

# Configuring Logical Interfaces

---

Use the information in this chapter to understand and configure the types of logical, or virtual, interfaces supported on Cisco routers and access servers. This chapter includes the following configuration instructions and examples:

- Configure a Loopback Interface
- Configure a Null Interface
- Configure a Tunnel Interface

For examples of configuration tasks, see “Logical Interface Configuration Examples” at the end of this chapter.

For hardware technical descriptions and information about installing interfaces, refer to the hardware installation and maintenance publication for your product. For complete descriptions of the logical interface commands, refer to the “Interface Commands” chapter of the *Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

---

**Note** In Cisco IOS Release 11.3, all commands supported on the Cisco 7500 series are also supported on the Cisco 7000 series.

---

## Configure a Loopback Interface

You can specify a software-only interface called a loopback interface to emulate an interface. It is supported on all platforms. A loopback interface is a virtual interface that is always up and allows BGP and RSRB sessions to stay up even if the outbound interface is down.

You can use the loopback interface as the termination address for BGP sessions, for RSRB connections, or to establish a Telnet session from the device’s console to its auxiliary port when all other interfaces are down. You can also use a loopback interface to configure IPX-PPP on asynchronous interfaces. To do so, you must associate an asynchronous interface with a loopback interface configured to run IPX. 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.

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. This means that the loopback interface serves as the Null 0 interface also.

---

**Note** Loopback does not work on an X.21 DTE because the X.21 interface definition does not include a loopback definition.

---

To specify a loopback interface and enter interface configuration mode, perform one of the following tasks in global configuration mode:

Task	Command
Begin interface configuration.	<b>interface loopback</b> <i>number</i>
Begin interface configuration for the Cisco 7200 series or the Cisco 7500 series.	<b>interface loopback</b> <i>slot/port</i>
Begin interface configuration for the Cisco 7500 series.	<b>interface loopback</b> <i>slot/port-adapter/port</i>

See the section “Run Interface Loopback Diagnostics” in the “Overview of Interface Configuration” chapter in this publication.

## Configure a Null Interface

The Cisco IOS software supports a “null” interface. This pseudo-interface functions similarly to the null devices available on most operating systems. This interface is always up and can never forward or receive traffic; encapsulation always fails. The only interface configuration command that you can specify for the null interface is **no ip unreachable**s.

The null interface provides an alternative method of filtering traffic. You can avoid the overhead involved with using access lists by directing undesired network traffic to the null interface.

To specify the null interface, perform the following task in global configuration mode:

Task	Command
Begin interface configuration.	<b>interface null 0</b>

Specify null 0 (or null0) as the interface type and number. The null interface can be used in any command that has an interface type as an argument. The following example configures a null interface for IP route 127.0.0.0:

```
ip route 127.0.0.0 255.0.0.0 null 0
```

## Configure a Tunnel Interface

Tunneling provides a way to encapsulate arbitrary packets inside of a transport protocol. This feature is 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. Because tunnels are point-to-point links, you must configure a separate tunnel for each link.

Tunneling has the following three primary components:

- Passenger protocol, which is the protocol you are encapsulating (AppleTalk, Banyan VINES, CLNP, DECnet, IP, or IPX)
- Carrier protocol, which is one of the following encapsulation protocols:
  - Generic route encapsulation (GRE), Cisco's multiprotocol carrier protocol
  - Cayman, a proprietary protocol for AppleTalk over IP
  - EON, a standard for carrying CLNP over IP networks
  - NOS, IP over IP compatible with the popular KA9Q program
  - Distance Vector Multicast Routing Protocol (DVMRP) (IP in IP tunnels, defined by RFC 20036)
- Transport protocol, which is the protocol used to carry the encapsulated protocol (IP only)

Figure 34 illustrates IP tunneling terminology and concepts.

