

Configuring Virtual Interfaces

Virtual interfaces are software-based interfaces that you create in the memory of the networking device using Cisco IOS XE commands. Virtual interfaces do not have a hardware component such as the RJ-45 female port on a 100BASE-T Fast Ethernet network interface card. This module describes the four common types of virtual, or logical, interfaces that can be configured using Cisco IOS XE software:

- Loopback interfaces
- Null interfaces
- Subinterfaces
- Tunnel interfaces

- Finding Feature Information, on page 1
- Prerequisites for Configuring Virtual Interfaces, on page 2
- Information About Configuring Virtual Interfaces, on page 2
- How to Configure Virtual Interfaces, on page 6
- Configuration Examples for Virtual Interfaces, on page 14
- Where to Go Next, on page 15
- Additional References, on page 15

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring Virtual Interfaces

Before virtual interfaces can be used in your network, you must have some physical (hardware) interfaces configured and you must be able to communicate between the networking devices on which you wish to use virtual interfaces.

Information About Configuring Virtual Interfaces

Virtual Interfaces

Virtual interfaces are network interfaces that are not associated with a physical interface. Physical interfaces have some form of physical element--for example, an RJ-45 male connector on an Ethernet cable. Virtual interfaces exist only in software; there are no physical elements. You identify an individual virtual interface using a numerical ID after the virtual interface name. For example: loopback 0, tunnel 1, and fastethernet 0/0/0.1. The ID is unique per virtual interface type to make the entire name string unique; for example both a loopback 0 interface and a null 0 interface can exist, but two loopback 0 interfaces cannot exist in a single networking device.

Cisco IOS XE software supports four types of virtual interfaces:

- Loopback
- Null
- Subinterface
- Tunnel

Benefits of Virtual Interfaces

A loopback interface can provide a stable interface on which you can assign a Layer 3 address such as an IP or IPX address. This address can be configured as the source address when the networking device needs to send data for protocols such as NetFlow or Cisco Discovery Protocol (CDP) to another device in your network and you want the receiving device to always see the same source IP address from the networking device. This is an issue in networks with multiple equal-cost paths because under normal circumstances the packets that are generated by a networking device use the IP address from the outbound interface as the source address for the packets and because in a network with two or more equal-cost paths from the networking device to the receiving host each packet might use a different outbound interface.

A null interface provides an alternative method of filtering without the overhead involved with using access lists. For example, instead of creating an outbound access list that prevents traffic to a destination network from being transmitted out an interface, you can configure a static route for the destination network that points to the null interface.

Subinterfaces were invented as a method of virtually subdividing a physical interface into two or more interfaces so that the IP routing protocols would see the network connection to each remote networking device as a separate physical interface even though the subinterfaces share a common physical interface. One of the first uses of subinterfaces was to resolve the problem with split horizon on Frame Relay WANs.

The following are several situations in which tunneling (encapsulating traffic in another protocol) is useful:

- To enable multiprotocol local networks over a single-protocol backbone
- To provide workarounds for networks that use protocols that have limited hop counts; for example, RIP version 1, AppleTalk
- To connect discontiguous subnetworks
- To allow virtual private networks across WANs

Loopback Interfaces

You can specify a software-only interface called a loopback interface to emulate a physical interface. Loopback interfaces are supported on all platforms. A loopback interface is a virtual interface on a Cisco router that remains up (active) after you issue the **no shutdown** command until you disable it with the **shutdown** command. Unlike subinterfaces, loopback interfaces are independent of the state of any physical interface.

The loopback interface can be considered stable because once you enable it, it will remain up until you shut it down. This makes loopback interfaces ideal for assigning Layer 3 addresses such as IP addresses when you want a single address as a reference that is independent of the status of any physical interfaces in the networking device. A good example of this is using the IP address of a loopback interface as the IP address for the domain name system (DNS) host address for the networking device. Before loopback interfaces were available, network administrators had to configure a DNS host entry for every interface on a router that had an IP address assigned to it because they could never be certain which interface IP address might be available at any given time for managing the router. In the sample interface configuration and DNS entries for Router A shown below, you can see that there is a DNS entry for each interface.

