-private-as**	(Optional) Remove private autonomous-system numbers from the autonomous-system path in outbound routing updates.
Step 7	**no synchronization**	(Optional) Disable synchronization between BGP and an IGP.
Step 8	**no auto-summary**	(Optional) Disable automatic network summarization. By default, when a subnet is redistributed from an IGP into BGP, only the network route is inserted into the BGP table.
Step 9	**bgp fast-external-fallover**	(Optional) Automatically reset a BGP session when a link between external neighbors goes down. By default, the session is not immediately reset.
Step 10	**bgp graceful-restart**	(Optional) Enable NSF awareness on switch. By default, NSF awareness is disabled.
Step 11	**end**	Return to privileged EXEC mode.

Step 12	**show ip bgp network** *network-number*
	or
	show ip bgp neighbor
		Verify the configuration.
		Verify that NSF awareness (graceful restart) is enabled on the neighbor.
		If NSF awareness is enabled on the switch and the neighbor, this message appears:
		`Graceful Restart Capability: advertised and received`
		If NSF awareness is enabled on the switch, but not on the neighbor, this message appears:
		`Graceful Restart Capability: advertised`

Step 13	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no router bgp** *autonomous-system* global configuration command to remove a BGP autonomous system. Use the **no network** *network-number* router configuration command to remove the network from the BGP table. Use the **no neighbor** {*ip-address* | *peer-group-name*} **remote-as** *number* router configuration command to remove a neighbor. Use the **no neighbor** {*ip-address* | *peer-group-name*} **remove-private-as** router configuration command to include private autonomous-system numbers in updates to a neighbor. Use the **synchronization** router configuration command to re-enable synchronization.

These examples show how to configure BGP on the routers in Figure 37-5.

Router A:

```
Switch(config)# router bgp 100
Switch(config-router)# neighbor 129.213.1.1 remote-as 200
```

Router B:

```
Switch(config)# router bgp 200
Switch(config-router)# neighbor 129.213.1.2 remote-as 100
Switch(config-router)# neighbor 175.220.1.2 remote-as 200
```

Router C:

```
Switch(config)# router bgp 200
Switch(config-router)# neighbor 175.220.212.1 remote-as 200
Switch(config-router)# neighbor 192.208.10.1 remote-as 300
```

Router D:

```
Switch(config)# router bgp 300
Switch(config-router)# neighbor 192.208.10.2 remote-as 200
```

To verify that BGP peers are running, use the **show ip bgp neighbors** privileged EXEC command. This is the output of this command on Router A:

```
Switch# show ip bgp neighbors
BGP neighbor is 129.213.1.1, remote AS 200, external link
 BGP version 4, remote router ID 175.220.212.1
 BGP state = established, table version = 3, up for 0:10:59
 Last read 0:00:29, hold time is 180, keepalive interval is 60 seconds
 Minimum time between advertisement runs is 30 seconds
 Received 2828 messages, 0 notifications, 0 in queue
 Sent 2826 messages, 0 notifications, 0 in queue
 Connections established 11; dropped 10
```

Anything other than *BGP state = established* means that the peers are not running. The remote router ID is the highest IP address on that router (or the highest loopback interface). Each time the table is updated with new information, the table version number increments. A table version number that continually increments means that a route is flapping, causing continual routing updates.

For exterior protocols, a reference to an IP network from the **network** router configuration command controls only which networks are advertised. This is in contrast to Interior Gateway Protocols (IGPs), such as EIGRP, which also use the **network** command to specify where to send updates.

For detailed descriptions of BGP configuration, see the "IP Routing Protocols" part of the *Cisco IOS IP Configuration Guide, Release 12.2*. For details about specific commands, see the *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.2*. See Appendix C, "Unsupported Commands in Cisco IOS Release 12.2(46)SE," for a list of BGP commands that are visible but not supported by the switch.

Managing Routing Policy Changes

Routing policies for a peer include all the configurations that might affect inbound or outbound routing table updates. When you have defined two routers as BGP neighbors, they form a BGP connection and exchange routing information. If you later change a BGP filter, weight, distance, version, or timer, or make a similar configuration change, you must reset the BGP sessions so that the configuration changes take effect.

There are two types of reset, *hard reset* and *soft reset*. Cisco IOS Releases 12.1 and later support a soft reset without any prior configuration. To use a soft reset without preconfiguration, both BGP peers must support the soft-route refresh capability, which is advertised in the OPEN message sent when the peers establish a TCP session. A soft reset allows the dynamic exchange of route refresh requests and routing information between BGP routers and the subsequent re-advertisement of the respective outbound routing table.

- When soft reset generates inbound updates from a neighbor, it is called *dynamic inbound soft reset*.
- When soft reset sends a set of updates to a neighbor, it is called *outbound soft reset*.

A soft inbound reset causes the new inbound policy to take effect. A soft outbound reset causes the new local outbound policy to take effect without resetting the BGP session. As a new set of updates is sent during outbound policy reset, a new inbound policy can also take effect.

Table 37-10 lists the advantages and disadvantages hard reset and soft reset.

Table 37-10 Advantages and Disadvantages of Hard and Soft Resets

Type of Reset	Advantages	Disadvantages
Hard reset	No memory overhead	The prefixes in the BGP, IP, and FIB tables provided by the neighbor are lost. Not recommended.
Outbound soft reset	No configuration, no storing of routing table updates	Does not reset inbound routing table updates.
Dynamic inbound soft reset	Does not clear the BGP session and cache Does not require storing of routing table updates and has no memory overhead	Both BGP routers must support the route refresh capability (in Cisco IOS Release 12.1 and later).

Beginning in privileged EXEC mode, follow these steps to learn if a BGP peer supports the route-refresh capability and to reset the BGP session:

	Command	Purpose
Step 1	**show ip bgp neighbors**	Display whether a neighbor supports the route-refresh capability. When supported, this message appears for the router: *Received route refresh capability from peer.*
Step 2	**clear ip bgp** {***** | *address* | *peer-group-name*}	Reset the routing table on the specified connection. • Enter an asterisk (*) to reset all connections. • Enter an IP *address* to reset a specific connection. • Enter a peer group name to reset the peer group.
Step 3	**clear ip bgp** {***** | *address* | *peer-group-name*} **soft out**	(Optional) Perform an outbound soft reset to reset the inbound routing table on the specified connection. Use this command if route refresh is supported. • Enter an asterisk (*) to reset all connections. • Enter an IP *address* to reset a specific connection. • Enter a peer group name to reset the peer group.
Step 4	**show ip bgp** **show ip bgp neighbors**	Verify the reset by reviewing the routing table information and BGP neighbor information.

Configuring BGP Decision Attributes

When a BGP speaker receives updates from multiple autonomous systems that describe different paths to the same destination, it must choose the single best path for reaching that destination. The selected path is then entered in the BGP routing table and propagated to its neighbors. The decision is based on the value of attributes that the update contains and other BGP-configurable factors.

When a BGP peer learns two EBGP paths for a prefix from a neighboring autonomous system, it chooses the best path and inserts that path in the IP routing table. If BGP multipath support is enabled and the EBGP paths are learned from the same neighboring autonomous systems, instead of a single best path, multiple paths are entered in the IP routing table. Then, during packet switching, per-packet or per-destination load-balancing is performed among the multiple paths. The **maximum-paths** router configuration command controls the number of paths allowed.

These factors summarize the order in which BGP evaluates the attributes for choosing the best path:

1. If the path specifies a next hop that is inaccessible, drop the update. The BGP next-hop attribute, automatically determined by the software, is the IP address of the next hop to be used to reach a destination. For EBGP, this is usually the IP address of the neighbor specified by the **neighbor remote-as** router configuration command. You can disable next-hop processing by using route maps or the **neighbor next-hop-self** router configuration command.
2. Prefer the path with the largest weight (a Cisco-proprietary parameter). The weight attribute is local to the router and not propagated in routing updates. By default, the weight attribute is 32768 for paths that the router originates and zero for other paths. Routes with the largest weight are preferred. You can use access lists, route maps, or the **neighbor weight** router configuration command to set weights.
3. Prefer the route with the highest local preference. Local preference is part of the routing update and is exchanged among routers in the same autonomous system. The default value of the local preference attribute is 100. You can set local preference by using the **bgp default local-preference** router configuration command or by using a route map.
4. Prefer the route that was originated by BGP running on the local router.
5. Prefer the route with the shortest autonomous-system path.
6. Prefer the route with the lowest origin type. An interior route or IGP is lower than a route learned by EGP, and an EGP-learned route is lower than one of unknown origin or learned in another way.
7. Prefer the route with the lowest multi-exit discriminator (MED) metric attribute if the neighboring autonomous system is the same for all routes considered. You can configure the MED by using route maps or by using the **default-metric** router configuration command. When an update is sent to an IBGP peer, the MED is included.
8. Prefer the external (EBGP) path over the internal (IBGP) path.
9. Prefer the route that can be reached through the closest IGP neighbor (the lowest IGP metric). This means that the router prefers the shortest internal path within the autonomous system to reach the destination (the shortest path to the BGP next hop).
10. If these conditions are all true, insert the route for this path into the IP routing table:
    - Both the best route and this route are external.
    - Both the best route and this route are from the same neighboring autonomous system.
    - maximum-paths is enabled.
11. If multipath is not enabled, prefer the route with the lowest IP address value for the BGP router ID. The router ID is usually the highest IP address on the router or the loopback (virtual) address, but might be implementation-specific.

Beginning in privileged EXEC mode, follow these steps to configure some decision attributes:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router bgp** *autonomous-system*	Enable a BGP routing process, assign it an autonomous-system number, and enter router configuration mode.
Step 3	**bgp best-path as-path ignore**	(Optional) Configure the router to ignore autonomous-system path length in selecting a route.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**	(Optional) Disable next-hop processing on BGP updates to a neighbor by entering a specific IP address to be used instead of the next-hop address.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **weight** *weight*	(Optional) Assign a weight to a neighbor connection. Values are from 0 to 65535; the largest weight is the preferred route. Routes learned through another BGP peer have a default weight of 0; routes sourced by the local router have a default weight of 32768.
Step 6	**default-metric** *number*	(Optional) Set a MED metric to set preferred paths to external neighbors. All routes without a MED are also set to this value. The range is 1 to 4294967295. The lowest value is the most desirable.
Step 7	**bgp bestpath med missing-as-worst**	(Optional) Configure the switch to consider a missing MED as having a value of infinity, making the path without a MED value the least desirable path.
Step 8	**bgp always-compare med**	(Optional) Configure the switch to compare MEDs for paths from neighbors in different autonomous systems. By default, MED comparison is only done among paths in the same autonomous system.
Step 9	**bgp bestpath med confed**	(Optional) Configure the switch to consider the MED in choosing a path from among those advertised by different subautonomous systems within a confederation.
Step 10	**bgp deterministic med**	(Optional) Configure the switch to consider the MED variable when choosing among routes advertised by different peers in the same autonomous system.
Step 11	**bgp default local-preference** *value*	(Optional) Change the default local-preference value. The range is 0 to 4294967295; the default value is 100. The highest local-preference value is preferred.
Step 12	**maximum-paths** *number*	(Optional) Configure the number of paths to be added to the IP routing table. The default is to only enter the best path in the routing table. The range is from 1 to 16. Having multiple paths allows load-balancing among the paths. (Although the switch software allows a maximum of 32 equal-cost routes, the switch hardware never uses more than 16 paths per route.)
Step 13	**end**	Return to privileged EXEC mode.

	Command	Purpose
Step 14	**show ip bgp** **show ip bgp neighbors**	Verify the reset by reviewing routing table information and BGP neighbor information.
Step 15	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no** form of each command to return to the default state.

Configuring BGP Filtering with Route Maps

Within BGP, route maps can control and modify routing information and t define the conditions by which routes are redistributed between routing domains. See the "Using Route Maps to Redistribute Routing Information" section on page 37-84 for more information about route maps. Each route map has a name that identifies the it (*map tag*) and an optional sequence number.