### Figure 34 IP Tunneling Terminology and Concepts

Normal packet

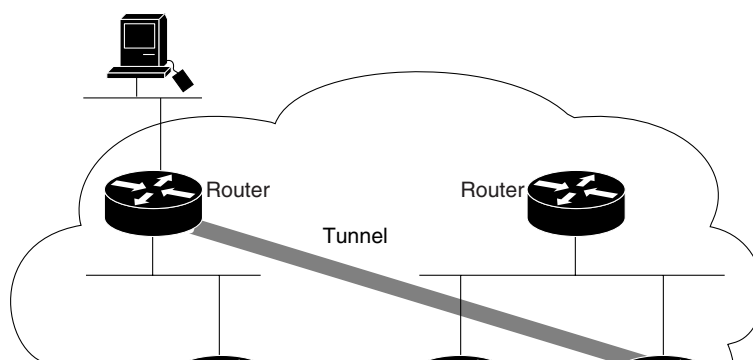
To understand the process of tunneling, consider connecting two AppleTalk networks with a non-AppleTalk backbone, such as IP. The relatively high bandwidth consumed by the broadcasting of Routing Table Maintenance Protocol (RTMP) data packets can severely hamper the backbone's network performance. This problem can be solved by tunneling AppleTalk through a foreign protocol, such as IP. Tunneling encapsulates an AppleTalk packet inside the foreign protocol packet, which is then sent across the backbone to a destination router. The destination router then de-encapsulates the AppleTalk packet and, if necessary, routes the packet to a normal AppleTalk network. Because the encapsulated AppleTalk packet is sent in a directed manner to a remote IP address, bandwidth usage is greatly reduced. Furthermore, the encapsulated packet benefits from any features normally enjoyed by IP packets, including default routes and load balancing.

### Advantages of Tunneling

The following are several situations where encapsulating traffic in another protocol is useful:

- To provide multiprotocol local networks over a single-protocol backbone
- To provide workarounds for networks containing protocols that have limited hop counts; for example, AppleTalk (see Figure 35)
- To connect discontinuous subnetworks
- To allow virtual private networks across wide-area networks (WANs)

**Figure 35** Providing Workarounds for Networks with Limited Hop Counts



### Special Considerations

The following are considerations and precautions to observe when you configure tunneling:

- Encapsulation and decapsulation at the tunnel end points are slow operations; in general, only processor switching is supported. However, fast switching of GRE tunnels was introduced in version 11.1 for the Cisco 2500 series and the Cisco 4000 series of routers.
- Consider security and topology issues. Be careful not to violate access control lists. You can configure a tunnel with a source and destination that are not restricted by firewall routers.
- Tunneling might create problems with transport protocols with limited timers (for example, DECnet) due to increased latency.
- Be aware of the environments across which you create tunnels. You might be tunneling across fast FDDI rings or through slow 9600-bps phone lines; some passenger protocols behave poorly in mixed media networks.
- Multiple point-to-point tunnels can saturate the physical link with routing information.

- Routing protocols that make their decisions based solely on hop count will often prefer a tunnel over a multipoint real link. A tunnel might appear to be a one-hop, point-to-point link and have the lowest-cost path, but may actually cost more. For example, in the topology shown in Figure 36, packets from Host 1 will travel across networks w, q, and z to get to Host 2 instead of taking the path w, x, y, z because it “appears” shorter.

**Figure 36 Tunnel Precautions: Hop Counts**

- An even worse problem will occur if routing information from the tunneled network mixes with the transport networks’ information. In this case, the best path to the “tunnel destination” is via the tunnel itself. This is called a recursive route and will cause the tunnel interface to temporarily shut down. To avoid recursive routing problems, keep passenger and transport network routing information disjointed:
  - Use a different AS number or tag.
  - Use a different routing protocol.
  - Use static routes to override the first hop (but watch for routing loops).
- If you see line protocol down, as in the following example, it might be because of a recursive route:

```
%TUN-RECURDOWN Interface Tunnel 0
temporarily disabled due to recursive routing
```

## IP Tunneling Configuration Task List

If you want to configure IP tunneling, you must perform the tasks in the following sections:

- Specify the Tunnel Interface
- Configure the Tunnel Source
- Configure the Tunnel Destination

The tasks in the following tunnel configuration sections are optional:

- Configure the Tunnel Mode
- Configure End-to-End Checksumming
- Configure a Tunnel Identification Key
- Configure a Tunnel Interface to Drop Out-of-Order Datagrams

- Configure Asynchronous Host Mobility

For commands that monitor IP tunnels, see the section “Monitor and Maintain the Interface” in the “Overview of Interface Configuration” chapter. For examples of configuring tunnels, see the section “IP Tunneling Examples” at the end of this chapter.