Router A Interface Configuration Before Loopback

```
GigabitEthernet0 10.10.10.1 255.255.255.0
GigabitEthernet1 10.10.11.1 255.255.255.0
GigabitEthernet2 10.10.12.1 255.255.255.0
GigabitEthernet3 10.10.13.1 255.255.255.0
GigabitEthernet4 10.10.14.1 255.255.255.0
GigabitEthernet5 10.10.15.1 255.255.255.0
```

Router A DNS Entries Before Loopback

```
RouterA   IN  A  10.10.10.1
          IN  A  10.10.11.1
          IN  A  10.10.12.1
          IN  A  10.10.13.1
          IN  A  10.10.14.1
          IN  A  10.10.15.1
```

Interfaces on networking devices can fail, and they can also be taken out of service for maintenance. If any of the interfaces in Router A fails or is taken out of service, another networking device will not be able to access that interface. When you configure a networking device with a loopback interface and assign it an IP address that is advertised throughout the network, the networking device will be reachable by using this IP address as long as the networking device has at least one network interface capable of sending and receiving IP traffic. In the sample interface configuration and DNS entries for Router A after a loopback interface is configured, you can see that there is now only one DNS entry that can be used to reach the router over any of its physical interfaces.

Router A Interface Configuration After Loopback

```
Loopback 172.16.78.1 255.255.255.0
GigabitEthernet0 10.10.10.1 255.255.255.0
GigabitEthernet1 10.10.11.1 255.255.255.0
GigabitEthernet2 10.10.12.1 255.255.255.0
GigabitEthernet3 10.10.13.1 255.255.255.0
GigabitEthernet4 10.10.14.1 255.255.255.0
GigabitEthernet5 10.10.15.1 255.255.255.0
```

Router A DNS Entries After Loopback

```
RouterA   IN  A  172.16.78.1
```

The configured IP address of the loopback interface--172.16.78.1--can be used as the source address for packets generated by the router and forwarded to networking management applications and routing protocols. Unless this loopback interface is explicitly shut down, it is always reachable.

You can use the loopback interface as the termination address for open shortest path first (OSPF) or border gateway protocol (BGP) sessions. A loopback interface can also be used to establish a Telnet session from the console port of the device to its auxiliary port when all other interfaces are down. In applications where other routers or access servers attempt to reach this loopback interface, you should configure a routing protocol to distribute the subnet assigned to the loopback address.

IP Packets routed to the loopback interface are rerouted back to the router or access server and processed locally. IP packets routed out the loopback interface but not destined to the loopback interface are dropped. Under these two conditions, the loopback interface can behave like a null interface.

Loopback Interfaces Versus Loopback Mode

Loopback interfaces provide a stable source interface to ensure that the IP address assigned to the interface is always reachable as long as the IP routing protocols continue to advertise the subnet assigned to the loopback interface. Loopback mode, however, is used to test and diagnose issues with WAN (serial) links such as bit loss or data corruption. The idea is to configure a loop to return the data packets that were received by the interface back out the same interface to the device that originated the traffic. Loopback mode is used to troubleshoot problems by checking that the data packets are returned in the same condition in which they were sent. Errors in the data packets indicate a problem with the WAN infrastructure. Many types of serial interfaces have their own form of loopback command syntax that is entered under interface or controller configuration mode.

Null Interfaces

The null interface is a virtual network interface that is similar to the loopback interface. Whereas traffic to the loopback interface is directed to the router itself, traffic sent to the null interface is discarded. This interface is always up and can never forward or receive traffic; encapsulation always fails. The null interface functions similarly to the null devices available on most operating systems.

Null interfaces are used as a low-overhead method of discarding unnecessary network traffic. For example, if you do not want your network users to be able to reach certain IP subnets, you can create static IP routes for the subnets that point to the null interface of a networking device. Using the static IP routes takes less CPU time for the networking device than using IP access lists. The static-route configuration is also easier to configure than IP access lists because it is done in global configuration mode instead of in interface configuration mode.

The null interface may not be configured with an address. Traffic can be sent to this interface only by configuring a static route where the next hop is the null interface--represented by Null 0. One example of configuring the next hop to be the null interface is to create a route to an aggregate network that can then be announced through the BGP, or to ensure that traffic to a particular range of addresses is not propagated through the router, perhaps for security purposes.

