CHAPTER 30

Configuring Layer 3 Interfaces

This chapter describes the Layer 3 interfaces on a Catalyst 4500 series switch. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

Note For complete syntax and usage information for the switch commands used in this chapter, look at the *Cisco Catalyst 4500 Series Switch Command Reference* and related publications at this location:

http://www.cisco.com/en/US/products/hw/switches/ps4324/index.html

If the command is not found in the Catalyst 4500 Command Reference, it is located in the larger Cisco IOS library. Refer to the *Cisco IOS Command Reference* and related publications at this location:

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

About Layer 3 Interfaces

The Catalyst 4500 family switch supports Layer 3 interfaces with the Cisco IOS IP and IP routing protocols. Layer 3, the *network* layer, is primarily responsible for the routing of data in packets across logical internetwork paths.

Layer 2, the *data link* layer, contains the protocols that control the *physical* layer (Layer 1) and how data is framed before being transmitted on the medium. The Layer 2 function of filtering and forwarding data in frames between two segments on a LAN is known as *bridging*.

The Catalyst 4500 series switch supports two types of Layer 3 interfaces. The logical Layer 3 VLAN interfaces integrate the functions of routing and bridging. The physical Layer 3 interfaces allow the Catalyst 4500 series switch to be configured like a traditional router.

Note On a Catalyst 4500 Series Switch, a physical Layer 3 interface has MAC address learning enabled.

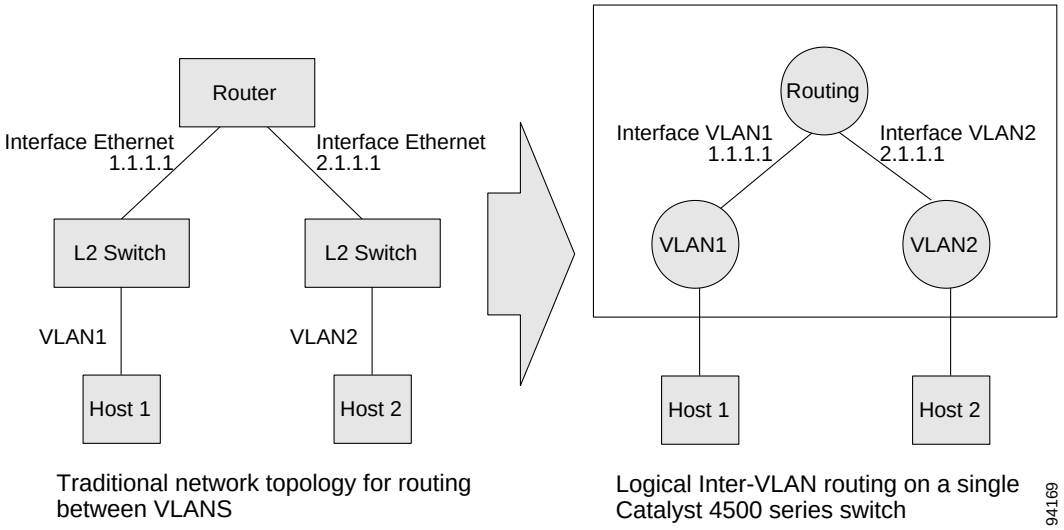
This section contains the following subsections:

Logical Layer 3 VLAN Interfaces

The logical Layer 3 VLAN interfaces provide logical routing interfaces to VLANs on Layer 2 switches. A traditional network requires a physical interface from a router to a switch to perform inter-VLAN routing. The Catalyst 4500 series switch supports inter-VLAN routing by integrating the routing and bridging functions on a single Catalyst 4500 series switch.

Figure 30-1 shows how the routing and bridging functions in the three physical devices of the traditional network are performed logically on one Catalyst 4500 series switch.

Figure 30-1 Logical Layer 3 VLAN Interfaces for the Catalyst 4500 Series Switch

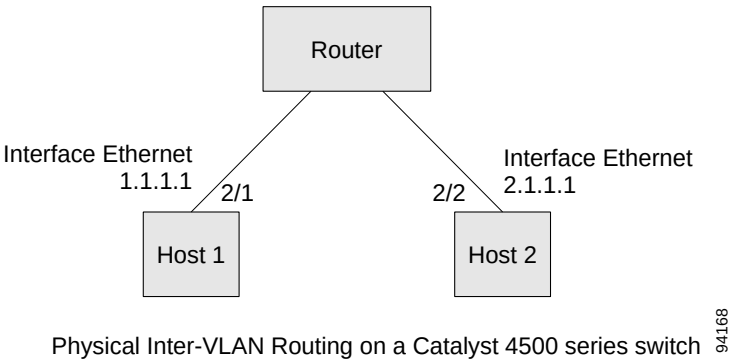


Physical Layer 3 Interfaces

The physical Layer 3 interfaces support capabilities equivalent to a traditional router. These Layer 3 interfaces provide hosts with physical routing interfaces to a Catalyst 4500 series switch.

Figure 30-2 shows how the Catalyst 4500 series switch functions as a traditional router.

Figure 30-2 Physical Layer 3 Interfaces for the Catalyst 4500 Series Switch



Understanding SVI Autostate Exclude

To be up/up, a router VLAN interface must fulfill the following general conditions:

- The VLAN exists and is active on the VLAN database of the switch.
- The VLAN interface exists on the router and is not administratively down.
- At least one Layer 2 (access port or trunk) port exists, has a link up on this VLAN, and is in spanning-tree forwarding state on the VLAN.