### Specify the Tunnel Interface

To specify a tunnel interface and enter interface configuration mode, perform one of the following tasks in global configuration mode:

Task	Command
Begin interface configuration.	<b>interface tunnel</b> <i>number</i>
Begin interface configuration for the Cisco 7500 series.	<b>interface tunnel</b> <i>slot/port</i>

### Configure the Tunnel Source

You must specify the tunnel interface’s source address by performing the following task in interface configuration mode:

Task	Command
Configure the tunnel source.	<b>tunnel source</b> { <i>ip-address</i>   <i>type number</i> }

**Note** You cannot have two tunnels using the same encapsulation mode with exactly the same source and destination address. The workaround is to create a loopback interface and source packets off of the loopback interface.

### Configure the Tunnel Destination

You must specify the tunnel interface’s destination by performing the following task in interface configuration mode:

Task	Command
Configure the tunnel destination.	<b>tunnel destination</b> { <i>hostname</i>   <i>ip-address</i> }

### Configure the Tunnel Mode

The encapsulation mode for the tunnel interface defaults to generic route encapsulation (GRE), so this task is considered optional. However, if you want a mode other than GRE, you must configure it by performing the following task in interface configuration mode:

Task	Command
Configure the tunnel mode.	<b>tunnel mode</b> { <b>aurp</b>   <b>cayman</b>   <b>dvmrp</b>   <b>eon</b>   <b>gre ip</b>   <b>nos</b> }

If you are tunneling AppleTalk, you must use either the AppleTalk Update Routing Protocol (AURP), Cayman or GRE tunneling mode. Cayman tunneling is designed by Cayman Systems and enables routers and access servers to interoperate with Cayman GatorBoxes. You can have Cisco

devices at either end of the tunnel, or you can have a GatorBox at one end and Cisco router or access server at the other end. Use Distance Vector Multicast Routing Protocol (DVMRP) mode when a router or access server connects to a mrouter to run DVMRP over a tunnel. It is required to configure Protocol-Independent Multicast (PIM) and an IP address on a DVMRP tunnel.



**Caution** Do not configure a Cayman tunnel with an AppleTalk network address.

If you use GRE, you must have only Cisco routers or access servers at both ends of the tunnel connection. When you use GRE to tunnel AppleTalk, you must configure an AppleTalk network address and a zone. Perform the following tasks to tunnel AppleTalk using GRE:

Task	Command
<b>Step 1</b> Enable tunneling on the interface.	<b>interface tunnel</b> <i>number</i>
<b>Step 2</b> Assign a cable range to an interface.	<b>appletalk cable-range</b> <i>start-end</i> [ <i>network.node</i> ]
<b>Step 3</b> Set a zone name for the connected AppleTalk network.	<b>appletalk zone</b> <i>zone-name</i>
<b>Step 4</b> Specify the interface out which the encapsulated packets will be sent, or specify the router's IP address.	<b>tunnel source</b> { <i>ip-address</i>   <i>type number</i> }
<b>Step 5</b> Specify the IP address of the router at the far end of the tunnel.	<b>tunnel destination</b> { <i>hostname</i>   <i>ip-address</i> }
<b>Step 6</b> Enable GRE tunneling.	<b>tunnel mode gre ip</b>

## Configure End-to-End Checksumming

Some passenger protocols rely on media checksums to provide data integrity. By default, the tunnel does not guarantee packet integrity. By enabling end-to-end checksums, the Cisco IOS software drops corrupted packets. To enable such checksums on a tunnel interface, perform the following task in interface configuration mode:

Task	Command
Configure end-to-end checksumming.	<b>tunnel checksum</b>

## Configure a Tunnel Identification Key

You can optionally enable an ID key for a tunnel interface. This key must be set to the same value on the tunnel endpoints. Tunnel ID keys can be used as a form of *weak* security to prevent misconfiguration or injection of packets from a foreign source.