The router always has a single null interface. By default, a packet sent to the null interface causes the router to respond by sending an Internet Control Message Protocol (ICMP) unreachable message to the source IP address of the packet. You can configure the router either to send these responses or to drop the packets silently.

Subinterfaces

Subinterfaces are associated with physical interfaces. Subinterfaces are enabled when the physical interface with which they are associated is enabled, and subinterfaces are disabled when the physical interface is shut down.



Note Subinterfaces can be enabled and shut down independently of the physical port with which they are associated. However, you cannot enable a subinterface of a physical interface that has been shut down.

Subinterfaces are created by subdividing the physical interface into two or more virtual interfaces on which you can assign unique Layer 3 network addresses such as IP subnets. One of the first uses of subinterfaces was to resolve the problem with split horizon on Frame Relay WANs. Split horizon is a behavior associated with IP routing protocols such as RIP in which IP subnets are not advertised back out the same physical interface that they were learned over. Split horizon was implemented to prevent routing loops in IP networks. A routing loop can be created when the networking devices at both ends of a network connection advertise the same IP routes to each other. Split horizon was an issue for Frame Relay multipoint network interfaces--interfaces that connect to two or more remote networking devices over a single physical interface--because the default behavior of many networking devices was to implement split horizon, which means that the networking device did not advertise the IP routes that were learned over an interface back out the interface to other devices that were also reachable via the same physical interface. Subinterfaces were invented as a method of virtually subdividing a physical interface into two or more interfaces so that the IP routing protocols would see the network connection to each remote networking device as a separate physical interface even though the subinterfaces share a common physical interface. Although TCP/IP now disables split horizon limitations by default, protocols such as AppleTalk and IPX are still constrained by split horizon.

Subinterfaces are identified by a prefix that consists of the hardware interface descriptor (IDB) followed by a period and then by a number that is unique for that prefix. The full subinterface number must be unique to the networking device. For example, the first subinterface for GigabitEthernet interface 0/0/0 might be named GigabitEthernet 0/0/0.1 where .1 indicates the subinterface.

Tunnel Interfaces

Tunneling provides a way to encapsulate arbitrary packets inside a transport protocol. Tunnels are implemented as a virtual interface to provide a simple interface for configuration. The tunnel interface is not tied to specific "passenger" or "transport" protocols, but, rather, it is an architecture that is designed to provide the services necessary to implement any standard point-to-point encapsulation scheme.

There are several ways to implement tunnel interfaces depending on the connectivity that you need to provide. One common use for tunnels is to carry data traffic for a network protocol such as IPX over devices in your

network that do not support IPX. For instance, if your network uses IPX in sites at the edge of your network but not in the core of your network, you can connect the IPX sites at the network edges by tunneling IPX in IP over the core of the network.

For more details about the various types of tunneling techniques available using Cisco IOS XE software, see the " Implementing Tunnels " module of the *Cisco IOS XE Interface and Hardware Component Configuration Guide.*

How to Configure Virtual Interfaces

Configuring a Loopback Interface

This task explains how to configure a loopback interface. A loopback interface can be considered stable because once you enable it, it will remain up until you shut it down. This makes loopback interfaces ideal for assigning Layer 3 addresses such as IP addresses when you want to have a single address to use as a reference that is independent of the status of any of the physical interfaces in the networking device.

Before you begin

The IP address for the loopback interface must be unique and not in use by another interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *number*
4. **ip address** *ip-address mask* [**secondary**]
5. **end**
6. **show interfaces loopback** *number*
7. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface loopback** *number* **Example:** `Router(config)# interface loopback 0`	Specifies a loopback interface and enters interface configuration mode. • Use the *number* argument to specify the number of the loopback interface that you want to create or configure. **Note** There is no limit on the number of loopback interfaces that you can create.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.20.1.2 255.255.255.0`	Specifies an IP address for the loopback interface and enables IP processing on the interface. • Use the *ip-address* and *mask* arguments to specify the subnet for the loopback address.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**show interfaces loopback** *number* **Example:** `Router# show interfaces loopback 0`	(Optional) Displays information about loopback interfaces. • Use the *number* argument to display information about one particular loopback interface. **Note** Only the syntax applicable to this task is used in this example. For more details, see the Cisco IOS Interface and Hardware Component Command Reference.
Step 7	**exit** **Example:** `Router# exit`	Exits privileged EXEC mode.