Note The protocol line state for the VLAN interfaces comes up when the first switch port belonging to the corresponding VLAN link comes up and is in spanning-tree forwarding state.

Ordinarily, when a VLAN interface has multiple ports in the VLAN, the SVI goes down when all the ports in the VLAN go down. The SVI Autostate Exclude feature provides a knob to mark a port so that it is not counted in the SVI up and down calculation. The feature applies to all VLANs that are enabled on that port.

A VLAN interface is brought up after the Layer 2 port has had time to converge (that is, transition from listening-learning to forwarding). This prevents routing protocols and other features from using the VLAN interface as if it were fully operational. It also prevents other problems from occurring, such as routing black holes.

Understanding Layer 3 Interface Counters

Note Catalyst 4900M, Catalyst 4948E, Supervisor Engine 6-E, and Supervisor Engine 6L-E do not support Layer 2 interface counters. However, they do support Layer 3 (SVI) interface counters.

When you run IPv4 and IPv6 on Catalyst 4900M, Catalyst 4948E, Supervisor Engine 6-E, and Supervisor Engine 6L-E, packets are routed in hardware by the forwarding engine. They support the following statistics for counting routed packets with a maximum of 4092 interfaces:

- Input unicast
- Input multicast

- Output unicast
- Output multicast

For each counter type, both the number of packets and the total number of bytes received or transmitted are counted. You can collect these statistics uniquely for IPv4 and IPv6 traffic.

Because the total number of supported Layer 3 interfaces exceeds the number of counters supported by hardware, all Layer 3 interfaces might not have counters. You assign counters to Layer 3 interfaces; the default configuration for a Layer 3 interface has no counters.

You can configure collection statistics at an interface level in one of the four ways (see Table 30-1). The maximum number of interfaces applied to the configuration depends on the collection mode.

Table 30-1 Configuring Statistics Collection Node

Counter Mode	Configuration CLI	Function	Maximum
IPv4 only	counter ipv4	Only IPv4 statistics are collected.	4092
IPv6 only	counter ipv6	Only IPv6 statistics are collected.	4092
IPv4 and IPv6 combined	counter	Both IPv4 and IPv6 statistics are collected but are displayed only as a sum.	4092
IPv4 and IPv6 separate	counter ipv4 ipv6 separate	Both IPv4 and IPv6 statistics are collected and can be displayed individually.	2046

When mixing these configured modes, the rule is as follows:

(number of v4/v6/v4v6combined interfaces) + 2*(number of v4v6separate interfaces) <= 4092

Note To enable Layer 3 interface counters, you need to enter the **counter** command in interface mode. For instructions, see the “Configuring Layer 3 Interface Counters” section on page 30-10.

The hardware counters are displayed in the output of the **show interface** command, as shown in the following example. Counter fields that are updated when the counter configuration is present are highlighted.

```
Switch# show interface gi3/1
   GigabitEthernet3/1 is up, line protocol is up (connected)
     Hardware is Gigabit Ethernet Port, address is 001f.9e9e.f43f (bia 001f.9e9e.f43f)
     Internet address is 10.10.10.2/24
     MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
        reliability 255/255, txload 1/255, rxload 1/255
     Encapsulation ARPA, loopback not set
     Keepalive set (10 sec)
     Full-duplex, 1000Mb/s, link type is auto, media type is 10/100/1000-TX
     input flow-control is on, output flow-control is on
     Auto-MDIX on (operational: on)
     ARP type: ARPA, ARP Timeout 04:00:00
     Last input never, output never, output hang never
     Last clearing of "show interface" counters never
     Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
```

```
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 53000 bits/sec, 122 packets/sec
5 minute output rate 53000 bits/sec, 122 packets/sec
L3 in Switched: ucast: 37522 pkt, 752892 bytes - mcast: 0 pkt, 0 bytes   <===== (A)
L3 out Switched: ucast: 37522 pkt, 752892 bytes - mcast: 0 pkt, 0 bytes  <===== (B)
IPv6 L3 in Switched: ucast: 24328 pkt, 145968 bytes - mcast: 0 pkt, 0 bytes  <==(C)
IPv6 L3 out Switched: ucast: 24328 pkt, 145968 bytes - mcast: 0 pkt, 0 bytes <===(D)
   103639 packets input, 6632896 bytes, 0 no buffer
   Received 0 broadcasts (0 IP multicasts)
   0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
   0 input packets with dribble condition detected
   103674 packets output, 6641715 bytes, 0 underruns
   0 output errors, 0 collisions, 1 interface resets
   0 babbles, 0 late collision, 0 deferred
   0 lost carrier, 0 no carrier
   0 output buffer failures, 0 output buffers swapped out
```

The output of the previous configuration depends on the counter configuration (Table 30-2).

Table 30-2 Fields Updated in Previous Configuration/Counter Configuration

Counter Configuration	Updated Fields
IPv4 only	(A) and (B) only
IPv6 only	(C) and (D) only
IPv4 and IPv6 combined	(A) and (B) only
IPv4 and IPv6 separate	(A) and (B) for IPv4 (C) and (D) for IPv6

Configuration Guidelines

The Catalyst 4500 series switch supports AppleTalk routing and IPX routing. For AppleTalk routing and IPX routing information, refer to “Configuring AppleTalk” and “Configuring Novell IPX” in the *Cisco IOS AppleTalk* and *Novell IPX c*onfiguration guides at the following URLs:

http://www.cisco.com/en/US/docs/ios/at/configuration/guide/12_4/atk_12_4_book.html

http://www.cisco.com/en/US/docs/ios/novipx/configuration/guide/config_novellipx_ps6350_TSD_Products_Configuration_Guide_Chapter.html

Note Catalyst 4900M, Catalyst 4948E, Supervisor Engine 6-E, and Supervisor Engine 6L- E do not support AppleTalk and IPX routing.