Beginning in privileged EXEC mode, follow these steps to use a route map to disable next-hop processing:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**route-map** *map-tag* [[**permit** | **deny**] | *sequence-number*]]	Create a route map, and enter route-map configuration mode.
Step 3	**set ip next-hop** *ip-address* [*...ip-address*] [**peer-address**]	(Optional) Set a route map to disable next-hop processing • In an inbound route map, set the next hop of matching routes as the neighbor peering address, overriding third-party next hops. • In an outbound route map of a BGP peer, set the next hop to the peering address of the local router, disabling the next-hop calculation.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show route-map** [*map-name*]	Display all route maps configured or only the one specified to verify configuration.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no route-map** *map-tag* command to delete the route map. Use the **no set ip next-hop** *ip-address* command to re-enable next-hop processing.

Configuring BGP Filtering by Neighbor

You can filter BGP advertisements by using autonomous-system-path filters, such as the **as-path access-list** global configuration command and the **neighbor filter-list** router configuration command. You can also use access lists with the **neighbor distribute-list** router configuration command. Distribute-list filters are applied to network numbers. See the "Controlling Advertising and Processing in Routing Updates" section on page 37-93 for information about the **distribute-list** command.

You can use route maps on a per-neighbor basis to filter updates and to modify various attributes. You can apply a route map to either inbound or outbound updates. Only the routes that pass the route map are sent or accepted in updates. On both inbound and outbound updates, matching is supported based on

autonomous-system path, community, and network numbers. Autonomous-system-path matching requires the **match as-path access-lis**t route-map command, community-based matching requires the **match community-list** route-map command, and network-based matching requires the **ip access-list** global configuration command.

Beginning in privileged EXEC mode, follow these steps to apply a per-neighbor route map:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router bgp** *autonomous-system*	Enable a BGP routing process, assign it an autonomous-system number, and enter router configuration mode.
Step 3	**neighbor** {*ip-address* | *peer-group name*} **distribute-list** {*access-list-number* | *name*} {**in** | **out**}	(Optional) Filter BGP routing updates to or from neighbors as specified in an access list. **Note** You can also use the **neighbor prefix-list** router configuration command to filter updates, but you cannot use both commands to configure the same BGP peer.
Step 4	**neighbor** {*ip-address* | *peer-group name*} **route-map** *map-tag* {**in** | **out**}	(Optional) Apply a route map to filter an incoming or outgoing route.
Step 5	**end**	Return to privileged EXEC mode.
Step 6	**show ip bgp neighbors**	Verify the configuration.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no neighbor distribute-list** command to remove the access list from the neighbor. Use the **no neighbor route-map** *map-tag* router configuration command to remove the route map from the neighbor.

Another filtering method is to specify an access list filter on both incoming and outbound updates, based on the BGP autonomous system paths. Each filter is an access list based on regular expressions. (See the "Regular Expressions" appendix in the *Cisco IOS Dial Technologies Command Reference, Release 12.2* for more information on forming regular expressions.) To use this method, define an autonomous-system-path access list, and apply it to updates to and from particular neighbors.

Beginning in privileged EXEC mode, follow these steps to configure BGP autonomous-system-path filtering:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip as-path access-list** *access-list-number* {**permit** | **deny**} *as-regular-expressions*	Define a BGP-related access list.
Step 3	**router bgp** *autonomous-system*	Enter BGP router configuration mode.
Step 4	**neighbor** {*ip-address* | *peer-group name*} **filter-list** {*access-list-number* | *name*} {**in** | **out** | **weight** *weight*}	Establish a BGP filter based on an access list.
Step 5	**end**	Return to privileged EXEC mode.
Step 6	**show ip bgp neighbors** [**paths** *regular-expression*]	Verify the configuration.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring Prefix Lists for BGP Filtering

You can use prefix lists as an alternative to access lists in many BGP route filtering commands, including the **neighbor distribute-list** router configuration command. The advantages of using prefix lists include performance improvements in loading and lookup of large lists, incremental update support, easier CLI configuration, and greater flexibility.

Prefix-list filtering matches the prefixes of routes with those listed in the prefix list, as when matching access lists. When there is a match, the route is used. Whether a prefix is permitted or denied is based upon these rules:

- An empty prefix list permits all prefixes.
- An implicit deny is assumed if a prefix does not match any entries in a prefix list.
- When multiple entries of a prefix list match a prefix, the sequence number of a prefix-list entry identifies the entry with the lowest sequence number.

By default, sequence numbers are generated automatically and incremented in units of five. If you disable the automatic generation of sequence numbers, you must specify the sequence number for each entry. You can specify sequence values in any increment. If you specify increments of one, you cannot insert additional entries into the list. If you choose very large increments, you might run out of values.

You do not need to specify a sequence number when removing a configuration entry. **Show** commands include the sequence numbers in their output.

Before using a prefix list in a command, you must set up the prefix list. Beginning in privileged EXEC mode, follow these steps to create a prefix list or to add an entry to a prefix list:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip prefix-list** *list-name* [**seq** *seq-value*] **deny** | **permit** *network*/*len* [**ge** *ge-value*] [**le** *le-value*]	Create a prefix list with an optional sequence number to **deny** or **permit** access for matching conditions. You must enter at least one **permit** or **deny** clause. • *network*/*len* is the network number and length (in bits) of the network mask. • (Optional) **ge** and **le** values specify the range of the prefix length to be matched.The specified *ge-value* and *le-value* must satisfy this condition: *len* < *ge-value* < *le-value* < 32.
Step 3	**ip prefix-list** *list-name* **seq** *seq-value* **deny** | **permit** *network*/*len* [**ge** *ge-value*] [**le** *le-value*]	(Optional) Add an entry to a prefix list, and assign a sequence number to the entry.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show ip prefix list** [**detail** | **summary**] *name* [*network*/*len*] [**seq** *seq-num*] [**longer**] [**first-match**]	Verify the configuration by displaying information about a prefix list or prefix-list entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To delete a prefix list and all its entries, use the **no ip prefix-list** *list-name* global configuration command. To delete an entry from a prefix list, use the **no ip prefix-list seq** *seq-value* global configuration command. To disable automatic generation of sequence numbers, use the **no ip prefix-list**

sequence number command; to reenable automatic generation, use the **ip prefix-list sequence number** command. To clear the hit-count table of prefix list entries, use the **clear ip prefix-list** privileged EXEC command.

Configuring BGP Community Filtering

One way that BGP controls the distribution of routing information is based on the value of the COMMUNITIES attribute. The attribute groups destinations into communities and applies routing decisions based on the communities. This method simplifies the configuration of a BGP speaker to control distribution of routing information.

A *community* is a group of destinations that share some common attribute. Each destination can belong to multiple communities. Autonomous-system administrators can define to which communities a destination belongs. By default, all destinations belong to the general Internet community. The community is identified by the COMMUNITIES attribute, an optional, transitive, global attribute in the numerical range from 1 to 4294967200. These are some predefined, well-known communities:

- **internet**—Advertise this route to the Internet community. All routers belong to it.
- **no-export**—Do not advertise this route to EBGP peers.
- **no-advertise**—Do not advertise this route to any peer (internal or external).
- **local-as**—Do not advertise this route to peers outside the local autonomous system.

Based on the community, you can control the routing information to accept, prefer, or distribute to other neighbors. A BGP speaker can set, append, or modify the community of a route when learning, advertising, or redistributing routes. When routes are aggregated, the resulting aggregate has a COMMUNITIES attribute that contains all communities from all the initial routes.

You can use community lists to create groups of communities to use in a match clause of a route map. As with an access list, a series of community lists can be created. Statements are evaluated until a match is found. As soon as one statement is met, the test stops.

To set the COMMUNITIES attribute and match clauses based on communities, see the **match community-list** and **set community** route-map configuration commands in the "Using Route Maps to Redistribute Routing Information" section on page 37-84.

By default, no COMMUNITIES attribute is sent to a neighbor. You can specify that the COMMUNITIES attribute be sent to the neighbor at an IP address by using the **neighbor send-community** router configuration command.

Beginning in privileged EXEC mode, follow these steps to create and to apply a community list:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip community-list** *community-list-number* {**permit** | **deny**} *community-number*	Create a community list, and assign it a number. • The *community-list-number* is an integer from 1 to 99 that identifies one or more permit or deny groups of communities. • The *community-number* is the number configured by a **set community** route-map configuration command.
Step 3	**router bgp** *autonomous-system*	Enter BGP router configuration mode.
Step 4	**neighbor** {*ip-address* | *peer-group name*} **send-community**	Specify that the COMMUNITIES attribute is sent to the neighbor at this IP address.

	Command	Purpose
Step 5	**set comm-list** *list-num* **delete**	(Optional) Remove communities from the community attribute of an inbound or outbound update that match a standard or extended community list specified by a route map.
Step 6	**exit**	Return to global configuration mode.
Step 7	**ip bgp-community new-format**	(Optional) Display and parse BGP communities in the format AA:NN. A BGP community appears in a two-part format 2 bytes long. The Cisco-default community format is NNAA. In the most recent RFC for BGP, a community takes the form AA:NN, where the first part is the autonomous-system number and the second part is a 2-byte number.
Step 8	**end**	Return to privileged EXEC mode.
Step 9	**show ip bgp community**	Verify the configuration.
Step 10	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring BGP Neighbors and Peer Groups

Often many BGP neighbors are configured with the same update policies (that is, the same outbound route maps, distribute lists, filter lists, update source, and so on). You can group neighbors with the same update policies into peer groups to simplify configuration and to make updating more efficient. When you have configured many peers, we recommend this approach.

To configure a BGP peer group, you create the peer group, assign options to the peer group, and add neighbors as peer group members. You configure the peer group by using the **neighbor** router configuration commands. By default, peer group members inherit all the configuration options of the peer group, including the remote-as (if configured), version, update-source, out-route-map, out-filter-list, out-dist-list, minimum-advertisement-interval, and next-hop-self. All peer group members also inherit changes made to the peer group. You can also configure members to override the options that do not affect outbound updates.

To assign configuration options to an individual neighbor, specify any of these router configuration commands by using the neighbor IP address. To assign the options to a peer group, specify any of the commands by using the peer-group name. You can disable a BGP peer or peer group without removing all the configuration information by using the **neighbor shutdown** router configuration command.

Beginning in privileged EXEC mode, use these commands to configure BGP peers:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router bgp** *autonomous-system*	Enter BGP router configuration mode.
Step 3	**neighbor** *peer-group-name* **peer-group**	Create a BGP peer group.
Step 4	**neighbor** *ip-address* **peer-group** *peer-group-name*	Make a BGP neighbor a member of the peer group.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *number*	Specify a BGP neighbor. If a peer group does not have a **remote-as** *number*, use this command to create peer groups containing EBGP neighbors. The range is 1 to 65535.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **description** *text*	(Optional) Associate a description with a neighbor.

	Command	Purpose
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **default-originate** [**route-map** *map-name*]	(Optional) Allow a BGP speaker (the local router) to send the default route 0.0.0.0 to a neighbor for use as a default route.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **send-community**	(Optional) Specify that the COMMUNITIES attribute is sent to the neighbor at this IP address.
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface*	(Optional) Allow internal BGP sessions to use any operational interface for TCP connections.
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop**	(Optional) Allow BGP sessions, even when the neighbor is not on a directly connected segment. The multihop session is not established if the only route to the multihop-peer address is the default route (0.0.0.0).
Step 11	**neighbor** {*ip-address* | *peer-group-name*} **local-as** *number*	(Optional) Specify an autonomous-system number to use as the local autonomous system. The range is 1 to 65535.
Step 12	**neighbor** {*ip-address* | *peer-group-name*} **advertisement-interval** *seconds*	(Optional) Set the minimum interval between sending BGP routing updates.
Step 13	**neighbor** {*ip-address* | *peer-group-name*} **maximum-prefix** *maximum* [*threshold*]	(Optional) Control how many prefixes can be received from a neighbor. The range is 1 to 4294967295. The *threshold* (optional) is the percentage of the maximum at which a warning message is generated. The default is 75 percent.
Step 14	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**	(Optional) Disable next-hop processing on the BGP updates to a neighbor.
Step 15	**neighbor** {*ip-address* | *peer-group-name*} **password** *string*	(Optional) Set MD5 authentication on a TCP connection to a BGP peer. The same password must be configured on both BGP peers, or the connection between them is not made.
Step 16	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}	(Optional) Apply a route map to incoming or outgoing routes.
Step 17	**neighbor** {*ip-address* | *peer-group-name*} **send-community**	(Optional) Specify that the COMMUNITIES attribute is sent to the neighbor at this IP address.
Step 18	**neighbor** {*ip-address* | *peer-group-name*} **timers** *keepalive holdtime*	(Optional) Set timers for the neighbor or peer group. • The *keepalive* interval is the time within which keepalive messages are sent to peers. The range is 1 to 4294967295 seconds; the default is 60. • The *holdtime* is the interval after which a peer is declared inactive after not receiving a keepalive message from it. The range is 1 to 4294967295 seconds; the default is 180.
Step 19	**neighbor** {*ip-address* | *peer-group-name*} **weight** *weight*	(Optional) Specify a weight for all routes from a neighbor.
Step 20	**neighbor** {*ip-address* | *peer-group-name*} **distribute-list** {*access-list-number* | *name*} {**in** | **out**}	(Optional) Filter BGP routing updates to or from neighbors, as specified in an access list.
Step 21	**neighbor** {*ip-address* | *peer-group-name*} **filter-list** *access-list-number* {**in** | **out** | **weight** *weight*}	(Optional) Establish a BGP filter.
Step 22	**neighbor** {*ip-address* | *peer-group-name*} **version** *value*	(Optional) Specify the BGP version to use when communicating with a neighbor.