The tunnel ID key is available with GRE only.

---

**Note** When using GRE, the ID key is carried in each packet. We do *not* recommend relying on this key for security purposes.

---

## Configure a Tunnel Interface

---

To configure a tunnel ID key, perform the following task in interface configuration mode:

Task	Command
Configure a tunnel identification key.	<b>tunnel key</b> <i>key-number</i>

## Configure a Tunnel Interface to Drop Out-of-Order Datagrams

You can optionally configure a tunnel interface to drop datagrams that arrive out of order. This is useful when carrying passenger protocols that behave poorly when they receive packets out of order (for example, LLC2-based protocols). This option is available with GRE only.

To use this option, perform the following task in interface configuration mode:

Task	Command
Configure a tunnel interface to drop out-of-order datagrams.	<b>tunnel sequence-datagrams</b>

## Configure Asynchronous Host Mobility

Increasingly, remote users are accessing networks through dial-up telephone connections. In contrast to local users who can connect directly into the network, remote users must first dial into an access server.

The access server supports a packet tunneling strategy that extends the internetwork—in effect creating a virtual private link for the mobile user. When a user activates asynchronous host mobility, the access server on which the remote user dials into becomes a remote point-of-presence (POP) for the user's home network. Once logged in, users experience a server environment identical to the one that they experience when they connect directly to the “home” access server.

Once the network layer connection is made, data packets are tunneled at the physical and/or data link layer instead of at the protocol layer. In this way, raw data bytes from dial-in users are transported directly to the “home” access server, which processes the protocols.

Figure 37 illustrates the implementation of asynchronous host mobility on an extended internetwork. A mobile user connects to an access server on the internetwork and, by activating asynchronous host mobility, is connected to a “home” access server configured with the appropriate user name. The user sees an authentication dialog or prompt from the “home” system and can proceed as if he or she were connected directly to that device.

**Figure 37 Asynchronous Host Mobility**

The remote user implements asynchronous host mobility by executing the **tunnel** command in the User EXEC mode. The **tunnel** command sets up a network layer connection to the specified destination. The access server accepts the connection, attaches it to a virtual terminal (VTY), and runs a command parser capable of running the normal dial-in services. After the connection is established, data is transferred between the modem and network connection with a minimum of interpretations. When communications are complete, the network connection can be closed and terminated from either end.

Refer to the *Cisco Access Connection Guide* for information about setting up the network layer connection with the **tunnel** command.

## Logical Interface Configuration Examples

This section provides examples that illustrate configuration of IP tunnels.