A Catalyst 4500 series switch does not support subinterfaces or the **encapsulation** keyword on Layer 3 Fast Ethernet, Gigabit Ethernet, 10-Gigabit Ethernet interfaces.

Note As with any Layer 3 interface running Cisco IOS software, the IP address and network assigned to an SVI cannot overlap those assigned to any other Layer 3 interface on the switch.

Configuring Logical Layer 3 VLAN Interfaces

Note Before you can configure logical Layer 3 VLAN interfaces, you must create and configure the VLANs on the switch, assign VLAN membership to the Layer 2 interfaces, enable IP routing if IP routing is disabled, and specify an IP routing protocol.

To configure logical Layer 3 VLAN interfaces, perform this task:

	Command	Purpose
Step 1	`Switch(config)# vlan vlan_ID`	Creates the VLAN.
Step 2	`Switch(config)# interface vlan vlan_ID`	Selects an interface to configure.
Step 3	`Switch(config-if)# ip address` ip_address *subnet_mask*	Configures the IP address and IP subnet.
Step 4	`Switch(config-if)# no shutdown`	Enables the interface.
Step 5	`Switch(config-if)# end`	Exits configuration mode.
Step 6	`Switch# copy running-config startup-config`	Saves your configuration changes to NVRAM.
Step 7	`Switch# show interfaces` [*type slot/interface*] `Switch# show ip interfaces` [*type slot/interface*] `Switch# show running-config interfaces` [*type slot/interface*] `Switch# show running-config interfaces vlan` vlan_ID	Verifies the configuration.

This example shows how to configure the logical Layer 3 VLAN interface VLAN 2 and assign an IP address:

```
Switch> enable
Switch# config term
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# vlan 2
Switch(config)# interface vlan 2
Switch(config-if)# ip address 10.1.1.1 255.255.255.248
Switch(config-if)# no shutdown
Switch(config-if)# end
```

This example shows how to use the **show interfaces** command to display the interface IP address configuration and status of Layer 3 VLAN interface VLAN 2:

```
Switch# show interfaces vlan 2
Vlan2 is up, line protocol is down
  Hardware is Ethernet SVI, address is 00D.588F.B604 (bia 00D.588F.B604)
  Internet address is 172.20.52.106/29
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
```

```
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
Switch#
```

This example shows how to use the **show running-config** command to display the interface IP address configuration of Layer 3 VLAN interface VLAN 2:

```
Switch# show running-config
Building configuration...

Current configuration : !
interface Vlan2
 ip address 10.1.1.1 255.255.255.248
!
ip classless
no ip http server
!
!line con 0
line aux 0
line vty 0 4
!
end
```

Configuring VLANs as Layer 3 Interfaces

This section consists of the following subsections:

Configuring SVI Autostate Exclude

Note The SVI Autostate Exclude feature is enabled by default and is synchronized with the STP state.

The SVI Autostate Exclude feature shuts down (or brings up) the Layer 3 interfaces of a switch when the following port configuration changes occur:

- When the last port on a VLAN goes down, the Layer 3 interface on that VLAN is shut down (SVI- autostated).
- When the first port on the VLAN is brought back up, the Layer 3 interface on the VLAN that was previously shut down is brought up.

SVI Autostate Exclude enables you to exclude the access ports and trunks in defining the status of the SVI (up or down) even if it belongs to the same VLAN. If the excluded access port and trunk is in up state and other ports are in down state in the VLAN, the SVI state is changed to down.

To make the SVI state up, at least one port in the VLAN should be up and not excluded. This action helps to exclude the monitoring port status when you are determining the status of the SVI.

To apply SVI Autostate Exclude, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface` *interface-id*	Enters interface configuration mode.
Step 3	`Switch(config-if)# switchport autostate exclude`	Excludes the access ports and trunks in defining the status of an SVI (up or down).
Step 4	`Switch(config)# end`	Exits configuration mode.
Step 5	`Switch# show run int g3/4`	Displays the running configuration.
Step 6	`Switch# show int g3/4 switchport`	Verifies the configuration.

This example shows how to apply SVI Autostate Exclude on interface g3/1:

```
Switch# conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface g3/1
Switch(config-if)# switchport autostate exclude
Switch(config-if)# end
Switch# show run int g3/4
Building configuration...

Current configuration : 162 bytes
!
interface GigabitEthernet3/4
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 2,3
 switchport autostate exclude                                    <=====
 switchport mode trunk
end

Switch# show int g3/4 switchport
Name: Gi3/4
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On Access Mode VLAN: 1 (default) Trunking Native Mode VLAN: 1
(default) Administrative Native VLAN tagging: enabled Voice VLAN: none Administrative
private-vlan host-association: none Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none Administrative private-vlan trunk
Native VLAN tagging: enabled Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none Administrative private-vlan trunk
associations: none Administrative private-vlan trunk mappings: none Operational
private-vlan: none Trunking VLANs Enabled: 2,3 Pruning VLANs Enabled: 2-1001 Capture Mode
Disabled Capture VLANs Allowed: ALL
Autostate mode exclude                                        <======

Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none
Switch#
```

Configuring IP MTU Sizes

You can set the protocol-specific maximum transmission unit (MTU) size of IPv4 or IPv6 packets that are sent on an interface.

For information on MTU limitations, refer to "Maximum Transmission Units" on page 23.