	Command	Purpose
Step 23	**neighbor** {*ip-address* | *peer-group-name*} **soft-reconfiguration inbound**	(Optional) Configure the software to store received updates.
Step 24	**end**	Return to privileged EXEC mode.
Step 25	**show ip bgp neighbors**	Verify the configuration.
Step 26	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable an existing BGP neighbor or neighbor peer group, use the **neighbor shutdown** router configuration command. To enable a previously existing neighbor or neighbor peer group that had been disabled, use the **no neighbor shutdown** router configuration command.

Configuring Aggregate Addresses

Classless interdomain routing (CIDR) creates aggregate routes (or *supernets*) to minimize the size of routing tables. You can configure aggregate routes in BGP either by redistributing an aggregate route into BGP or by creating an aggregate entry in the BGP routing table. An aggregate address is added to the BGP table when there is at least one more specific entry in the BGP table.

Beginning in privileged EXEC mode, use these commands to create an aggregate address in the routing table:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router bgp** *autonomous-system*	Enter BGP router configuration mode.
Step 3	**aggregate-address** *address mask*	Create an aggregate entry in the BGP routing table. The aggregate route is advertised as coming from the autonomous system, and the atomic aggregate attribute is set to show that information might be missing.
Step 4	**aggregate-address** *address mask* **as-set**	(Optional) Generate autonomous-system set path information. The command creates an aggregate entry following the same rules as the previous command, but the advertised path is an AS_SET consisting of all elements contained in all paths. Do not use this keyword when aggregating many paths because this route must be continually withdrawn and updated.
Step 5	**aggregate-address** *address-mask* **summary-only**	(Optional) Advertise only summary addresses.
Step 6	**aggregate-address** *address mask* **suppress-map** *map-name*	(Optional) Suppress selected, more specific routes.
Step 7	**aggregate-address** *address mask* **advertise-map** *map-name*	(Optional) Generate an aggregate based on conditions specified by the route map.
Step 8	**aggregate-address** *address mask* **attribute-map** *map-name*	(Optional) Generate an aggregate with attributes specified in the route map.
Step 9	**end**	Return to privileged EXEC mode.
Step 10	**show ip bgp neighbors** [**advertised-routes**]	Verify the configuration.
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To delete an aggregate entry, use the **no aggregate-address** *address mask* router configuration command. To return options to the default values, use the command with keywords.

Configuring Routing Domain Confederations

You can reduce the IBGP mesh is to divide an autonomous system into multiple subautonomous systems and to group them into a single *confederation* that appears as a single autonomous system. Each autonomous system is fully meshed within itself and has a few connections to other autonomous systems in the same confederation. Even though the peers in different autonomous systems have EBGP sessions, they exchange routing information as if they were IBGP peers. Specifically, the next hop, MED, and local-preference information is preserved. You can then use a single IGP for all of the autonomous systems.

To configure a BGP confederation, you must specify a confederation identifier that acts as the autonomous-system number for the autonomous-system group.

Beginning in privileged EXEC mode, use these commands to configure a BGP confederation:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router bgp** *autonomous-system*	Enter BGP router configuration mode.
Step 3	**bgp confederation identifier** *autonomous-system*	Configure a BGP confederation identifier.
Step 4	**bgp confederation peers** *autonomous-system* [*autonomous-system autonomous-system* ...]	Specify the autonomous systems that belong to the confederation and that are treated as special EBGP peers.
Step 5	**end**	Return to privileged EXEC mode.
Step 6	**show ip bgp neighbor** **show ip bgp network**	Verify the configuration.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring BGP Route Reflectors

BGP requires that all of the IBGP speakers be fully meshed. When a router receives a route from an external neighbor, it must advertise it to all internal neighbors. To prevent a routing information loop, all IBPG speakers must be connected. The internal neighbors do not send routes learned from internal neighbors to other internal neighbors.

With route reflectors, all IBGP speakers do not need to be fully meshed. When you configure an internal BGP peer as a *route reflector*, it passes IBGP learned routes to a set of IBGP neighbors. The internal peers of the route reflector are divided into *client peers* and *nonclient peers (*all the other routers in the autonomous system). A route reflector reflects routes between these two groups. The route reflector and its client peers form a *cluster.* The nonclient peers must be fully meshed with each other, but the client peers need not be fully meshed. The clients in the cluster do not communicate with IBGP speakers outside their cluster.

When the route reflector receives an advertised route, it takes one of these actions, depending on the neighbor:

- A route from an external BGP speaker is advertised to all clients and nonclient peers.
- A route from a nonclient peer is advertised to all clients.

- A route from a client is advertised to all clients and nonclient peers. Hence, the clients need not be fully meshed.

Usually, a cluster of clients have a single route reflector, and the cluster is identified by the route reflector router ID. To increase redundancy and to avoid a single point of failure, a cluster can have more than one route reflector. In this case, you must configure all route reflectors in the cluster with the same 4-byte cluster ID so that a route reflector recognizes updates from route reflectors in the same cluster. All the route reflectors serving a cluster should be fully meshed and should have identical sets of client and nonclient peers.

Beginning in privileged EXEC mode, use these commands to configure a route reflector and clients:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router bgp** *autonomous-system*	Enter BGP router configuration mode.
Step 3	**neighbor** {*ip-address* | *peer-group-name*} **route-reflector-client**	Configure the local router as a BGP route reflector and the specified neighbor as a client.
Step 4	**bgp cluster-id** *cluster-id*	(Optional) Configure the cluster ID if the cluster has more than one route reflector.
Step 5	**no bgp client-to-client reflection**	(Optional) Disable client-to-client route reflection. By default, the routes from a route reflector client are reflected to other clients. However, if the clients are fully meshed, the route reflector does not need to reflect routes to clients.
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show ip bgp**	Verify the configuration. Display the originator ID and the cluster-list attributes.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring Route Dampening

Route flap dampening is a BGP feature to minimize the propagation of flapping routes across an internetwork. A route is flapping when it is repeatedly available, then unavailable, then available, then unavailable, and so on. When route dampening is enabled, a numeric *penalty* value is assigned to a route when it flaps. When the accumulated penalties for a route reach the configured limit, BGP suppresses advertisements of the route, even if the route is running. The *reuse limit* is a configurable value that is compared with the penalty. If the penalty is less than the reuse limit, a suppressed route that is up is advertised again.

Dampening is not applied to routes that are learned by IBGP. This policy prevents the IBGP peers from having a higher penalty for routes external to the autonomous system.

Beginning in privileged EXEC mode, use these commands to configure BGP route dampening:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router bgp** *autonomous-system*	Enter BGP router configuration mode.
Step 3	**bgp dampening**	Enable BGP route dampening.

	Command	Purpose
Step 4	**bgp dampening** *half-life reuse suppress max-suppress* [**route-map** *map*]	(Optional) Change the default values of route dampening factors.
Step 5	**end**	Return to privileged EXEC mode.
Step 6	**show ip bgp flap-statistics** [{**regexp** *regexp*} | {**filter-list** *list*} | {*address mask* [**longer-prefix**]}]	(Optional) Monitor the flaps of all paths that are flapping. The statistics are deleted when the route is not suppressed and is stable.
Step 7	**show ip bgp dampened-paths**	(Optional) Display the dampened routes, including the time remaining before they are suppressed.
Step 8	**clear ip bgp flap-statistics** [{**regexp** *regexp*} | {**filter-list** *list*} | {*address mask* [**longer-prefix**]}	(Optional) Clear BGP flap statistics to make it less likely that a route is dampened.
Step 9	**clear ip bgp dampening**	(Optional) Clear route dampening information, and unsuppress the suppressed routes.
Step 10	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable flap dampening, use the **no bgp dampening** router configuration command without keywords. To set dampening factors back to the default values, use the **no bgp dampening** router configuration command with values.

Monitoring and Maintaining BGP

You can remove all contents of a particular cache, table, or database. This might be necessary when the contents have become or are suspected to be invalid.

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

Table 37-8 lists the privileged EXEC commands for clearing and displaying BGP. For explanations of the display fields, see the *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.2*.

Table 37-11 IP BGP Clear and Show Commands

Command	Purpose
clear ip bgp *address*	Reset a particular BGP connection.
clear ip bgp *	Reset all BGP connections.
clear ip bgp peer-group *tag*	Remove all members of a BGP peer group.
show ip bgp *prefix*	Display peer groups and peers not in peer groups to which the prefix has been advertised. Also display prefix attributes such as the next hop and the local prefix.
show ip bgp cidr-only	Display all BGP routes that contain subnet and supernet network masks.
show ip bgp community [*community-number*] [**exact**]	Display routes that belong to the specified communities.
show ip bgp community-list *community-list-number* [**exact-match**]	Display routes that are permitted by the community list.

Table 37-11 IP BGP Clear and Show Commands (continued)

Command	Purpose
show ip bgp filter-list *access-list-number*	Display routes that are matched by the specified autonomous-system path access list.
show ip bgp inconsistent-as	Display the routes with inconsistent originating autonomous systems.
show ip bgp regexp *regular-expression*	Display the routes that have an autonomous-system path that matches the specified regular expression entered on the command line.
show ip bgp	Display the contents of the BGP routing table.
show ip bgp neighbors [*address*]	Display detailed information on the BGP and TCP connections to individual neighbors.
show ip bgp neighbors [*address*] [**advertised-routes** | **dampened-routes** | **flap-statistics** | **paths** *regular-expression* | **received-routes** | **routes**]	Display routes learned from a particular BGP neighbor.
show ip bgp paths	Display all BGP paths in the database.
show ip bgp peer-group [*tag*] [**summary**]	Display information about BGP peer groups.
show ip bgp summary	Display the status of all BGP connections.

You can also enable the logging of messages generated when a BGP neighbor resets, comes up, or goes down by using the **bgp log-neighbor changes** router configuration command.

Configuring Multi-VRF CE

Virtual Private Networks (VPNs) provide a secure way for customers to share bandwidth over an ISP-backbone network. A VPN is a collection of sites sharing a common routing table. A customer site is connected to the service-provider network by one or more interfaces, and the service provider associates each interface with a VPN routing table known as a VPN routing/forwarding (VRF) table.

The switch supports multiple VPN routing/forwarding (multi-VRF) instances in customer-edge (CE) devices (multi-VRF CE) when the switch is running the IP services or advanced IP services feature set. A service provider uses multi-VRF CE allows to support two or more VPNs with overlapping IP addresses.

Note The switch does not use Multiprotocol Label Switching (MPLS) to support VPNs. For information about MPLS VRF, see the *Cisco IOS Switching Services Configuration Guide, Release 12.2*.

These sections contain this information:

- Understanding Multi-VRF CE, page 37-66
- Default Multi-VRF CE Configuration, page 37-68
- Multi-VRF CE Configuration Guidelines, page 37-68
- Configuring VRFs, page 37-69
- Configuring VRF-Aware Services, page 37-70

Understanding Multi-VRF CE

Multi-VRF CE is a feature that allows a service provider to support two or more VPNs overlapping IP addresses among the VPNs. Multi-VRF CE uses input interfaces to distinguish routes for different VPNs and forms virtual-packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF. Interfaces in a VRF can be either physical, such as Ethernet ports, or logical, such as VLAN switch virtual interfaces (SVIs), but an interface cannot belong to more than one VRF at any time.