Examples

The following is sample output for the **show interfaces loopback** command.

```
Router# show interfaces loopback
Loopback0 is up, line protocol is up
  Hardware is Loopback
  Internet address is 10.20.1.2/24
  MTU 1514 bytes, BW 8000000 Kbit, DLY 5000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation LOOPBACK, loopback not set
  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: 0/0 (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 abort
   0 packets output, 0 bytes, 0 underruns
   0 output errors, 0 collisions, 0 interface resets
   0 output buffer failures, 0 output buffers swapped out
```

Configuring a Null Interface

This task explains how to configure a null interface. Null interfaces provide an alternative method to access control lists for filtering traffic. All unwanted traffic can be directed to the null interface; the null interface cannot receive or forward traffic, or allow its traffic to be encapsulated.

The only interface configuration command that you can specify for the null interface is the **no ip unreachables** command.

ICMP Unreachable Messages from Null Interfaces

To disable the sending of ICMP unreachable messages in response to packets sent to the null interface, use the **no ip unreachables** command in interface configuration mode. To reenable the sending of ICMP unreachable messages in response to packets sent to the null interface, use the **ip unreachables** command in interface configuration mode.

You can configure only one null interface on a device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface null** *number*
4. **no ip unreachables**
5. **end**
6. **show interfaces null** [*number*] [**accounting**]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface null** *number* **Example:** `Device(config)# interface null 0`	Specifies a null interface and number, and enters interface configuration mode. • The number argument is always 0.
Step 4	**no ip unreachables** **Example:** `Device(config-if)# no ip unreachables`	Prevents the generation of ICMP unreachable messages on an interface. • This command affects all types of ICMP unreachable messages.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits to privileged EXEC mode.
Step 6	**show interfaces null** [*number*] [**accounting**] **Example:** `Device# show interfaces null 0`	(Optional) Displays information about null interfaces. • For null interfaces, the *number* argument is always 0. **Note** Only the syntax applicable to this task is used in this example. For more details, see the Cisco IOS Interface and Hardware Component Command Reference.

Examples

The following is sample output for the **show interfaces null** command.

```
Device# show interfaces null

Null0 is up, line protocol is up
  Hardware is Unknown
  MTU 1500 bytes, BW 10000000 Kbit, DLY 0 usec,
     reliability 0/255, txload 0/255, rxload 0/255
  Encapsulation ARPA, loopback not set
  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
  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 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Configuring a Subinterface

This task explains how to configure a subinterface. Subinterfaces can be enabled and shut down independently of the physical port with which they are associated. However, you cannot enable a subinterface of a physical interface that has been shut down.

Before you begin

The IP address for the interface must be unique and not in use by another interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number.subinterface-number*
4. **ip address** *ip-address mask* [**secondary**]
5. **end**
6. **show interfaces** *type number.subinterface-number*
7. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number.subinterface-number* **Example:** `Router(config)# interface GigabitEthernet 2/3.5`	Specifies the interface type, interface number, and subinterface number and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 209.165.200.225 255.255.255.224`	Specifies an IP address for the interface and enables IP processing on the interface.
Step 5	**end** **Example:**	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-if)# end`	
Step 6	**show interfaces** *type number.subinterface-number* **Example:** `Router# show interfaces GigabitEthernet 2/3.5`	(Optional) Displays information about the interfaces.
Step 7	**exit** **Example:** `Router# exit`	Exits privileged EXEC mode.

Examples

The following is sample output from the **show interfaces**command:

```
Router# show interfaces GigabitEthernet 2/3.5
GigabitEthernet2/3.5432 is down, line protocol is down (notconnect)
  Hardware is c7600 1Gb 802.3, address is 001b.0de6.c100 (bia 001b.0de6.c100)
  Description: *sample*
  Internet address is 10.11.12.13/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID  2339.
  ARP type: ARPA, ARP Timeout 04:00:00
  Keepalive set (10 sec)
  Last clearing of "show interface" counters never
```

Configuring a Subinterface

This task explains how to configure a subinterface. Subinterfaces can be enabled and shut down independently of the physical port with which they are associated. However, you cannot enable a subinterface of a physical interface that has been shut down.

Before you begin

The IP address for the interface must be unique and not in use by another interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number.subinterface-number*
4. **ip address** *ip-address mask* [**secondary**]
5. **end**
6. **show interfaces** *type number.subinterface-number*
7. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number.subinterface-number* **Example:** `Router(config)# interface GigabitEthernet 2/3.5`	Specifies the interface type, interface number, and subinterface number and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 209.165.200.225 255.255.255.224`	Specifies an IP address for the interface and enables IP processing on the interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**show interfaces** *type number.subinterface-number* **Example:** `Router# show interfaces GigabitEthernet 2/3.5`	(Optional) Displays information about the interfaces.
Step 7	**exit** **Example:** `Router# exit`	Exits privileged EXEC mode.

Examples

The following is sample output from the **show interfaces**command:

```
Router# show interfaces GigabitEthernet 2/3.5
GigabitEthernet2/3.5432 is down, line protocol is down (notconnect)
  Hardware is c7600 1Gb 802.3, address is 001b.0de6.c100 (bia 001b.0de6.c100)
```

```
Description: *sample*
Internet address is 10.11.12.13/24
MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
   reliability 255/255, txload 1/255, rxload 1/255
Encapsulation 802.1Q Virtual LAN, Vlan ID  2339.
ARP type: ARPA, ARP Timeout 04:00:00
Keepalive set (10 sec)
Last clearing of "show interface" counters never
```

Configuring Logical Layer 3 VLAN Interfaces

Before you begin

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

SUMMARY STEPS

1. **configure terminal**
2. **interface vlan** *vlan ID*
3. **ip address** *ip_address subnet_mask*
4. **no shutdown**
5. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface vlan** *vlan ID* **Example:** `Device(config)# interface vlan vlan_ID`	Selects an interface to configure.
Step 3	**ip address** *ip_address subnet_mask* **Example:** `Device(config-if)# ip address ip_address subnet_mask`	Configures the IP address and IP subnet.
Step 4	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Enables the interface.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-if)# end`	Exits the configuration mode.

Examples

The following is sample output from the **show interface vlan**command, which displays the interface IP address configuration and status of Layer 3 VLAN interface vlan 2

```
Device# show interface vlan
 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
```

Configuration Examples for Virtual Interfaces

Example Configuring a Loopback Interface

The following example shows the configuration sequence of a loopback interface, loopback 0:

```
interface loopback 0
 ip address 209.165.200.225 255.255.255.0
 end
```

Example Configuring a Null Interface

The following example shows the configuration sequence of a null interface and how to drop the ICMP unreachable messages. All packets sent to the null interface are dropped and in this example, the ICMP messages usually sent in response to packets being sent to the null interface are dropped.

```
interface null 0
```

```
 no ip unreachables
 end
```

Example Configuring a Subinterface

The following example shows the configuration sequence of a subinterface:

```
interface GigabitEthernet 2/3.5
 description *sample*
 encapsulation dot1Q 2339
 ip address 209.165.200.225 255.255.255.224
end
```

Where to Go Next

- If you want to implement tunnels in your network, see the " Implementing Tunnels " module of the *Cisco IOS XE Interface and Hardware Component Configuration Guide* .
- If you want to implement physical (hardware) interfaces (such as Gigabit Ethernet or serial interfaces) in your network, see the " Configuring Physical Interfaces " module of the *Cisco IOS XE Interface and Hardware Component Configuration Guide* .

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Interface commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS Interface and Hardware Component Command Reference*
Cisco IOS XE Interface and Hardware Component configuration modules	*Cisco IOS XE Interface and Hardware Component Configuration Guide*
Configuration example showing how to use loopback interfaces with BGP	Sample Configuration for iBGP and eBGP With or Without a Loopback Address

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html