Note To set the nonprotocol-specific MTU value for an interface, use the **mtu** interface configuration command. Changing the MTU value (with the **mtu** interface configuration command) can affect the IP MTU value. If the current IP MTU value matches the MTU value, and you change the MTU value, the IP MTU value is modified automatically to match the new MTU. However, the reverse is not true; changing the IP MTU value has no effect on the value for the **mtu** command.

For information on how to configure MTU size, refer to "Configuring MTU Sizes" on page 25.

To set the protocol-specific maximum transmission unit (MTU) size of IPv4 or IPv6 packets sent on an interface, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface` *interface-id*	Enters interface configuration mode.
Step 3	`Switch(config-if)# [no] ip mtu` *mtu_size* or `Switch(config-if)# [no] ipv6 mtu` *mtu_size*	Configures the IPv4 MTU size Configures the IPv6 MTU size. The **no** form of the command reverts to the default MTU size (1500 bytes).
Step 4	`Switch(config-if)# exit`	Exits configuration interface mode.
Step 5	`Switch(config)# end`	Exits configuration mode.
Step 6	`Switch# show run interface` *interface-id*	Displays the running configuration.

This example shows how to configure IPv4 MTU on an interface:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface vlan 1
Switch(config-if)# ip mtu 68
Switch(config-if)# exit
Switch(config)# end
Switch# show ip interface vlan 1
Vlan1 is up, line protocol is up
  Internet address is 10.10.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 68 bytes
  Helper address is not set
.....................(continued)
```

The following example shows how to configure IPv6 MTU on an interface:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface vlan 1
Switch(config-if)# ipv6 mtu 1280
Switch(config)# end
```

This example shows how to verify the configuration

```
Switch# show ipv6 interface vlan 1
Vlan1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::214:6AFF:FEBC:DEEA
  Global unicast address(es):
    1001::1, subnet is 1001::/64
  Joined group address(es):
    FF02::1
    FF02::1:FF00:1
    FF02::1:FFBC:DEEA
  MTU is 1280 bytes
...................(continued)
```

Note When IPv6 is enabled on an interface using any CLI command, you may see the following message:

```
% Hardware MTU table exhausted
```

In this situation, the IPv6 MTU value programmed in hardware differs from the IPv6 interface MTU value. This situation occurs if no room exists in the hardware MTU table to store additional values. You must free up some space in the table by unconfiguring some unused MTU values and subsequently disable and reenable IPv6 on the interface or reapply the MTU configuration.

Configuring Layer 3 Interface Counters

Note Catalyst 4900M, Catalyst 4948E, Supervisor Engine 6-E, and Supervisor Engine 6L-E do not support Layer 2 interface counters.

To configure Layer 3 interface counters (assign counters to a Layer 3 interface), perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface` *interface-id*	Enters interface configuration mode.

	Command	Purpose
Step 3	`Switch(config-if)# counter {ipv4 | ipv6 | ipv4 ipv6 separate>`	Enables counters. **counter** —Enables collection of IPv4 and IPv6 statistics and displays them as a sum **counter ipv4** — Enables collection of IPv4 statistics only **counter ipv6** — Enables collection of IPv6 statistics only **counter ipv4 ipv6 separate** —Enables collection of IPv4 and IPv6 statistics and displays them individually
Step 4	`Switch(config)# end`	Exits configuration mode.
Step 5	`Switch# show run interface interface-id`	Displays the running configuration.

This example shows how to enable counters on interface VLAN 1:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface vlan 1
Switch(config-if)# counter ipv4
Switch(config-if)# end
Switch#
00:17:15: %SYS-5-CONFIG_I: Configured from console by console
Switch# show run interface vlan 1
Building configuration...

Current configuration : 63 bytes
!
interface Vlan1
 ip address 10.0.0.1 255.0.0.0
 counter ipv4
end
```

Note To remove the counters, use the **no counter** command.

If you have already assigned the maximum number of counters, the **counter** command fails and displays an error message:

```
Switch# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fa3/2
Switch(config-if)# no switchport
Switch(config-if)# counter ipv6
 Counter resource exhausted for interface fa3/2
Switch(config-if)# end
Switch#
00:24:18: %SYS-5-CONFIG_I: Configured from console by console
```

In this situation, you must release a counter from another interface for use by the new interface.

Configuring Physical Layer 3 Interfaces

Note Before you can configure physical Layer 3 interfaces, you must enable IP routing if IP routing is disabled, and specify an IP routing protocol.

To configure physical Layer 3 interfaces, perform this task:

	Command	Purpose
Step 1	`Switch(config)#ip routing`	Enables IP routing (required only if disabled)
Step 2	`Switch(config)# interface {fastethernet | gigabitethernet | tengigabitethernet} slot/port} | {port-channel port_channel_number}`	Selects an interface to configure.
Step 3	`Switch(config-if)#no switchport`	Converts this port from physical Layer 2 port to physical Layer 3 port.
Step 4	`Switch(config-if)# ip address ip_address subnet_mask`	Configures the IP address and IP subnet.
Step 5	`Switch(config-if)# no shutdown`	Enables the interface.
Step 6	`Switch(config-if)# end`	Exits configuration mode.
Step 7	`Switch# copy running-config startup-config`	Saves your configuration changes to NVRAM.

Step 8	`Switch# show interfaces [type slot/interface]`
	`Switch# show ip interfaces [type slot/interface]`
	`Switch# show running-config interfaces [type slot/interface]`
		Verifies the configuration.

This example shows how to configure an IP address on Fast Ethernet interface 2/1:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# interface fastethernet 2/1
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.1.1.1 255.255.255.248
Switch(config-if)# no shutdown
Switch(config-if)# end
Switch#
```

This example shows how to use the **show running-config** command to display the interface IP address configuration of Fast Ethernet interface 2/1:

```
Switch# show running-config
Building configuration...
!
interface FastEthernet2/1
 no switchport
 ip address 10.1.1.1 255.255.255.248
!
…
ip classless
no ip http server
!
!
line con 0
line aux 0
```

```
line vty 0 4
!
end
```

Configuring EIGRP Stub Routing

This section consists of the following subsections:

About EIGRP Stub Routing

The EIGRP stub routing feature, available in all images, reduces resource utilization by moving routed traffic closer to the end user.