### IP Tunneling Examples

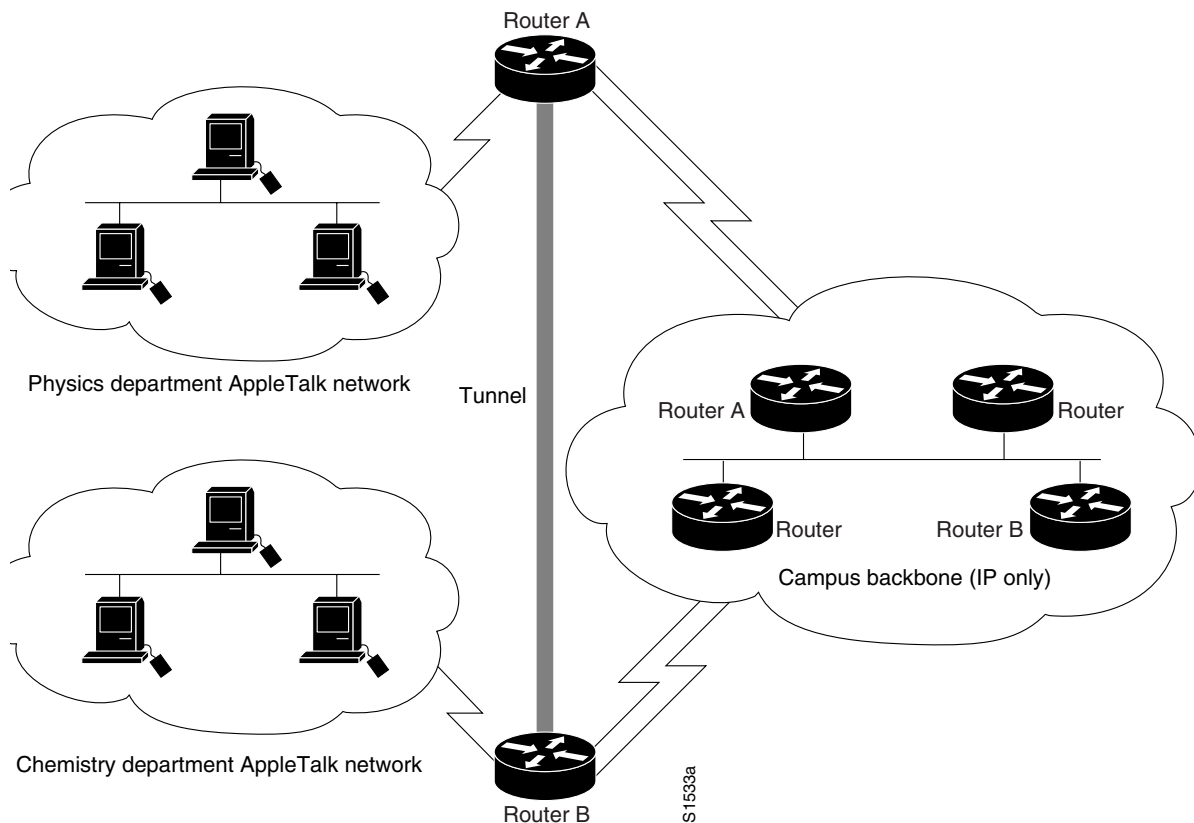
The following example shows an IP tunneling configuration with commented (!) explanations:

```
!Creates the interface
interface tunnel 0
!enables IPX on the interface
novell network 1e
!enables appletalk
appletalk cable-range 4001-4001 128
!enables IP
ip address 10.1.2.3. 255.255.255.0
!enables DECnet
DECnet cost 4
!sets the source address, or interface, for packets
tunnel source ethernet 0
!determines where the encapsulated packets are to go
tunnel destination 131.108.14.12
!sets the protocol
tunnel mode gre
!computes a checksum on passenger packets if protocol doesn't already have reliable
!checksum
tunnel checksum needed
!sets the id key
tunnel key 42
!set to drop out of order packets
tunnel sequence-datagrams
```

### Routing Two AppleTalk Networks across an IP-Only Backbone Example

Figure 38 is an example of connecting multiprotocol subnetworks across a single-protocol backbone. The configurations of Router A and Router B follow.

**Figure 38 Connecting AppleTalk Networks across an IP-Only Backbone**



#### Router A

```

interface ethernet 0
  description physics department AppleTalk lan
  appletalk cable-range 4001-4001 32
  appletalk zone ethernet
interface fddi 0
  description connection to campus backbone
  ip address 36.0.8.108 255.255.255.0
interface tunnel 0
  tunnel source fddi 0
  tunnel destination 36.0.21.20
  appletalk cable-range 5313-5313 1
  appletalk zone tunnel
  