Note Multi-VRF CE interfaces must be Layer 3 interfaces.

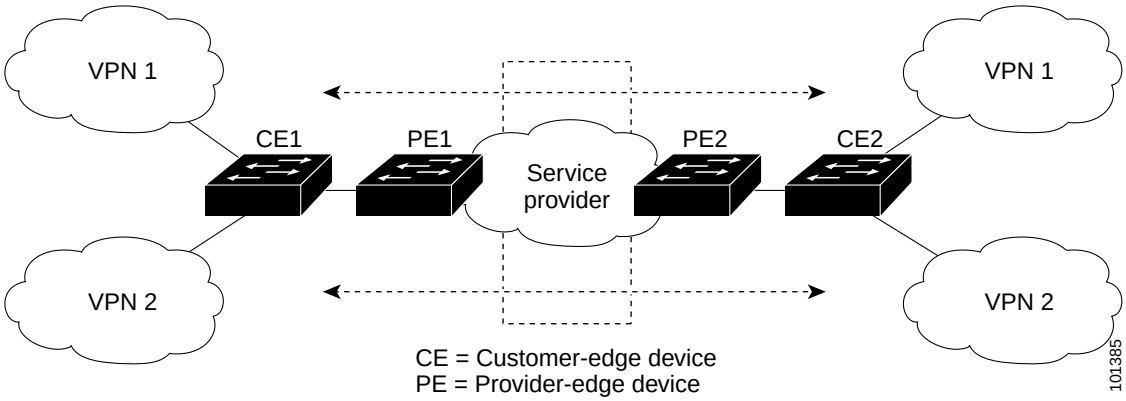
Multi-VRF CE includes these devices:

- Customer-edge (CE) devices provide customers access to the service-provider network over a data link to one or more provider edge routers. The CE device advertises the site local routes to the router and learns the remote VPN routes from it. A switch can be a CE.
- Provider-edge (PE) routers exchange routing information with CE devices by using static routing or a routing protocol such as BGP, RIPv2, OSPF, or EIGRP. The PE is only required to maintain VPN routes for those VPNs to which it is directly attached. The PE only needs to maintain all of the service-provider VPN routes. Each PE router maintains a VRF for each of its directly connected sites. Multiple interfaces on a PE router can be associated with a single VRF if all of these sites participate in the same VPN. Each VPN is mapped to a specified VRF. After learning local VPN routes from CEs, a PE router exchanges VPN routing information with other PE routers by using internal BGP (IBPG).
- Provider routers or core routers are any routers in the service provider network that are not attached to CE devices.

With multi-VRF CE, multiple customers can share one CE, and only one physical link is used between the CE and the PE. The shared CE maintains separate VRF tables for each customer and switches or routes packets for each customer based on its own routing table. Multi-VRF CE extends limited PE functionality to a CE device. It can then maintain separate VRF tables to extend the privacy and security of a VPN to the branch office.

Figure 37-6 is an example of switches as multiple virtual CEs. This scenario is suited for customers who have low bandwidth requirements for their VPN service, for example, small companies. In this case, multi-VRF CE support is required in the switches. Because multi-VRF CE is a Layer 3 feature, each interface in a VRF must be a Layer 3 interface.

Figure 37-6 Switches Acting as Multiple Virtual CEs



When the CE switch receives a command to add a Layer 3 interface to a VRF, it sets up the mapping between the VLAN ID and the policy label (PL) in multi-VRF-CE-related data structures and adds the VLAN ID and PL to the VLAN database.

When multi-VRF CE is configured, the Layer 3 forwarding table is virtually partitioned into two sections:

- The multi-VRF CE routing section contains the routes from different VPNs.
- The global routing section contains routes to non-VPN networks, such as the Internet.

VLAN IDs from different VRFs are mapped to different policy labels, which are used to distinguish the VRFs during processing. For each new VPN route learned, the Layer 3 setup function retrieves the policy label by using the VLAN ID of the ingress port and inserts the policy label and the new route to the multi-VRF CE routing section. If the packet is received from a routed port, the port internal VLAN ID number is used; if the packet is received from an SVI, the VLAN number is used.

This is the packet-forwarding process in a multi-VRF-CE-enabled network:

- When the switch receives a packet from a VPN, the switch looks up the routing table based on the input policy label number. When a route is found, the switch forwards the packet to the PE.
- When the ingress PE receives a packet from the CE, it performs a VRF lookup. When a route is found, the router adds a corresponding MPLS label to the packet and sends it to the MPLS network.
- When an egress PE receives a packet from the network, it strips the label and uses the label to identify the correct VPN routing table. Then it performs the normal route lookup. When a route is found, it forwards the packet to the correct adjacency.
- When a CE receives a packet from an egress PE, it uses the input policy label to look up the correct VPN routing table. If a route is found, the PE forwards the packet within the VPN.

To configure VRF, you create a VRF table and specify the Layer 3 interface associated with the VRF. Then you configure the routing protocols in the VPN and between the CE and the PE. BGP is the preferred routing protocol used to distribute VPN routing information across the provider's backbone. The multi-VRF CE network has these major components:

- VPN route target communities—lists of all other members of a VPN community. You need to configure VPN route targets for each VPN community member.
- Multiprotocol BGP peering of VPN community PE routers—propagates VRF reachability information to all members of a VPN community. You need to configure BGP peering in all PE routers within a VPN community.

- VPN forwarding—transports all traffic between all VPN community members across a VPN service-provider network.

Default Multi-VRF CE Configuration

Table 37-12 shows the default multi-VRF CE configuration.

Table 37-12 Default Multi-VRF CE Configuration

Feature	Default Setting
VRF	Disabled. No VRFs are defined.
Maps	No import maps, export maps, or route maps are defined.
VRF maximum routes	Fast Ethernet switches: 8000 Gigabit Ethernet switches: 12,000.
Forwarding table	The default for an interface is the global routing table.

Multi-VRF CE Configuration Guidelines

To use multi-VRF CE, you must have the IP services or advanced IP services feature set enabled on your switch.

These are considerations when configuring multi-VRF CE in your network:

- A switch with multi-VRF CE is shared by multiple customers, and each customer has its own routing table.
- Because customers use different VRF tables, the same IP addresses can be reused. Overlapped IP addresses are allowed in different VPNs.
- Multi-VRF CE lets multiple customers share the same physical link between the PE and the CE. Trunk ports with multiple VLANs separate packets among customers. Each customer has its own VLAN.
- Multi-VRF CE does not support all MPLS-VRF functionality. It does not support label exchange, LDP adjacency, or labeled packets.
- The PE router does not recognize a difference between using multi-VRF CE or using multiple CEs. In Figure 37-6, multiple virtual Layer 3 interfaces are connected to the multi-VRF CE device.
- The switch supports VRF over physical ports, VLAN SVIs, or a combination of both. The SVIs can be connected through an access port or a trunk port.
- A customer can use multiple VLANs if they do not overlap with those of other customers. A customer's VLANs are mapped to a specific routing table ID that identifies the appropriate routing tables stored on the switch.
- A switch supports 1 global network and up to 26 VRFs.
- Most routing protocols (BGP, OSPF, RIP, and static routing) can be used between the CE and the PE. However, we recommend using external BGP (EBGP) for these reasons:
  - BGP does not require multiple algorithms to communicate with multiple CEs.
  - BGP is designed for passing routing information between systems run by different administrations.
  - BGP simplifies the passing of attributes of the routes to the CE.

- Multi-VRF CE does not affect the packet switching rate.
- VPN multicast is not supported.
- You can configure 104 policies whether or not VRFs are configured on the switch or the switch stack.
- You can enable VRF on a private VLAN and the reverse.
- You cannot enable VRF when policy-based routing (PBR) is enabled on an interface and the reverse.
- You cannot enable VRF when Web Cache Communication Protocol (WCCP) is enabled on an interface and the reverse.

Configuring VRFs

Beginning in privileged EXEC mode, follow these steps to configure one or more VRFs. For complete syntax and usage information for the commands, see the switch command reference for this release and the *Cisco IOS Switching Services Command Reference, Release 12.2*.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip routing**	Enable IP routing.
Step 3	**ip vrf** *vrf-name*	Name the VRF, and enter VRF configuration mode.
Step 4	**rd** *route-distinguisher*	Create a VRF table by specifying a route distinguisher. Enter either an autonomous-system number and an arbitrary number (nnn:y) or an IP address and arbitrary number (A.B.C.D:y)
Step 5	**route-target** {**export** | **import** | **both**} *route-target-ext-community*	Create a list of import, export, or import and export route-target communities for the specified VRF. Enter either an autonomous-system number and an arbitrary number (nnn:y) or an IP address and an arbitrary number (A.B.C.D:y). The *route-target-ext-community* should be the same as the *route-distinguisher* entered in Step 4.
Step 6	**import map** *route-map*	(Optional) Associate a route map with the VRF.
Step 7	**interface** *interface-id*	Specify the Layer 3 interface to associate with the VRF, and enter interface configuration mode. The interface can be a routed port or an SVI.
Step 8	**ip vrf forwarding** *vrf-name*	Associate the VRF with the Layer 3 interface.
Step 9	**end**	Return to privileged EXEC mode.
Step 10	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*]	Verify the configuration. Display information about the configured VRFs.
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no ip vrf** *vrf-name* global configuration command to delete a VRF and to remove all interfaces from it. Use the **no ip vrf forwarding** interface configuration command to remove a specific interface from the VRF.

Configuring VRF-Aware Services

IP services can be configured on global interfaces that run within the global routing instance. IP services are enhanced to run on multiple routing instances; they are VRF-aware. Any configured VRF in the system can be specified for a VRF-aware service.

VRF-aware services are implemented in platform-independent modules. VRF means multiple routing instances in Cisco IOS. Each platform has its own limit on the number of VRFs it supports.

VRF-aware services have the following characteristics:

- The user can ping a host in a user-specified VRF.
- ARP entries are learned in separate VRFs. The user can display Address Resolution Protocol (ARP) entries for specific VRFs.

These services are VRF-aware:

- ARP
- Ping
- Simple Network Management Protocol (SNMP)
- Hot Standby Router Protocol (HSRP)
- Unicast Reverse Path Forwarding (uRPF)
- Syslog
- Traceroute
- FTP and TFTP

User Interface for ARP

Beginning in privileged EXEC mode, follow these steps to configure VRF-aware services for ARP. For complete syntax and usage information for the commands, refer to the switch command reference for this release and the *Cisco IOS Switching Services Command Reference, Release 12.2*.

Command	Purpose
show ip arp vrf *vrf-name*	Display the ARP table in the specified VRF.

User Interface for PING

Beginning in privileged EXEC mode, follow these steps to configure VRF-aware services for ping. For complete syntax and usage information for the commands, refer to the switch command reference for this release and the *Cisco IOS Switching Services Command Reference, Release 12.2*.

Command	Purpose
ping vrf *vrf-name* **ip-host**	Display the ARP table in the specified VRF.

User Interface for SNMP

Beginning in privileged EXEC mode, follow these steps to configure VRF-aware services for SNMP. For complete syntax and usage information for the commands, refer to the switch command reference for this release and the *Cisco IOS Switching Services Command Reference, Release 12.2*.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**snmp-server trap authentication vrf**	Enable SNMP traps for packets on a VRF.
Step 3	**snmp-server engineID remote** *host* **vrf** *vpn-instance engine-id string*	Configure a name for the remote SNMP engine on a switch.
Step 4	**snmp-server host** *host* **vrf** *vpn-instance* **traps** *community*	Specify the recipient of an SNMP trap operation, and specify the VRF table used for sending SNMP traps.
Step 5	**snmp-server host** *host* **vrf** *vpn-instance* **informs** *community*	Specify the recipient of an SNMP inform operation, and specify the VRF table used for sending SNMP informs.
Step 6	**snmp-server user** *user group* **remote** *host* **vrf** *vpn-instance security model*	Add a user to an SNMP group for a remote host on a VRF for SNMP access.
Step 7	**end**	Return to privileged EXEC mode.

User Interface for HSRP

HSRP support for VRFs ensures that HSRP virtual IP addresses are added to the correct IP routing table.

