



Configuring Layer 3 Interfaces

This chapter describes the Layer 3 interfaces on the Catalyst 4006 switch with Supervisor Engine III. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

- [Overview of Layer 3 Interfaces, page 10-1](#)
- [Configuration Guidelines, page 10-3](#)
- [Configuring Logical Layer 3 VLAN Interfaces, page 10-3](#)
- [Configuring Physical Layer 3 Interfaces, page 10-4](#)



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at the following URL:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm>

Overview of Layer 3 Interfaces

The Catalyst 4006 switch with Supervisor Engine III 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 4006 switch with Supervisor Engine III 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 4006 switch with Supervisor Engine III to be configured like a traditional router.

This section contains the following subsections:

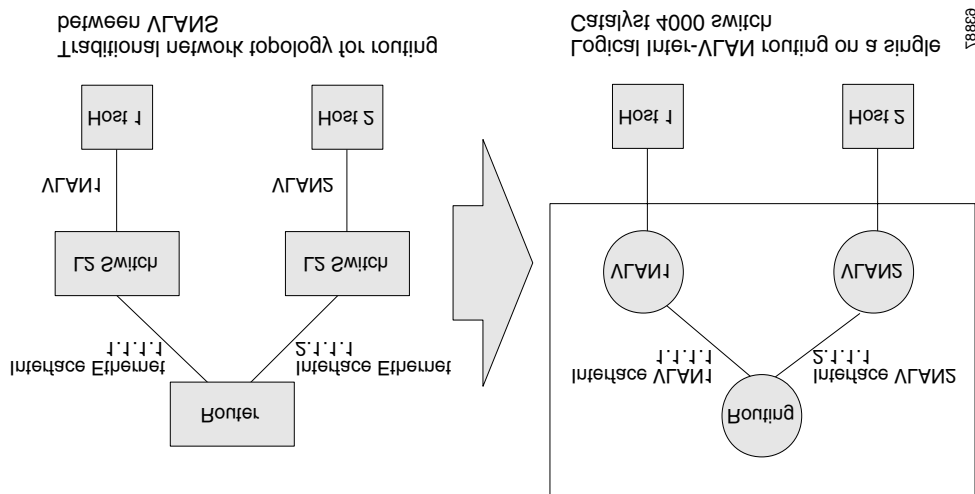
- [Logical Layer 3 VLAN Interfaces, page 10-2](#)
- [Physical Layer 3 Interfaces, page 10-2](#)

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 4006 switch with Supervisor Engine III supports inter-VLAN routing by integrating the routing and bridging functions on a single Catalyst 4006 switch with Supervisor Engine III.

Figure 10-1 shows how the routing and bridging functions in the three physical devices of the traditional network are performed logically on one Catalyst 4006 switch with Supervisor Engine III.

Figure 10-1 Logical Layer 3 VLAN Interfaces for the Catalyst 4006 Switch with Supervisor Engine III

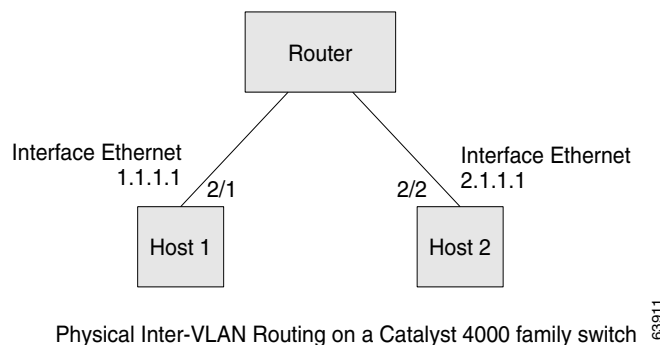


Physical Layer 3 Interfaces

The physical Layer 3 interfaces support capabilities equivalent to a traditional router. These Layer 3 interfaces provide hosts with physical routing interfaces to the Catalyst 4006 switch with Supervisor Engine III.

Figure 10-2 shows how the Catalyst 4006 switch with Supervisor Engine III functions as a traditional router.

Figure 10-2 Physical Layer 3 Interfaces for the Catalyst 4006 with Supervisor Engine III



Configuration Guidelines

The Catalyst 4006 switch with Supervisor Engine III does not support:

- Subinterfaces or the **encapsulation** keyword on Layer 3 Fast Ethernet or Gigabit Ethernet interfaces
- IPX routing and Appletalk routing

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 procedure:

Task	Command
Step 1 Create the VLAN.	Switch(config)# vlan <i>vlan_ID</i>
Step 1 Select an interface to configure.	Switch(config)# interface vlan <i>vlan_ID</i>
Step 2 Configure the IP address and IP subnet.	Switch(config-if)# ip address <i>ip_address subnet_mask</i>
Step 3 Enable the interface.	Switch(config-if)# no shutdown
Step 4 Exit configuration mode.	Switch(config-if)# end
Step 5 Save your configuration changes to NVRAM.	Switch# copy running-config startup-config
Step 6 Verify the configuration.	Switch# show interfaces [<i>type slot/interface</i>] Switch# show ip interfaces [<i>type slot/interface</i>] Switch# show running-config interfaces [<i>type slot/interface</i>] Switch# show running-config interfaces vlan <i>vlan_ID</i>

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 uses 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 uses 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 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 procedure:

	Task	Command
Step 1	Enable IP routing (Required only if disabled.)	Switch(config)# ip routing
Step 1	Select an interface to configure.	Switch(config)# interface { fastethernet gigabitethernet } <i>slot/port</i> { port-channel <i>port_channel_number</i> }
Step 2	Convert this port from physical Layer 2 port to physical Layer 3 port.	Switch(config-if)# no switchport
Step 3	Configure the IP address and IP subnet.	Switch(config-if)# ip address <i>ip_address</i> <i>subnet_mask</i>
Step 4	Enable the interface.	Switch(config-if)# no shutdown

	Task	Command
Step 5	Exit configuration mode.	Switch(config-if)# end
Step 6	Save your configuration changes to NVRAM.	Switch# copy running-config startup-config
Step 7	Verify the configuration.	Switch# show interfaces [<i>type slot/interface</i>] Switch# show ip interfaces [<i>type slot/interface</i>] Switch# show running-config interfaces [<i>type slot/interface</i>]

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 uses 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
```