The IP base image contains only EIGRP stub routing. The IP services image contains complete EIGRP routing.

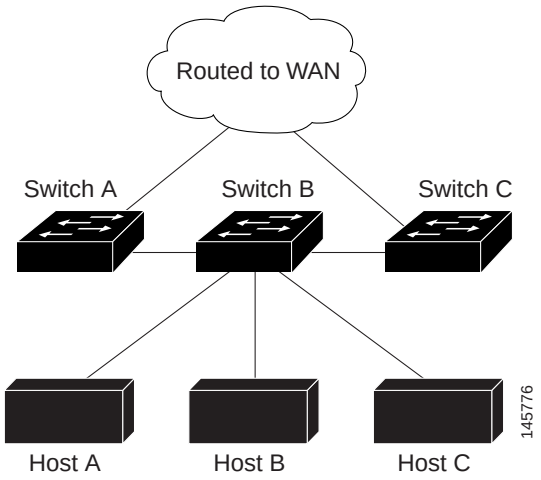
In a network using EIGRP stub routing, the only route for IP traffic to follow to the user is through a switch that is configured with EIGRP stub routing. The switch sends the routed traffic to interfaces that are configured as user interfaces or are connected to other devices.

When using EIGRP stub routing, you need to configure the distribution and remote switches 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 switch for any routes, and a switch that has a stub peer does not query that peer. The stub switch depends on the distribution switch to send the proper updates to all peers.

In Figure 30-3, switch B is configured as an EIGRP stub switch. Switches A and C are connected to the rest of the WAN. Switch B advertises connected, static, redistribution, and summary routes from switch A and C to Hosts A, B, and C. Switch B does not advertise any routes learned from switch A (and the reverse).

Figure 30-3 EIGRP Stub Switch Configuration



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

Configuring EIGRP Stub Routing

The EIGRP stub routing feature improves network stability, reduces resource utilization, and simplifies stub switch configuration.

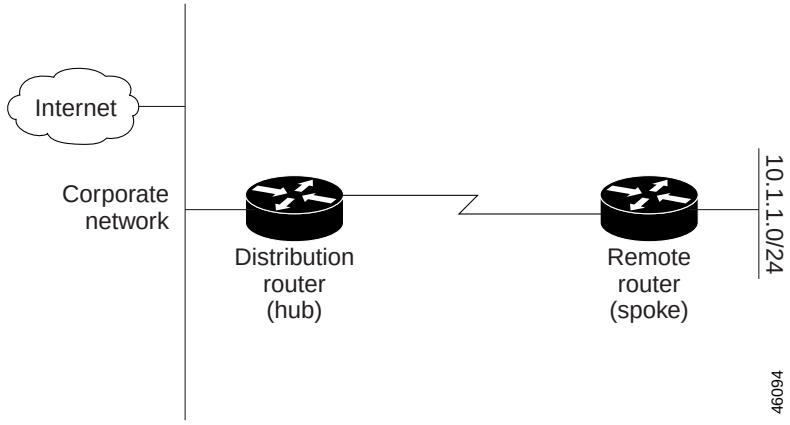
Stub routing is commonly used in a hub-and-spoke network topology. In a hub-and-spoke network, one or more end (stub) networks are connected to a remote switch (the spoke) that is connected to one or more distribution switches (the hub). The remote switch is adjacent only to one or more distribution switches. The only route for IP traffic to follow into the remote switch is through a distribution switch. This type of configuration is commonly used in WAN topologies where the distribution switch is directly connected to a WAN. The distribution switch can be connected to many more remote switches. Often, the distribution switch is connected to 100 or more remote routers. In a hub-and-spoke topology, the remote router must forward all nonlocal traffic to a distribution router, so it becomes unnecessary for the remote router to hold a complete routing table. Generally, the distribution router need not send anything more than a default route to the remote router.

When using the EIGRP stub routing feature, you need to configure the distribution and remote routers to use EIGRP, and to configure only the remote router as a stub. Only specified routes are propagated from the remote (stub) router. The stub router responds to all queries for summaries, connected routes, redistributed static routes, external routes, and internal routes with the message "inaccessible." A router that is configured as a stub sends a special peer information packet to all neighboring routers to report its status as a stub router.

Any neighbor that receives a packet 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.

Figure 30-4 shows a simple hub-and-spoke configuration.

Figure 30-4 Simple Hub-and-Spoke Network



The stub routing feature does not prevent routes from being advertised to the remote router. In the example in Figure 30-4, the remote router can access the corporate network and the Internet using a distribution router only. In this example, having a full route table on the remote router serves no purpose because the path to the corporate network and the Internet always uses a distribution router. The larger route table only reduces the amount of memory required by the remote router. Bandwidth and memory can be conserved by summarizing and filtering routes in the distribution router. The remote router need not receive routes that have been learned from other networks because the remote router must send all nonlocal traffic, regardless of destination, to the distribution router. If a true stub network is desired, the distribution router should be configured to send only a default route to the remote router. The EIGRP stub routing feature does not automatically enable summarization on the distribution router. In most cases, the network administrator needs to configure summarization on the distribution routers.