Beginning in privileged EXEC mode, follow these steps to configure VRF-aware services for HSRP. For complete syntax and usage information for the commands, refer to the switch command reference for this release and the *Cisco IOS Switching Services Command Reference, Release 12.2*.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode, and specify the Layer 3 interface to configure.
Step 3	**no switchport**	Remove the interface from Layer 2 configuration mode if it is a physical interface.
Step 4	**ip vrf forwarding** *vrf-name*	Configure VRF on the interface.
Step 5	**ip address** *ip-address*	Enter the IP address for the interface.
Step 6	**standby 1 ip** *ip-address*	Enable HSRP, and configure the virtual IP address.
Step 7	**end**	Return to privileged EXEC mode.

User Interface for uRPF

uRPF can be configured on an interface assigned to a VRF, and source lookup occurs in the VRF table.

Beginning in privileged EXEC mode, follow these steps to configure VRF-aware services for uRPF. For complete syntax and usage information for the commands, refer to the switch command reference for this release and the *Cisco IOS Switching Services Command Reference, Release 12.2*.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode, and specify the Layer 3 interface to configure.
Step 3	**no switchport**	Remove the interface from Layer 2 configuration mode if it is a physical interface.
Step 4	**ip vrf forwarding** *vrf-name*	Configure VRF on the interface.
Step 5	**ip address** *ip-address*	Enter the IP address for the interface.
Step 6	**ip verify unicast reverse-path**	Enable uRPF on the interface.
Step 7	**end**	Return to privileged EXEC mode.

User Interface for Syslog

Beginning in privileged EXEC mode, follow these steps to configure VRF-aware services for syslog. For complete syntax and usage information for the commands, refer to the switch command reference for this release and the *Cisco IOS Switching Services Command Reference, Release 12.2*.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**logging on**	Enable or temporarily disable logging of storage-router event message.
Step 3	**logging host** *ip-address* **vrf** *vrf-name*	Specify the host address of the syslog server where logging messages are to be sent.
Step 4	**logging buffered** *logging buffered size* **debugging**	Log messages to an internal buffer.
Step 5	**logging trap debugging**	Limit the logging messages sent to the syslog server.
Step 6	**logging facility** *facility*	Send system logging messages to a logging facility.
Step 7	**end**	Return to privileged EXEC mode.

User Interface for Traceroute

Beginning in privileged EXEC mode, follow these steps to configure VRF-aware services for traceroute. For complete syntax and usage information for the commands, refer to the switch command reference for this release and the *Cisco IOS Switching Services Command Reference, Release 12.2*.

Command	Purpose
traceroute vrf *vrf-name ipaddress*	Specify the name of a VPN VRF in which to find the destination address.

User Interface for FTP and TFTP

So that FTP and TFTP are VRF-aware, you must configure command-line interface (CLI) commands for FTP/TFTP. For example, if you want to use a VRF table that is attached to an interface, say E1/0, you need to configure the CLI **ip [t]ftp source-interface E1/0** to inform [t]ftp to use a specific routing table. In this example, the VRF table looks up the destination IP address. These changes are backward-compatible and do not affect existing behavior. That is, you can use the source-interface CLI to send packets out a particular interface even if no VRF is configured on that interface.

To specify the source IP address for FTP connections, use the **ip ftp source-interface** show mode command. To use the address of the interface where the connection is made, use the **no** form of this command.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip ftp source-interface** *interface-type interface-number*	Specify the source IP address for FTP connections.
Step 3	**end**	Return to privileged EXEC mode.

To specify an interface IP address as the source address for TFTP connections, use the **ip tftp source-interface** show mode command. To return to the default, use the **no** form of this command.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip tftp source-interface** *interface-type interface-number*	Specify the source IP address for TFTP connections.
Step 3	**end**	Return to privileged EXEC mode.

Configuring Multicast VRFs

Beginning in privileged EXEC mode, follow these steps to configure a multicast within a VRF table. For complete syntax and usage information for the commands, see the switch command reference for this release and the *Cisco IOS Switching Services Command Reference, Release 12.2*.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip routing**	Enable IP routing mode.
Step 3	**ip vrf** *vrf-name*	Name the VRF, and enter VRF configuration mode.
Step 4	**rd** *route-distinguisher*	Create a VRF table by specifying a route distinguisher. Enter either an autonomous-system number and an arbitrary number (nnn:y) or an IP address and an arbitrary number (A.B.C.D:y)

	Command	Purpose
Step 5	**route-target** {**export** | **import** | **both**} *route-target-ext-community*	Create a list of import, export, or import and export route target communities for the specified VRF. Enter either an autonomous-system number and an arbitrary number (nnn:y) or an IP address and an arbitrary number (A.B.C.D:y). The *route-target-ext-community* should be the same as the *route-distinguisher* entered in Step 4.
Step 6	**import map** *route-map*	(Optional) Associate a route map with the VRF.
Step 7	**ip multicast-routing vrf** *vrf*-**name distributed**	(Optional) Enable global multicast routing for VRF table.
Step 8	**interface** *interface-id*	Specify the Layer 3 interface to be associated with the VRF, and enter interface configuration mode. The interface can be a routed port or an SVI.
Step 9	**ip vrf forwarding** *vrf-name*	Associate the VRF with the Layer 3 interface.
Step 10	**ip address** *ip-address* **mask**	Configure the IP address for the Layer 3 interface.
Step 11	**ip pim sparse-dense mode**	Enable PIM on the VRF-associated Layer 3 interface.
Step 12	**end**	Return to privileged EXEC mode.
Step 13	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*]	Verify the configuration. Display information about the configured VRFs.
Step 14	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

For more information about configuring a multicast within a Multi-VRF CE, see the *Cisco IOS IP Multicast Configuration Guide, Release 12.4*.

Configuring a VPN Routing Session

Routing within the VPN can be configured with any supported routing protocol (RIP, OSPF, EIGRP, or BGP) or with static routing. The configuration shown here is for OSPF, but the process is the same for other protocols.

Beginning in privileged EXEC mode, follow these steps to configure OSPF in the VPN:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router ospf** *process-id* **vrf** *vrf-name*	Enable OSPF routing, specify a VPN forwarding table, and enter router configuration mode.
Step 3	**log-adjacency-changes**	(Optional) Log changes in the adjacency state. This is the default state.
Step 4	**redistribute bgp** *autonomous-system-number* **subnets**	Set the switch to redistribute information from the BGP network to the OSPF network.
Step 5	**network** *network-number* **area** *area-id*	Define a network address and mask on which OSPF runs and the area ID for that network address.
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show ip ospf** *process-id*	Verify the configuration of the OSPF network.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no router ospf** *process-id* **vrf** *vrf-name* global configuration command to disassociate the VPN forwarding table from the OSPF routing process.

Configuring BGP PE to CE Routing Sessions

Beginning in privileged EXEC mode, follow these steps to configure a BGP PE to CE routing session:

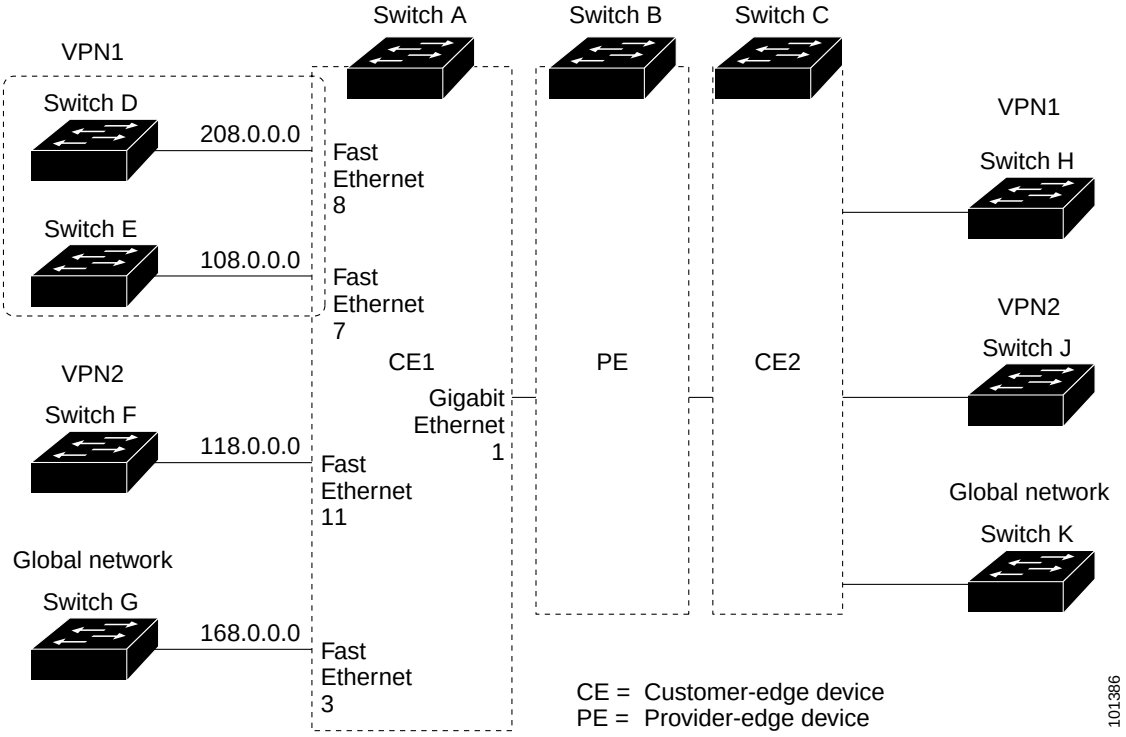
	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router bgp** *autonomous-system-number*	Configure the BGP routing process with the autonomous-system number passed to other BGP routers, and enter router configuration mode.
Step 3	**network** *network-number* **mask** *network-mask*	Specify a network and mask to advertise through BGP.
Step 4	**redistribute ospf** *process-id* **match internal**	Set the switch to redistribute OSPF internal routes.
Step 5	**network** *network-number* **area** *area-id*	Define a network address and mask on which OSPF runs and the area ID for that network address.
Step 6	**address-family ipv4 vrf** *vrf-name*	Define BGP parameters for PE to CE routing sessions, and enter VRF address-family mode.
Step 7	**neighbor** *address* **remote-as** *as-number*	Define a BGP session between PE and CE routers.
Step 8	**neighbor** *address* **activate**	Activate the advertisement of the IPv4 address family.
Step 9	**end**	Return to privileged EXEC mode.
Step 10	**show ip bgp** [**ipv4**] [**neighbors**]	Verify BGP configuration.
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no router bgp** *autonomous-system-number* global configuration command to delete the BGP routing process. Use the command with keywords to delete routing characteristics.

Multi-VRF CE Configuration Example

Figure 37-7 is a simplified example of the physical connections in a network similar to that in Figure 37-6. OSPF is the protocol used in VPN1, VPN2, and the global network. BGP is used in the CE-to-PE connections. The examples following the illustration show how to configure a switch as a CE Switch A, and the VRF configuration for customer switches D and F. Commands for configuring CE Switch C and the other customer switches are not included but would be similar. The example also includes commands for configuring traffic to Switch A for a Catalyst 6000 or Catalyst 6500 switch acting as a PE router.

Figure 37-7 Multi-VRF CE Configuration Example



Configuring Switch A

On Switch A, enable routing, and configure VRF.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# ip vrf v11
Switch(config-vrf)# rd 800:1
Switch(config-vrf)# route-target export 800:1
Switch(config-vrf)# route-target import 800:1
Switch(config-vrf)# exit
Switch(config)# ip vrf v12
Switch(config-vrf)# rd 800:2
Switch(config-vrf)# route-target export 800:2
Switch(config-vrf)# route-target import 800:2
Switch(config-vrf)# exit
```

Configure the loopback and physical interfaces on Switch A. Gigabit Ethernet port 1 is a trunk connection to the PE. Gigabit Ethernet ports 8 and 11 connect to VPNs:

```
Switch(config)# interface loopback1
Switch(config-if)# ip vrf forwarding v11
Switch(config-if)# ip address 8.8.1.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface loopback2
Switch(config-if)# ip vrf forwarding v12
Switch(config-if)# ip address 8.8.2.8 255.255.255.0
Switch(config-if)# exit
```

```
Switch(config)# interface gigabitethernet1/0/5
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface gigabitethernet1/0/8
Switch(config-if)# switchport access vlan 208
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface gigabitethernet1/0/11
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit
```

Configure the VLANs used on Switch A. VLAN 10 is used by VRF 11 between the CE and the PE. VLAN 20 is used by VRF 12 between the CE and the PE. VLANs 118 and 208 are used for the VPNs that include Switch F and Switch D, respectively:

```
Switch(config)# interface vlan10
Switch(config-if)# ip vrf forwarding v11
Switch(config-if)# ip address 38.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface vlan20
Switch(config-if)# ip vrf forwarding v12
Switch(config-if)# ip address 83.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface vlan118
Switch(config-if)# ip vrf forwarding v12
Switch(config-if)# ip address 118.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface vlan208
Switch(config-if)# ip vrf forwarding v11
Switch(config-if)# ip address 208.0.0.8 255.255.255.0
Switch(config-if)# exit
```

Configure OSPF routing in VPN1 and VPN2.