```

### Router B

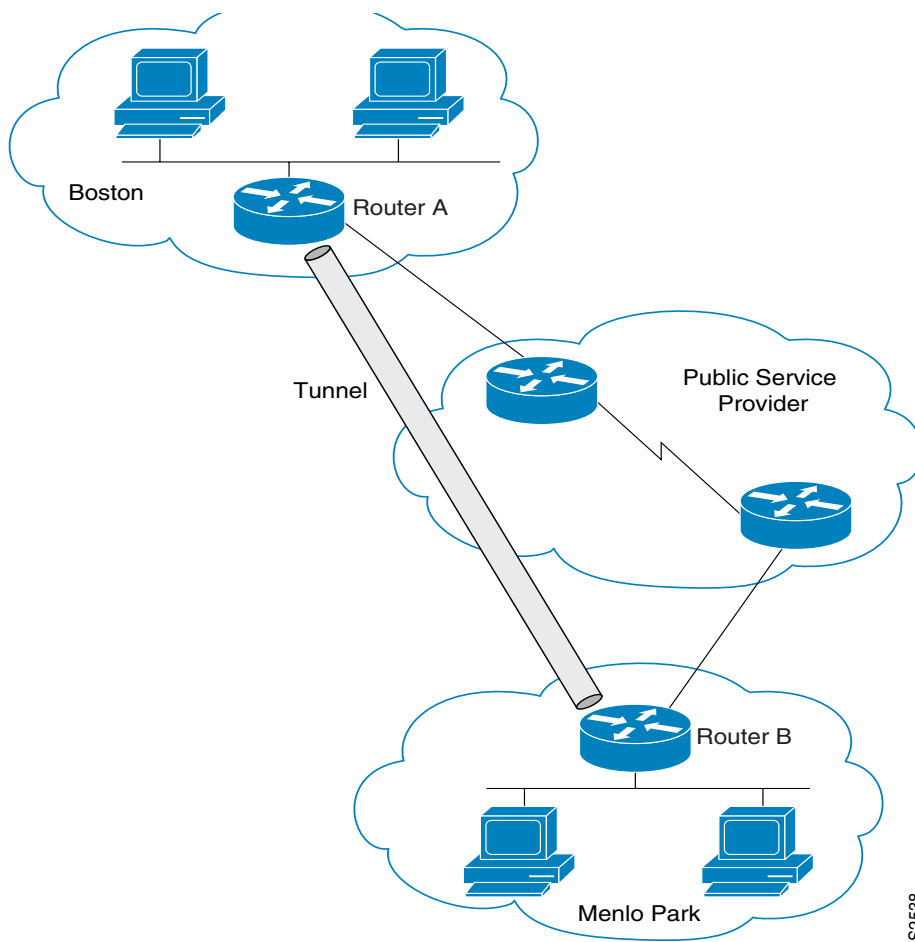
```

interface ethernet 0
  description chemistry department appletalk lan
  appletalk cable-range 9458-9458 3
  appletalk zone ethernet
interface fddi 0
  description connection to campus backbone
  ip address 36.0.21.20 255.255.255.0
interface tunnel 0
  tunnel source fddi 0
  tunnel destination 36.0.8.108
  appletalk cable-range 5313-5313 2
  appletalk zone tunnel
  
```

### Routing a Private IP Network and a Novell Net across a Public Service Provider Example

Figure 39 is an example of routing a private IP network and a Novell network across a public service provider.

**Figure 39 Creating Virtual Private Networks across WANs**



S2538

### Router A

```
interface ethernet 0
  description boston office
  ip address 10.1.1.1 255.255.255.0
  novell network 1e
!
interface serial 0
  description connection to NEARnet
  ip address 192.13.2.1 255.255.255.0
!
interface tunnel 0
  tunnel source serial 0
  tunnel destination 131.108.5.2
  ip address 10.1.2.1 255.255.255.0
  novell network 1f
```

### Router B

```
interface ethernet 0
  description menlo park office
  ip address 10.1.3.1 255.255.255.0
  novell network 31
!
interface serial 4
  description connection to BARRnet
  ip address 131.108.5.2 255.255.255.0
!
interface tunnel 0
  tunnel source serial 4
  tunnel destination 192.13.2.1
  ip address 10.1.2.2 255.255.255.0
  novell network 1f
```