Note When configuring the distribution router to send only a default route to the remote router, you must use the **ip classless** command on the remote router. By default, the **ip classless** command is enabled in all Cisco IOS images that support the EIGRP stub routing feature.

Without the stub feature, even after the routes that are sent from the distribution router to the remote router have been filtered or summarized, a problem might occur. If a route is lost somewhere in the corporate network, EIGRP could send a query to the distribution router, which in turn sends a query to the remote router even if routes are being summarized. If there is a problem communicating over the WAN link between the distribution router and the remote router, an EIGRP stuck in active (SIA) condition could occur and cause instability elsewhere in the network. The EIGRP stub routing feature allows a network administrator to prevent queries from being sent to the remote router.

Dual-Homed Remote Topology

In addition to a simple hub-and-spoke network where a remote router is connected to a single distribution router, the remote router can be dual-homed to two or more distribution routers. This configuration adds redundancy and introduces unique issues, and the stub feature helps to address some of these issues.

A dual-homed remote router has two or more distribution (hub) routers. However, the principles of stub routing are the same as they are with a hub-and-spoke topology. Figure 30-5 shows a common dual-homed remote topology with one remote router, but 100 or more routers could be connected on the same interfaces on distribution router 1 and distribution router 2. The remote router uses the best route to reach its destination. If distribution router 1 experiences a failure, the remote router can still use distribution router 2 to reach the corporate network.

Figure 30-5 Simple Dual-Homed Remote Topology

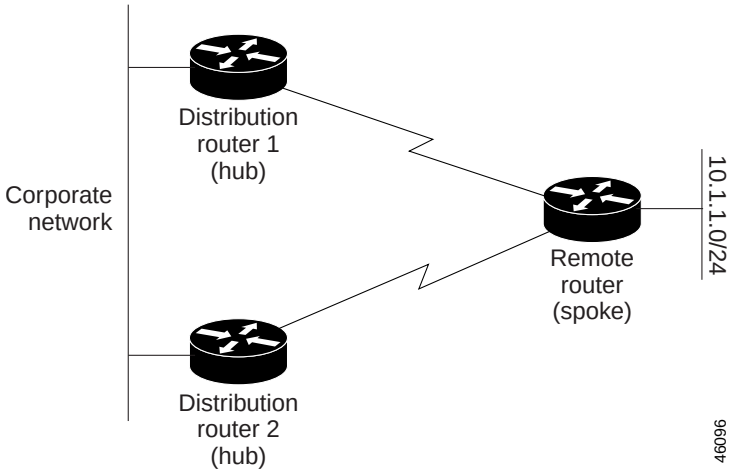


Figure 30-5 shows a simple dual-homed remote with one remote router and two distribution routers. Both distribution routers maintain routes to the corporate network and stub network 10.1.1.0/24.

Dual-homed routing can introduce instability into an EIGRP network. In Figure 30-6, distribution router 1 is directly connected to network 10.3.1.0/24. If summarization or filtering is applied on distribution router 1, the router advertises network 10.3.1.0/24 to all of its directly connected EIGRP neighbors (distribution router 2 and the remote router).

Figure 30-6 Dual-Homed Remote Topology With Distribution Router 1 Connected to Two Networks

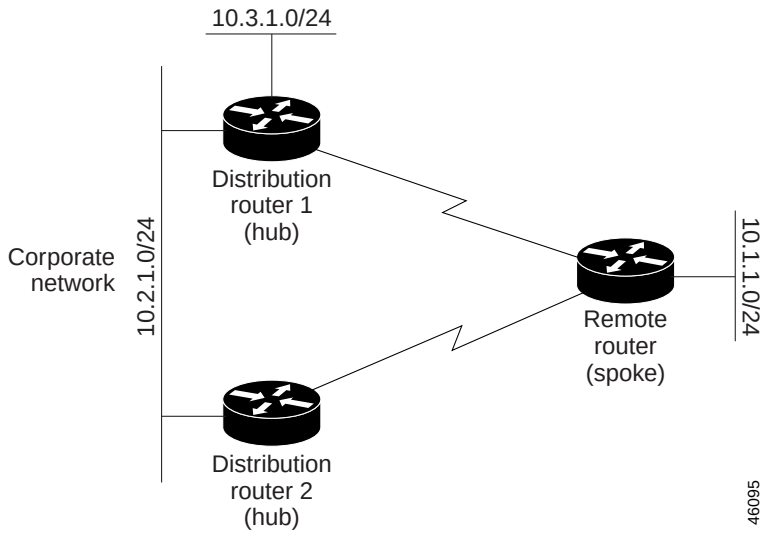
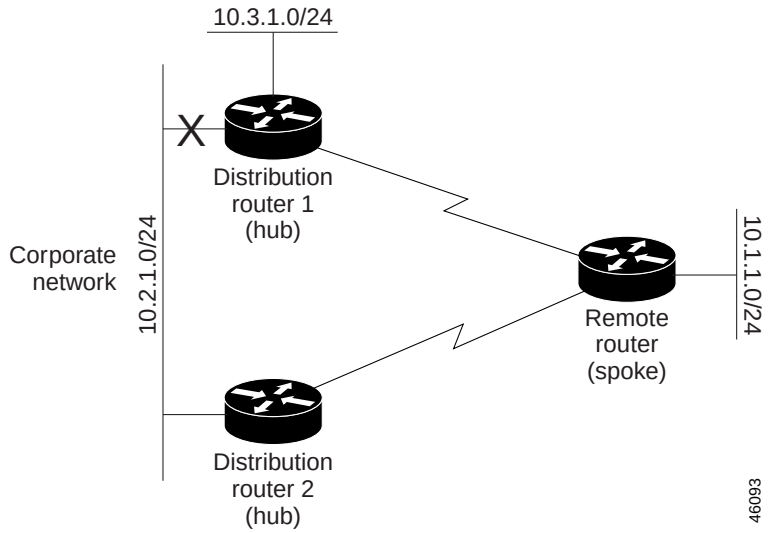