```
Switch(config)# router ospf 1 vrf vl1
Switch(config-router)# redistribute bgp 800 subnets
Switch(config-router)# network 208.0.0.0 0.0.0.255 area 0
Switch(config-router)# exit
Switch(config)# router ospf 2 vrf vl2
Switch(config-router)# redistribute bgp 800 subnets
Switch(config-router)# network 118.0.0.0 0.0.0.255 area 0
Switch(config-router)# exit
```

Configure BGP for CE-to-PE routing.

```
Switch(config)# router bgp 800
Switch(config-router)# address-family ipv4 vrf vl2
Switch(config-router-af)# redistribute ospf 2 match internal
Switch(config-router-af)# neighbor 83.0.0.3 remote-as 100
Switch(config-router-af)# neighbor 83.0.0.3 activate
Switch(config-router-af)# network 8.8.2.0 mask 255.255.255.0
Switch(config-router-af)# exit

Switch(config-router)# address-family ipv4 vrf vl1
Switch(config-router-af)# redistribute ospf 1 match internal
```

```
Switch(config-router-af)# neighbor 38.0.0.3 remote-as 100
Switch(config-router-af)# neighbor 38.0.0.3 activate
Switch(config-router-af)# network 8.8.1.0 mask 255.255.255.0
Switch(config-router-af)# end
```

Configuring Switch D

Switch D belongs to VPN 1. Configure the connection to Switch A by using these commands.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# interface gigabitethernet1/0/2
Switch(config-if)# no switchport
Switch(config-if)# ip address 208.0.0.20 255.255.255.0
Switch(config-if)# exit

Switch(config)# router ospf 101
Switch(config-router)# network 208.0.0.0 0.0.0.255 area 0
Switch(config-router)# end
```

Configuring Switch F

Switch F belongs to VPN 2. Configure the connection to Switch A by using these commands.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# interface gigabitethernet1/0/1
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface vlan118
Switch(config-if)# ip address 118.0.0.11 255.255.255.0
Switch(config-if)# exit

Switch(config)# router ospf 101
Switch(config-router)# network 118.0.0.0 0.0.0.255 area 0
Switch(config-router)# end
```

Configuring PE Switch B

When used on switch B (the PE router), these commands configure only the connections to the CE device, Switch A.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip vrf v1
Router(config-vrf)# rd 100:1
Router(config-vrf)# route-target export 100:1
Router(config-vrf)# route-target import 100:1
Router(config-vrf)# exit

Router(config)# ip vrf v2
Router(config-vrf)# rd 100:2
Router(config-vrf)# route-target export 100:2
Router(config-vrf)# route-target import 100:2
Router(config-vrf)# exit

Router(config)# ip cef
Router(config)# interface loopback1
```

```
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 3.3.1.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface loopback2
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 3.3.2.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface gigabitethernet1/1/0.10
Router(config-if)# encapsulation dot1q 10
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 38.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface gigabitethernet1/1/0.20
Router(config-if)# encapsulation dot1q 20
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 83.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# router bgp 100
Router(config-router)# address-family ipv4 vrf v2
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.2.0 mask 255.255.255.0
Router(config-router-af)# exit
Router(config-router)# address-family ipv4 vrf v1
Router(config-router-af)# neighbor 38.0.0.8 remote-as 800
Router(config-router-af)# neighbor 38.0.0.8 activate
Router(config-router-af)# network 3.3.1.0 mask 255.255.255.0
Router(config-router-af)# end
```

Displaying Multi-VRF CE Status

You can use the privileged EXEC commands in Table 37-13 to display information about multi-VRF CE configuration and status.

Table 37-13 Commands for Displaying Multi-VRF CE Information

Command	Purpose
show ip protocols vrf *vrf-name*	Display routing protocol information associated with a VRF.
show ip route vrf *vrf-name* [**connected**] [*protocol* [*as-number*]] [**list**] [**mobile**] [**odr**] [**profile**] [**static**] [**summary**] [**supernets-only**]	Display IP routing table information associated with a VRF.
show ip vrf [**brief** | **detail** | **interfaces**] [*vrf-name*]	Display information about the defined VRF instances.

For more information about the information in the displays, see the *Cisco IOS Switching Services Command Reference, Release 12.2*.

Configuring Unicast Reverse Path Forwarding

The unicast reverse path forwarding (uRPF) feature helps to mitigate problems that are caused by the introduction of malformed or forged (spoofed) IP source addresses into a network. uRPF discards IP packets without a verifiable IP source address. For example, a number of common denial-of-service (DoS) attacks, including Smurf and Tribal Flood Network (TFN), take advantage of forged or rapidly changing source-IP addresses to allow attackers to avoid efforts to locate or to filter the attacks. For Internet service providers (ISPs) that provide public access, uRPF deflects such attacks by forwarding only packets with valid source addresses and that are consistent with the IP routing table. This action protects the network of the ISP, its customer, and the rest of the Internet.

For detailed IP uRPF configuration information, see the *Other Security Features* chapter in the *Cisco IOS Security Configuration Guide, Release 12.2* at this URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_book09186a0080087df1.html

Configuring Protocol-Independent Features

This section describes how to configure IP routing protocol-independent features. These features are available on switches running the IP base or the IP services feature set. However, on the IP base feature set, protocol-related features are available only for RIP. For a complete description of the IP routing protocol-independent commands in this chapter, see the "IP Routing Protocol-Independent Commands" chapter of the *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.2* from the Cisco.com page under **Documentation** > **Cisco IOS Software** > **12.2 Mainline** > **Command References**.

These sections contain this configuration information:

Configuring Cisco Express Forwarding and Distributed Cisco Express Forwarding

Cisco Express Forwarding (CEF) is a Layer 3 IP switching technology used to optimize network performance. CEF implements an advanced IP look-up and forwarding algorithm to deliver maximum Layer 3 switching performance. CEF is less CPU-intensive than fast-switching route-caching, providing more CPU processing power dedicated to packet forwarding. In a switch stack, a stack member uses distributed CEF (dCEF) in the stack. On a standalone switch , the switch uses CEF. In dynamic networks, fast-switching cache entries are frequently invalidated because of routing changes, which causes traffic

to be process-switched using the routing table, instead of fast-switched using the route cache. CEF and dCEF use the Forwarding Information Base (FIB) lookup table for destination-based switching of IP packets.

The two main components in CEF and dCEF are the distributed FIB and the distributed adjacency tables.

- The FIB is similar to a routing table or information base and maintains a mirror image of the forwarding information in the IP routing table. When routing or topology changes occur in the network, the IP routing table is updated, and those changes are reflected in the FIB. The FIB maintains next-hop address information based on the information in the IP routing table. Because the FIB contains all known routes in the routing table, CEF eliminates route cache maintenance, is more efficient for switching traffic, and is not affected by traffic patterns.
- Nodes in the network are said to be adjacent if they can reach each other with a single hop across a link layer. CEF uses adjacency tables to prepend Layer 2 addressing information. The adjacency table maintains Layer 2 next-hop addresses for all FIB entries.

Because the switch or switch stack uses ASICs to achieve Gigabit-speed line rate IP traffic, CEF or dCEF forwarding applies only to the software-forwarding path, that is, traffic that is forwarded by the CPU.

The default configuration is CEF or dCEF enabled on all Layer 3 interfaces. Entering the **no ip route-cache cef** interface configuration command disables CEF for traffic that is being forwarded by software. This command does not affect the hardware forwarding path. You can re-enable CEF or dCEF by using the **ip cef** or **ip cef distributed** global configuration command. To enable CEF on an interface for the software-forwarding path, use the **ip route-cache cef** interface configuration command.

Caution Although the **no ip route-cache cef** interface configuration command to disable CEF on an interface is visible in the CLI, we strongly recommend that you do not disable CEF or dCEF on interfaces except for debugging purposes.

Beginning in privileged EXEC mode, follow these steps to enable CEF or dCEF globally and on an interface if it has been disabled:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip cef** or **ip cef distributed**	Enable CEF operation on a standalone switch, or enable dCEF operation on a switch stack.
Step 3	**interface** *interface-id*	Enter interface configuration mode, and specify the Layer 3 interface to configure.
Step 4	**ip route-cache cef**	Enable CEF on the interface.
Step 5	**end**	Return to privileged EXEC mode.
Step 6	**show ip cef**	Display the CEF status on all interfaces.

	Command	Purpose
Step 7	**show cef linecard** [**detail**] or **show cef linecard** [*stack-member-number*] [**detail**]	Display CEF-related interface information on a standalone switch, or display dCEF-related interface information for all switches in the stack or for the specified stack member. (Optional) For *stack-member-number*, specify the stack member.
Step 8	**show cef interface** [*interface-id*]	Display detailed CEF information for all interfaces or the specified interface.
Step 9	**show adjacency**	Display CEF adjacency table information.
Step 10	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring the Number of Equal-Cost Routing Paths

When a router has two or more routes to the same network with the same metrics, these routes can be thought of as having an equal cost. A *parallel path* provides another way to see occurrences of equal-cost routes in an IP routing table. A router with two or more equal-cost paths to a network can use them concurrently. Parallel paths provide redundancy in case of a circuit failure and also enable a router to load balance packets for more efficient use of available bandwidth. Equal-cost routes are supported across switches in a stack.

The router automatically learns about and configures equal-cost routes, but you can control the maximum number of parallel paths supported by an IP routing protocol in its routing table. Although the switch software allows a maximum of 32 equal-cost routes, the switch never uses more than 16 paths per route.

Beginning in privileged EXEC mode, follow these steps to change the maximum number of parallel paths in a routing table from the default:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router** {**bgp** | **rip** | **ospf** | **eigrp**}	Enter router configuration mode. A switch running the IP base feature set supports only the **rip** keyword.
Step 3	**maximum-paths** *maximum*	Set the maximum number of parallel paths for the protocol routing table. The range is from 1 to 16; the default is 4 for most IP routing protocols, but only 1 for BGP.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show ip protocols**	Verify the setting in the *Maximum path* field.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no maximum-paths** router configuration command to restore the default value.

Configuring Static Unicast Routes

Static unicast routes are user-defined routes that cause packets moving between a source and a destination to take a specified path. Static routes can be important if the router cannot dynamically build a route to a particular destination and are useful for specifying a gateway of last resort to which all unroutable packets are sent.

Beginning in privileged EXEC mode, follow these steps to configure a static route:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip route** *prefix mask* {*address* | *interface*} [*distance*]	Establish a static route.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show ip route**	Display the routing table to verify the configuration.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no ip route** *prefix mask* {*address* | *interface*} global configuration command to remove a static route.

The switch retains static routes until you remove them. However, you can override static routes with dynamic routing information by assigning administrative distance values. Each dynamic routing protocol has a default administrative distance, as listed in Table 37-14. If you want a static route to be overridden by information from a dynamic routing protocol, set the administrative distance of the static route higher than that of the dynamic protocol.

Table 37-14 Dynamic Routing Protocol Default Administrative Distances

Route Source	Default Distance
Connected interface	0
Static route	1
Enhanced IRGP summary route	5
External BGP	20
Internal Enhanced IGRP	90
IGRP	100
OSPF	110
Internal BGP	200
Unknown	225

Static routes that point to an interface are advertised through RIP, IGRP, and other dynamic routing protocols, whether or not static **redistribute** router configuration commands were specified for those routing protocols. These static routes are advertised because static routes that point to an interface are considered as connected in the routing table and hence lose their static nature. However, if you define a static route to an interface that is not one of the networks defined in a network command, no dynamic routing protocols advertise the route unless a **redistribute** static command is specified for these protocols.