Figure 30-6 shows a simple dual-homed remote router where distribution router 1 is connected to both network 10.3.1.0/24 and network 10.2.1.0/24.

If the 10.2.1.0/24 link between distribution router 1 and distribution router 2 has failed, the lowest cost path to network 10.3.1.0/24 from distribution router 2 is using the remote router (see Figure 30-7). This route is not desirable because the traffic that was previously traveling across the corporate network 10.2.1.0/24 is now sent across a much lower bandwidth connection. The over utilization of the lower bandwidth WAN connection can cause a number of problems that might affect the entire corporate

network. The use of the lower bandwidth route that passes using the remote router might cause WAN EIGRP distribution routers to be dropped. Serial lines on distribution and remote routers could also be dropped, and EIGRP SIA errors on the distribution and core routers could occur.

Figure 30-7 Dual-Homed Remote Topology with a Failed Route to a Distribution Router



It is not desirable for traffic from distribution router 2 to travel through any remote router in order to reach network 10.3.1.0/24. If the links are sized to handle the load, it is acceptable to use one of the backup routes. However, most networks of this type have remote routers located at remote offices with relatively slow links. This problem can be prevented if proper summarization is configured on the distribution router and remote router.

It is typically undesirable for traffic from a distribution router to use a remote router as a transit path. A typical connection from a distribution router to a remote router has much less bandwidth than a connection at the network core. Attempting to use a remote router with a limited bandwidth connection as a transit path generally produces excessive congestion to the remote router. The EIGRP stub routing feature can prevent this problem by preventing the remote router from advertising core routes back to distribution routers. Routes learned by the remote router from distribution router 1 are not advertised to distribution router 2. Because the remote router does not advertise core routes to distribution router 2, the distribution router does not use the remote router as a transit for traffic destined for the network core.

The EIGRP stub routing feature can help to provide greater network stability. In the event of network instability, this feature prevents EIGRP queries from being sent over limited bandwidth links to nontransit routers. Instead, distribution routers to which the stub router is connected answer the query on behalf of the stub router. This feature greatly reduces the chance of further network instability due to congested or problematic WAN links. The EIGRP stub routing feature also simplifies the configuration and maintenance of hub-and-spoke networks. When stub routing is enabled in dual-homed remote configurations, it is no longer necessary to configure filtering on remote routers to prevent those remote routers from appearing as transit paths to the hub routers.

Caution EIGRP stub routing should only be used on stub routers. A stub router is defined as a router connected to the network core or distribution layer through which core transit traffic should not flow. A stub router should not have any EIGRP neighbors other than distribution routers. Ignoring this restriction causes undesirable behavior.

Note Multi-access interfaces, such as ATM, Ethernet, Frame Relay, ISDN PRI, and X.25, are supported by the EIGRP stub routing feature only when all routers on that interface, except the hub, are configured as stub routers.

EIGRP Stub Routing Configuration Tasks

To configure EIGRP stub routing, perform the tasks described in the following sections. The tasks in the first section are required; the task in the last section is optional.

- Configuring EIGRP Stub Routing (required)
- Verifying EIGRP Stub Routing (optional)

Configuring EIGRP Stub Routing

To configure a remote or spoke router for EIGRP stub routing, perform this task:

	Command	Purpose
Step 1	`Switch(config)# router eigrp 1`	Configures a remote or distribution router to run an EIGRP process.
Step 2	`Switch(config-router)# network network-number`	Specifies the network address of the EIGRP distribution router.
Step 3	`Switch(config-router)# eigrp stub [receive-only | connected | static | summary | redistributed]`	Configures a remote router as an EIGRP stub router. The **receive-only** keyword sets the router as a receive-only neighbor. The **connected** keyword advertises connected routes. The **static** keyword advertises static routes. The **summary** keyword advertises summary routes. The **redistributed** keyword enables you to send routes that have been redistributed from other dynamic routing protocols. **Note** It is still necessary to redistribute the routes from the other routing processes with the **redistribute** command.

Verifying EIGRP Stub Routing

To verify that a remote router has been configured as a stub router with EIGRP, use the **show ip eigrp neighbor detail** command from the distribution router in privileged EXEC mode. The last line of the output shows the stub status of the remote or spoke router. The following example shows output is from the **show ip eigrp neighbor detail** command:

```
Switch# show ip eigrp neighbor detail
IP-EIGRP neighbors for process 1
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq Type
```

```
                                          (sec)         (ms)       Cnt Num
0   10.1.1.2                Se3/1           11 00:00:59    1  4500  0  7
   Version 12.1/1.2, Retrans: 2, Retries: 0
   Stub Peer Advertising ( CONNECTED SUMMARY ) Routes
```

Monitoring and Maintaining EIGRP

To delete neighbors from the neighbor table, use the following command:

Command	Purpose
`Switch# clear ip eigrp neighbors [ip-address | interface]`	Deletes neighbors from the neighbor table.