When an interface goes down, all static routes through that interface are removed from the IP routing table. When the software can no longer find a valid next hop for the address specified as the forwarding-router address in a static route, the static route is also removed from the IP routing table.

Specifying Default Routes and Networks

A router might not be able to learn the routes to all other networks. To provide complete routing capability, you can use some routers as smart routers and give the remaining routers default routes to the smart router. (Smart routers have routing table information for the entire internetwork.) These default routes can be dynamically learned or can be configured in the individual routers. Most dynamic interior routing protocols can cause a smart router to generate dynamic default information that is then forwarded to other routers.

If a router has a directly connected interface to the specified default network, the dynamic routing protocols running on that device generate a default route. In RIP, it advertises the pseudonetwork 0.0.0.0.

A router that is generating the default route for a network also might need a default route of its own. One way a router can generate its own default router is to specify a static route to the network 0.0.0.0 through the appropriate device.

Beginning in privileged EXEC mode, follow these steps to define a default static route to a network:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip default-network** *network number*	Specify a default network.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show ip route**	Display the selected default route in the gateway of last resort output.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no ip default-network** *network number* global configuration command to remove the route.

When default information passes through a dynamic routing protocol, no further configuration is required. The system periodically scans its routing table to choose the best default network as its default route. IGRP networks might have several candidate networks for the system default. Cisco routers use administrative distance and metric information to set the default route or the gateway of last resort.

If dynamic default information is not being passed to the system, you can specify candidates for the default route with the **ip default-network** global configuration command. If this network appears in the routing table from any source, it is flagged as a possible choice for the default route. If the router has no interface on the default network, but does have a path to it, the network is considered as a possible candidate, and the gateway to the best default path becomes the gateway of last resort.

Using Route Maps to Redistribute Routing Information

The switch can run multiple routing protocols simultaneously, and it can redistribute information from one routing protocol to another. Information redistribution from one routing protocol to another applies to all supported IP-based routing protocols.

You can also conditionally control the redistribution of routes between routing domains by defining enhanced packet filters or route maps between the two domains. The **match** and **set** route-map configuration commands define the condition portion of a route map. The **match** command specifies that

a criterion must be matched. The **set** command specifies an action to be taken if the routing update meets the conditions defined by the match command. Although redistribution is a protocol-independent feature, some of the **match** and **set** route-map configuration commands are specific to a particular protocol.

One or more **match** commands and one or more **set** commands follow a **route-map** command. If there are no **match** commands, everything matches. If there are no **set** commands, only the match action occurs. Therefore, you need at least one **match** or **set** command.

You can also identify route-map statements as **permit** or **deny**. If the statement is marked as a deny, the packets meeting the match criteria are sent back through the normal forwarding channels (destination-based routing). If the statement is marked as a permit, set clauses are applied to packets meeting the match criteria. Packets that do not meet the match criteria are forwarded through the normal routing channel.

You can use the BGP route map **continue** clause to execute additional entries in a route map after an entry is executed with successful match and set clauses. You can use the **continue** clause to configure and organize more modular policy definitions so that specific policy configurations are not repeated within the same route map. The switch supports the **continue** clause for outbound policies. For more information about using the route map **continue** clause, see the BGP Route-Map Continue Support for an Outbound Policy feature guide for Cisco IOS Release 12.4(4)T at this URL:

http://www.cisco.com/en/US/products/ps6441/products_feature_guides_list.html

Note Although each of Steps 3 through 14 in the following section is optional, you must enter at least one **match** route-map configuration command and one **set** route-map configuration command.

Beginning in privileged EXEC mode, follow these steps to configure a route map for redistribution:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**route-map** *map-tag* [**permit** | **deny**] [*sequence number*]	Define any route maps used to control redistribution, and enter route-map configuration mode. *map-tag*—A meaningful name for the route map. The **redistribute** router configuration command uses this name to reference this route map. Multiple route maps can share the same map tag name. [**permit** | **deny**] (Optional)—If **permit** is specified and the match criteria are met for this route map, the route is redistributed as controlled by the set actions. If **deny** is specified, the route is not redistributed. *sequence number* (Optional)—Number that defines the position a new route map is to have in the list of route maps already configured with the same name.
Step 3	**match as-path** *path-list-number*	Match a BGP autonomous-system path access list.
Step 4	**match community-list** *community-list-number* [**exact**]	Match a BGP community list.
Step 5	**match ip address** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*]	Match a standard access list by specifying the name or number. It can be an integer from 1 to 199.

	Command	Purpose
Step 6	**match metric** *metric-value*	Match the specified route metric. The *metric-value* can be an EIGRP metric with a specified value from 0 to 4294967295.
Step 7	**match ip next-hop** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*]	Match a next-hop router address passed by one of the access lists specified (numbered from 1 to 199).
Step 8	**match tag** *tag value* [*...tag-value*]	Match the specified tag value in a list of one or more route tag values. Each can be an integer from 0 to 4294967295.
Step 9	**match interface** *type number* [*...type number*]	Match the specified next-hop route out one of the specified interfaces.
Step 10	**match ip route-source** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*]	Match the address specified by the specified advertised access lists.
Step 11	**match route-type** {**local** | **internal** | **external** [**type-1** | **type-2**]}	Match the specified **route-type**: • **local**—Locally generated BGP routes. • **internal**—OSPF intra-area and interarea routes or EIGRP internal routes. • **external**—OSPF external routes (Type 1 or Type 2) or EIGRP external routes.
Step 12	**set dampening** *halflife reuse suppress max-suppress-time*	Set BGP route dampening factors.
Step 13	**set local-preference** *value*	Assign a value to a local BGP path.
Step 14	**set origin** {**igp** | **egp** *as* | **incomplete**}	Set the BGP origin code.
Step 15	**set as-path** {**tag** | **prepend** *as-path-string*}	Modify the BGP autonomous system path.
Step 16	**set level** {**level-1** | **level-2** | **level-1-2** | **stub-area** | **backbone**}	Set the level for routes that are advertised into the specified area of the routing domain. The **stub-area** and **backbone** are OSPF NSSA and backbone areas.
Step 17	**set metric** *metric value*	Set the metric value to give the redistributed routes (only for EIGRP). The *metric value* is an integer from –294967295 to 294967295.

	Command	Purpose
Step 18	**set metric** *bandwidth delay reliability loading mtu*	Set the metric value to give the redistributed routes (only for EIGRP): • *bandwidth*—Metric value or IGRP bandwidth of the route in Kb/s in the range 0 to 4294967295 • *delay*—Route delay in tens of microseconds in the range 0 to 4294967295. • *reliability*—Likelihood of successful packet transmission expressed as a number between 0 and 255, where 255 means 100 percent reliability, and 0 means no reliability. • *loading*— Effective bandwidth of the route expressed as a number from 0 to 255 (255 is 100 percent loading). • *mtu*—Minimum maximum transmission unit (MTU) size of the route in bytes in the range 0 to 4294967295.
Step 19	**set metric-type** {**type-1** | **type-2**}	Set the OSPF external metric type for redistributed routes.
Step 20	**set metric-type internal**	Set the multi-exit discriminator (MED) value on prefixes advertised to an external BGP neighbor to match the IGP metric of the next hop.
Step 21	**set weight**	Set the BGP weight for the routing table. The value is from 1 to 65535.
Step 22	**end**	Return to privileged EXEC mode.
Step 23	**show route-map**	Display all configured route maps or only the one specified to verify configuration.
Step 24	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To delete an entry, use the **no route-map** *map tag* global configuration command or the **no match** or **no set** route-map configuration commands.

You can distribute routes from one routing domain into another and control route distribution.

Beginning in privileged EXEC mode, follow these steps to control route redistribution. Note that the keywords are the same as defined in the previous procedure.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router** {**bgp** | **rip** | **ospf** | **eigrp**}	Enter router configuration mode. A switch running the IP base feature set supports only the **rip** keyword.
Step 3	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** *type-value*] [**match internal** | **external** *type-value*] [**tag** *tag-value*] [**route-map** *map-tag*] [**weight** *weight*] [**subnets**]	Redistribute routes from one routing protocol to another routing protocol. If no route maps are specified, all routes are redistributed. If the keyword **route-map** is specified with no *map-tag*, no routes are distributed.

	Command	Purpose
Step 4	**default-metric** *number*	Cause the current routing protocol to use the same metric value for all redistributed routes (BGP, RIP and OSPF).
Step 5	**default-metric** *bandwidth delay reliability loading mtu*	Cause the EIGRP routing protocol to use the same metric value for all non-EIGRP redistributed routes.
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show route-map**	Display all configured route maps or only the one specified to verify configuration.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable redistribution, use the **no** form of the commands.

The metrics of one routing protocol do not necessarily translate into the metrics of another. For example, the RIP metric is a hop count, and the IGRP metric is a combination of five qualities. In these situations, an artificial metric is assigned to the redistributed route. Uncontrolled exchanging of routing information between different routing protocols can create routing loops and seriously degrade network operation.

If you have not defined a default redistribution metric that replaces metric conversion, some automatic metric translations occur between routing protocols:

- RIP can automatically redistribute static routes. It assigns static routes a metric of 1 (directly connected).
- Any protocol can redistribute other routing protocols if a default mode is in effect.

Configuring Policy-Based Routing

You can use policy-based routing (PBR) to configure a defined policy for traffic flows. By using PBR, you have more control over routing by reducing the reliance on routes derived from routing protocols. PBR can specify and implement routing policies that allow or deny paths based on:

- Identity of a particular end system
- Application
- Protocol

You can use PBR to provide equal-access and source-sensitive routing, routing based on interactive instead of batch traffic, or routing based on dedicated links. For example, you could transfer stock records to a corporate office on a high-bandwidth, high-cost link for a short time while sending routine application data such as e-mail over a low-bandwidth, low-cost link.

With PBR, you classify traffic using access control lists (ACLs) and then make traffic go through a different path. PBR is applied to incoming packets. All packets received on an interface with PBR enabled are passed through route maps. Based on the criteria defined in the route maps, packets are forwarded (routed) to the appropriate next hop.

- If packets do not match any route map statements, all set clauses are applied.
- If a statement is marked as permit and the packets do not match any route-map statements, the packets are sent through the normal forwarding channels, and destination-based routing is performed.
- For PBR, route-map statements marked as deny are not supported.

For more information about configuring route maps, see the "Using Route Maps to Redistribute Routing Information" section on page 37-84.

You can use standard IP ACLs to specify match criteria for a source address or extended IP ACLs to specify match criteria based on an application, a protocol type, or an end station. The process proceeds through the route map until a match is found. If no match is found, normal destination-based routing occurs. There is an implicit deny at the end of the list of match statements.

If match clauses are satisfied, you can use a set clause to specify the IP addresses that identify the next hop router in the path.

For details about PBR commands and keywords, see the *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.2.* For a list of PBR commands that are visible but not supported by the switch, see Appendix C, "Unsupported Commands in Cisco IOS Release 12.2(46)SE."

PBR configuration is applied to the whole stack, and all switches use the stack-master configuration.

Note This software release does not support Policy-Based Routing (PBR) when processing IPv4 and IPv6 traffic.

PBR Configuration Guidelines

Before configuring PBR, you should be aware of this information:

- To use PBR, you must have the IP services feature set enabled on the switch or the stack master.
- Multicast traffic is not policy routed. PBR applies to only to unicast traffic.
- You can enable PBR on a routed port or on an SVI.
- The switch does not support **route-map deny** statements for PBR.
- You can apply a policy route map to an EtherChannel port channel in Layer 3 mode, but you cannot apply a policy route map to a physical interface that is a member of the EtherChannel. If you try to do so, the command is rejected. When a policy route map is applied to a physical interface, that interface cannot become a member of an EtherChannel.
- You can define a maximum of 246 IP policy route maps on the switch or the switch stack.
- You can define a maximum of 512 access control entries (ACEs) for PBR on the switch or the switch stack.
- When configuring match criteria in a route map, follow these guidelines:
  - Do not match ACLs that permit packets destined for a local address. PBR would forward these packets, which could cause ping or Telnet failure or route protocol flappping.
  - Do not match ACLs with deny ACEs. Packets that match a deny ACE are sent to the CPU, which could cause high CPU utilization.
- To use PBR, you must first enable the routing template by using the **sdm prefer routing** global configuration command. PBR is not supported with the VLAN and default templates. For more information on the SDM templates, see Chapter 8, "Configuring SDM Templates."
- VRF and PBR are mutually exclusive on a switch interface. You cannot enable VRF when PBR is enabled on an interface. The reverse is also true; you cannot enable PBR when VRF is enabled on an interface.
- Web Cache Communication Protocol (WCCP) and PBR are mutually exclusive on a switch interface. You cannot enable WCCP when PBR is enabled on an interface. The reverse is also true; you cannot enable PBR when WCCP is enabled on an interface.

- The number of hardware entries used by PBR depends on the route map itself, the ACLs used, and the order of the ACLs and route-map entries.
- Policy-based routing based on packet length, ToS, set interface, set default next hop, or set default interface are not supported. Policy maps with no valid set actions or with a set action set to *Don't Fragment* are not supported.
- The switch supports QoS DSCP and IP precedence matching in PBR route maps, with these limitations:
  - You cannot apply QoS DSCP mutation maps and PBR route maps to the same interface.
  - You cannot configure DSCP transparency and PBR DSCP route maps on the same switch.
  - When you configure PBR with QoS DSCP, you can set QoS as enabled (by entering the **mls qos** global configuration command) or disabled (by entering the **no mls qos** command). When QoS is enabled, to ensure that the DSCP value of the traffic is unchanged, you should configure the DSCP trust state on the port where traffic enters the switch by entering the **mls qos trust dscp** interface configuration command. If the trust state is not DSCP, by default, all nontrusted traffic has the DSCP value marked as 0.

Enabling PBR

By default, PBR is disabled on the switch. To enable PBR, you must create a route map that specifies the match criteria and the resulting action if all of the match clauses are met. Then, you must enable PBR on an interface for that route map. All packets arriving on that interface that match the match clauses are subject to PBR.

PBR can be fast-switched or implemented at speeds that do not slow down the switch. Fast-switched PBR supports most match and set commands. You must first enable PBR before you enable fast-switched PBR. By default, fast-switched PBR is disabled.

Packets that are generated by the switch, or local packets, are not normally policy routed. When you globally enable local PBR on the switch, all packets that originate on the switch are subject to local PBR. Local PBR is disabled by default.

Note To enable PBR, the switch or stack master must be running the IP services feature set.

Beginning in privileged EXEC mode, follow these steps to configure PBR:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**route-map** *map-tag* [**permit**] [*sequence number*]	Define any route maps used to control from where packets are sent, and enter route-map configuration mode. • *map-tag*—A meaningful name for the route map. The **ip policy route-map** interface configuration command uses this name to reference the route map. Multiple route maps might share the same map-tag name. • (Optional) If **permit** is specified and the match criteria are met for this route map, the route is policy routed as controlled by the set actions. **Note** The **route-map deny** statement is not supported in PBR route maps to be applied to an interface. • *sequence number* (Optional)—Number that shows the position of a new route map in the list of route maps already configured with the same name.
Step 3	**match ip address** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*]	Match the source and destination IP addresses that are permitted by one or more standard or extended access lists. **Note** Do not enter an ACL with a deny ACE or an ACL that permits a packet destined for a local address. If you do not specify a **match** command, the route map applies to all packets.
Step 4	**set ip next-hop** *ip-address* [*...ip-address*]	Specify the action to take on the packets that match the criteria. Set the next hop to which to route the packet (the next hop must be adjacent).
Step 5	**exit**	Return to global configuration mode.
Step 6	**interface** *interface-id*	Enter interface configuration mode, and specify the interface to configure.
Step 7	**ip policy route-map** *map-tag*	Enable PBR on a Layer 3 interface, and identify the route map to use. You can configure only one route map on an interface. However, you can have multiple route map entries with different sequence numbers. These entries are evaluated in sequence number order until the first match. If there is no match, packets are routed as usual. **Note** If the IP policy route map contains a **deny** statement, the configuration fails.
Step 8	**ip route-cache policy**	(Optional) Enable fast-switching PBR. You must first enable PBR before enabling fast-switching PBR.
Step 9	**exit**	Return to global configuration mode.
Step 10	**ip local policy route-map** *map-tag*	(Optional) Enable local PBR for policy-based routing on packets originating at the switch. This applies to packets generated by the switch and not to incoming packets.

	Command	Purpose
Step 11	**end**	Return to privileged EXEC mode.
Step 12	**show route-map** [*map-name*]	(Optional) Display all configured route maps or only the one specified to verify configuration.
Step 13	**show ip policy**	(Optional) Display policy route maps attached to interfaces.
Step 14	**show ip local policy**	(Optional) Display whether or not local policy routing is enabled and, if so, the route map being used.
Step 15	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no route-map** *map-tag* global configuration command or the **no match** or **no set** route-map configuration commands to delete an entry. Use the **no ip policy route-map** *map-tag* interface configuration command to disable PBR on an interface. Use the **no ip route-cache policy** interface configuration command to disable fast-switching PBR. Use the no **ip local policy route-map** *map-tag* global configuration command to disable policy-based routing on packets originating on the switch.

Filtering Routing Information

You can filter routing protocol information by performing the tasks described in this section.

Note When routes are redistributed between OSPF processes, no OSPF metrics are preserved.

Setting Passive Interfaces

To prevent other routers on a local network from dynamically learning about routes, you can use the **passive-interface** router configuration command. Then routing update messages are not being sent through a router interface. When you use this command in the OSPF protocol, the interface address that you specify as passive appears as a stub network in the OSPF domain. OSPF routing information is neither sent nor received through that interface.

In networks with many interfaces, you do not need to manually set them as passive. You can set all interfaces to be passive by default by using the **passive-interface default** router configuration command. You can then manually setting interfaces where adjacencies are desired.

Beginning in privileged EXEC mode, follow these steps to configure passive interfaces:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router** {**bgp** | **rip** | **ospf** | **eigrp**}	Enter router configuration mode. A switch running the IP base feature set supports only the **rip** keyword.
Step 3	**passive-interface** *interface-id*	Suppress sending of routing updates through the specified Layer 3 interface.
Step 4	**passive-interface default**	(Optional) Set all interfaces as passive by default.
Step 5	**no passive-interface** *interface type*	(Optional) Activate only those interfaces that need to have adjacencies sent.

	Command	Purpose
Step 6	**network** *network-address*	(Optional) Specify the list of networks for the routing process. The *network-address* is an IP address.
Step 7	**end**	Return to privileged EXEC mode.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use a network monitoring privileged EXEC command such as **show ip ospf interface** to verify the interfaces that you enabled as passive, or use the **show ip interface** privileged EXEC command to verify the interfaces that you enabled as active.

To re-enable the sending of routing updates, use the **no passive-interface** *interface-id* router configuration command. The **default** keyword sets all interfaces as passive by default. You can then configure individual interfaces where you want adjacencies by using the **no passive-interface** router configuration command. The **default** keyword is useful in Internet service provider and large enterprise networks where many of the distribution routers have more than 200 interfaces.

Controlling Advertising and Processing in Routing Updates

You can use the **distribute-list** router configuration command with access control lists to suppress routes from being advertised in routing updates and to prevent other routers from learning one or more routes. When used in OSPF, this feature applies only to external routes, and you cannot specify an interface name.

You can also use a **distribute-list** router configuration command to avoid processing certain routes listed in incoming updates. (This feature does not apply to OSPF.)

Beginning in privileged EXEC mode, follow these steps to control the advertising or processing of routing updates:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router** {**bgp** | **rip** | **ospf** | **eigrp**}	Enter router configuration mode. A switch running the IP base feature set supports only the **rip** keyword.
Step 3	**distribute-list** {*access-list-number* | *access-list-name*} **out** [*interface-name* | *routing process* | *autonomous-system-number*]	Permit or deny routes from being advertised in routing updates, depending upon the action listed in the access list.
Step 4	**distribute-list** {*access-list-number* | *access-list-name*} **in** [*type-number*]	Suppress processing in routes listed in updates.
Step 5	**end**	Return to privileged EXEC mode.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no distribute-list in** router configuration command to change or cancel a filter. To cancel suppression of network advertisements in updates, use the **no distribute-list out** router configuration command.

Filtering Sources of Routing Information

Because some routing information might be more accurate than others, you can use filtering to prioritize information coming from different sources. An *administrative distance* is a rating of the trustworthiness of a routing information source, such as a router or group of routers. In a large network, some routing protocols can be more reliable than others. By specifying administrative distance values, you enable the router to intelligently discriminate between sources of routing information. The router always picks the route whose routing protocol has the lowest administrative distance. Table 37-14 on page 37-83 shows the default administrative distances for various routing information sources.

Because each network has its own requirements, there are no general guidelines for assigning administrative distances.

Beginning in privileged EXEC mode, follow these steps to filter sources of routing information:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router** {**bgp** | **rip** | **ospf** | **eigrp**}	Enter router configuration mode. A switch running the IP base feature set supports only the **rip** keyword.
Step 3	**distance** *weight* {*ip-address* {*ip-address mask*}} [*ip access list*]	Define an administrative distance. *weight*—The administrative distance as an integer from 10 to 255. Used alone, *weight* specifies a default administrative distance that is used when no other specification exists for a routing information source. Routes with a distance of 255 are not installed in the routing table. (Optional) *ip access list*—An IP standard or extended access list to be applied to incoming routing updates.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show ip protocols**	Display the default administrative distance for a specified routing process.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To remove a distance definition, use the **no distance** router configuration command.

Managing Authentication Keys

Key management is a method of controlling authentication keys used by routing protocols. Not all protocols can use key management. Authentication keys are available for EIGRP and RIP Version 2.

Before you manage authentication keys, you must enable authentication. See the appropriate protocol section to see how to enable authentication for that protocol. To manage authentication keys, define a key chain, identify the keys that belong to the key chain, and specify how long each key is valid. Each key has its own key identifier (specified with the **key** *number* key chain configuration command), which is stored locally. The combination of the key identifier and the interface associated with the message uniquely identifies the authentication algorithm and Message Digest 5 (MD5) authentication key in use.

You can configure multiple keys with life times. Only one authentication packet is sent, regardless of how many valid keys exist. The software examines the key numbers in order from lowest to highest and uses the first valid key it encounters. The lifetimes allow for overlap during key changes. Note that the router must know these lifetimes.

Beginning in privileged EXEC mode, follow these steps to manage authentication keys:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**key chain** *name-of-chain*	Identify a key chain, and enter key chain configuration mode.
Step 3	**key** *number*	Identify the key number. The range is 0 to 2147483647.
Step 4	**key-string** *text*	Identify the key string. The string can contain from 1 to 80 uppercase and lowercase alphanumeric characters, but the first character cannot be a number.
Step 5	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}	(Optional) Specify the time period during which the key can be received. The *start-time* and *end-time* syntax can be either *hh*:*mm*:*ss Month date year* or *hh*:*mm*:*ss date Month year*. The default is forever with the default *start-time* and the earliest acceptable date as January 1, 1993. The default *end-time* and **duration** is **infinite**.
Step 6	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}	(Optional) Specify the time period during which the key can be sent. The *start-time* and *end-time* syntax can be either *hh*:*mm*:*ss Month date year* or *hh*:*mm*:*ss date Month year*. The default is forever with the default *start-time* and the earliest acceptable date as January 1, 1993. The default *end-time* and **duration** is infinite.
Step 7	**end**	Return to privileged EXEC mode.
Step 8	**show key chain**	Display authentication key information.
Step 9	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To remove the key chain, use the **no key chain** *name-of-chain* global configuration command.

Monitoring and Maintaining the IP Network

You can remove all contents of a particular cache, table, or database. You can also display specific statistics. Use the privileged EXEC commands in Table 37-15 to clear routes or display status:

Table 37-15 Commands to Clear IP Routes or Display Route Status

Command	Purpose
clear ip route {*network* [*mask* | ***]}	Clear one or more routes from the IP routing table.
show ip protocols	Display the parameters and state of the active routing protocol process.
show ip route [*address* [*mask*] [**longer-prefixes**]] | [*protocol* [*process-id*]]	Display the state of the routing table.
show ip route summary	Display the state of the routing table in summary form.
show ip route supernets-only	Display supernets.
show ip cache	Display the routing table used to switch IP traffic.
show route-map [*map-name*]	Display all configured route maps or only the one specified.