To display various routing statistics, use the following commands:

Command	Purpose
`Switch# show ip eigrp interfaces [interface] [as-number]`	Displays information about interfaces configured for EIGRP.
`Switch# show ip eigrp neighbors [type|number|static]`	Displays the EIGRP discovered neighbors.
`Switch# show ip eigrp topology [autonomous-system-number | [[ip-address] mask]]`	Displays the EIGRP topology table for a given process.
`Switch# show ip eigrp traffic [autonomous-system-number]`	Displays the number of packets sent and received for all or a specified EIGRP process.

EIGRP Configuration Examples

This section contains the following examples:

- Route Summarization Example
- Route Authentication Example
- Stub Routing Example

Route Summarization Example

The following example disables autosummarization and causes EIGRP to summarize network 10.0.0.0 out Ethernet interface 0 only:

```
interface Ethernet 0
 ip summary-address eigrp 1 10.0.0.0 255.0.0.0
!
router eigrp 1
 network 172.16.0.0
 no auto-summary
```

Note You should not use the **ip summary-address eigrp** summarization command to generate the default route (0.0.0.0) from an interface because it creates an EIGRP summary default route to the null 0 interface with an administrative distance of 5. The low administrative distance of this default route can cause this route to displace default routes learned from other neighbors from the routing table. If the

default route learned from the neighbors is displaced by the summary default route, or if the summary route is the only default route present, all traffic destined for the default route does not leave the router. Instead, this traffic is sent to the null 0 interface where it is dropped.

The recommended way to send only the default route out a given interface is to use a **distribute-list** command. You can configure this command to filter all outbound route advertisements sent out the interface with the exception of the default (0.0.0.0).

Route Authentication Example

The following example enables MD5 authentication on EIGRP packets in autonomous system 1:

Router A

```
interface ethernet 1
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 holly
key chain holly
 key 1
  key-string 0987654321
  accept-lifetime 04:00:00 Dec 4 1996 infinite
  send-lifetime 04:00:00 Dec 4 1996 04:48:00 Dec 4 1996
 exit
 key 2
  key-string 1234567890
  accept-lifetime 04:00:00 Dec 4 1996 infinite
  send-lifetime 04:45:00 Dec 4 1996 infinite
```

Router B

```
interface ethernet 1
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 mikel
key chain mikel
 key 1
  key-string 0987654321
  accept-lifetime 04:00:00 Dec 4 1996 infinite
  send-lifetime 04:00:00 Dec 4 1996 infinite
 exit
 key 2
  key-string 1234567890
  accept-lifetime 04:00:00 Dec 4 1996 infinite
  send-lifetime 04:45:00 Dec 4 1996 infinite
```

Router A accepts and attempts to verify the MD5 digest of any EIGRP packet with a key equal to 1. It also accepts a packet with a key equal to 2. All other MD5 packets are dropped. Router A sends all EIGRP packets with key 2.

Router B accepts key 1 or key 2, and sends key 1. In this scenario, MD5 authenticates.

Stub Routing Example

A router that is configured as a stub with the **eigrp stub** command shares connected and summary routing information with all neighbor routers by default. Four optional keywords can be used with the **eigrp stub** command to modify this operation:

- receive-only
- connected

- static
- summary

This section provides configuration examples for all forms of the **eigrp stub** command. The **eigrp stub** command can be modified with several options, and these options can be used in any combination except for the **receive-only** keyword. The **receive-only** keyword restricts the router from sharing any of its routes with any other router in that EIGRP autonomous system, and the **receive-only** keyword does not permit any other option to be specified because it prevents any type of route from being sent. The three other optional keywords (**connected**, **static**, and **summary**) can be used in any combination but cannot be used with the **receive-only** keyword. If any of these three keywords is used individually with the **eigrp stub** command, connected and summary routes are not sent automatically.

The **connected** keyword permits the EIGRP stub routing feature to send connected routes. If the connected routes are not covered by a network statement, it may be necessary to redistribute connected routes with the **redistribute connected** command under the EIGRP process. This option is enabled by default.

The **static** keyword permits the EIGRP stub routing feature to send static routes. Without the **static** keyword, EIGRP does not send any static routes, including internal static routes that normally are automatically redistributed. It is still necessary to redistribute static routes with the **redistribute static** command.

The **summary** keyword permits the EIGRP stub routing feature to send summary routes. Summary routes can be created manually with the **summary address** command or automatically at a major network border router with the **auto-summary** command enabled. This option is enabled by default.

In the following example, the **eigrp stub** command is used to configure the router as a stub that advertises connected and summary routes:

```
router eigrp 1
network 10.0.0.0
eigrp stub
```

In the following example, the **eigrp stub connected static** command is used to configure the router as a stub that advertises connected and static routes (sending summary routes is not permitted):

```
router eigrp 1
network 10.0.0.0
eigrp stub connected static
```

In the following example, the **eigrp stub receive-only** command is used to configure the router as a stub, and connected, summary, or static routes is not sent:

```
router eigrp 1
network 10.0.0.0 eigrp
stub receive-only
```

In the following example, the **eigrp stub redistributed** command is used to configure the router as a stub that advertises redistributed routes (sending connected, static, or summary routes is not permitted):

```
router eigrp 1
network 10.0.0.0
eigrp stub redistributed
```

In the following example, the **eigrp stub** command is used to configure the router as a stub that advertises redistributed, static, connected and summary routes:

```
router eigrp 1
network 10.0.0.0
eigrp stub connected static summary redistributed
```