CHAPTER 6

Configuring Interfaces

Use the information in this chapter to understand the types of interfaces supported on our routers. Our routers support two types of interfaces: physical and virtual interfaces. The physical types of interfaces you have depend on the appliques or interface processors you have. The virtual interfaces our routers support include subinterfaces and IP tunnels.

Our routers support the following types of interfaces:

- Asynchronous serial
- Asynchronous Transfer Mode (ATM)
- Channelized E1
- Channelized T1
- Dialer
- Ethernet
- Fiber Distributed Data Interface (FDDI)
- High-Speed Serial Interface (HSSI)
- ISDN Basic Rate Interface (BRI)
- ISDN MultiBasic Rate Interface (MBRI)
- ISDN Primary Rate Interface (PRI)
- LAN Extender
- Loopback
- Null
- Synchronous serial
- Token Ring
- Tunnel

In addition to the interface types, the router supports subinterfaces. See each protocol chapter for specific information on how to configure a subinterface for that protocol.

For hardware technical descriptions and information about installing the router interfaces, refer to the hardware installation and maintenance publication for your product. For command descriptions and usage information, refer to the "Interface Commands" chapter of the *Router Products Command Reference* publication. For a conversion table of the modular products and Cisco 7000 processors, refer to the "Cisco 7000 Processors" appendix in the *Router Products Command Reference* publication.

Interface Configuration Task List

You can perform the tasks in the following sections to configure and maintain the interfaces supported on our routers:

- Understand Interface Configuration
- Configure an Asynchronous Serial Interface
- Configure an ATM Interface
- Configure Channelized E1
- Configure Channelized T1
- Configure a Dialer Interface
- Configure an Ethernet Interface
- Configure a Fiber Distributed Data Interface (FDDI)
- Configure a High-Speed Serial Interface (HSSI)
- Configure a Hub Interface
- Configure an ISDN Basic BRI, MBRI, or PRI Interface
- Configure a LAN Extender Interface
- Configure a Loopback Interface
- Configure a Null Interface
- Configure a Synchronous Serial Interface
- Configure a Token Ring Interface
- Configure PCbus Token Ring Interface Management
- Configure a Tunnel Interface
- Understand Subinterfaces
- Configure Features Available on Any Interface
- Configure Dial Backup Service
- Understand Online Insertion and Removal (OIR)
- Understand Fast, Autonomous, and SSE Switching Support
- Monitor and Maintain the Interface

Note For information about the Channel Interface Processor (CIP), see the chapter entitled "Configuring IBM Channel Attach." The CIP is described in a separate chapter because of the interrelation of host system configuration values and router configuration values.

See the end of this chapter for "Interface Configuration Examples."

Understand Interface Configuration

Begin interface configuration in global configuration mode. To configure an interface, follow these steps:

Step 1 Enter the **configure** EXEC command at the privileged EXEC prompt to enter global configuration mode.

Step 2 Once in the global configuration mode, start configuring the interface by entering the **interface** command. Identify the interface type followed by the number of the connector or interface card. These numbers are assigned at the factory at the time of installation or when added to a system and can be displayed with the **show interfaces** EXEC command. A report is provided for each interface the router supports, as seen in the following partial sample display:

```
Serial 0 is administratively down, line protocol is down
Hardware is MCI Serial
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation HDLC, loopback not set, keepalive set (10 sec)
```

Use the **show hardware** EXEC command to see a list of the system software and hardware.

For example, to begin configuring serial interface 0, you would add the following line to the configuration file:

```
interface serial 0
```

Note It is not necessary to add a space between the interface type and interface number. For example, in the preceding line you can specify either serial 0 or serial0.

Step 3 Follow each **interface** command with the interface configuration commands your particular interface requires. These command define the protocols and applications that will run on this interface. The commands are collected and applied to the **interface** command until you enter another **interface** command, a command that is not an interface configuration command, or you type the Ctrl-Z sequence to get out of configuration mode and return to privileged EXEC mode.

Step 4 Once an interface is configured, you can check its status by entering the EXEC **show** commands described after the task tables that follow.

Note When you configure channelized T1, you must first define the channels and the timeslots that comprise the channels by using the **controller t1** and the **channel-group** controller configuration commands. Then configure the virtual serial interfaces using the **interface serial** global configuration commands. See the section “Configure Channelized T1” later in this chapter for T1 configuration tasks.

The following sections show how to configure each interface type. Follow the **interface** command with the routing or bridging interface configuration commands for your particular protocol or application, as described in this chapter and subsequent chapters.

See the section “Examples of Enabling Interface Configuration” at the end of this chapter.

Configure an Asynchronous Serial Interface

All of our router platforms configured with an auxiliary port support the asynchronous serial interface. To configure an asynchronous serial interface on the router, you must establish asynchronous serial line connections using PPP or SLIP. PPP and SLIP define methods of sending Internet packets over a standard RS-232 asynchronous serial line. PPP also defines methods for sending IPX packets.

Asynchronous Serial Task List

To use the asynchronous device as a network interface via PPP or SLIP, complete the tasks in the following sections:

- Specify Asynchronous Serial Interface 1
- Configure Asynchronous Serial Encapsulation
- Configure the Addressing Method
- Configure DHCP
- Configure Dedicated or Interactive Mode
- Enable Asynchronous Routing
- Connect to Remote Routers via PPP or SLIP

Note You can also configure support for SLIP and PPP using extended BOOTP requests. See the chapter entitled "Loading System Images, Microcode Images, and Configuration Files."

Specify Asynchronous Serial Interface 1

Only the auxiliary port on a router can be configured as an asynchronous serial interface. To configure an asynchronous serial interface on the router, you must establish asynchronous serial line connections using PPP or SLIP, as described in the next section.

The auxiliary port's absolute line number is 1. When you configure an asynchronous serial interface with the **interface async 1** command, you enable asynchronous routing over the auxiliary port to support PPP and SLIP connections to remote routers. The interface number is the same as the absolute line number.

The router automatically associates the interface number 1 with the absolute line number 1 of the auxiliary port, and treats the interface as an asynchronous line. However, to configure the auxiliary port as an asynchronous interface, you must also configure it as an auxiliary line with the **line aux 1** command as described in the chapter entitled "Configuring Terminal Lines and Modem Support." Follow the **line** command with the appropriate line configuration commands for modem control, such as speed. Perform the following task in global configuration mode to specify the auxiliary port line as an asynchronous interface:

Task	Command
Specify an asynchronous serial interface.	**interface async 1**

Only IP packets can be sent across lines configured for SLIP. PPP supports transmission of both IP and IPX packets.

Configure Asynchronous Serial Encapsulation

There are two asynchronous serial encapsulation methods:

- SLIP
- Asynchronous PPP

SLIP and PPP are methods of encapsulating datagrams and other network-layer protocol information over point-to-point links. SLIP is the default method. Perform the following task in interface configuration mode to configure PPP or SLIP encapsulation on the asynchronous interface:

Task	Command
Configure PPP or SLIP encapsulation on an asynchronous line.	**encapsulation** {**ppp** | **slip**}

The configured SLIP or PPP encapsulation method applies to an interface configured for *dedicated* asynchronous mode or dial-on-demand routing (DDR). On an asynchronous interface configured for *interactive* mode, the encapsulation type is specified by the user with the **slip** or **ppp** EXEC command. See the "Configure Dedicated or Interactive Mode" section later in this section.

In order to use SLIP or PPP, the router must be configured with an IP routing protocol or with the **ip host-routing** command. This configuration is done automatically if you are using old-style **slip address** commands. However, you must configure it manually if you configure SLIP or PPP via the **interface async** command.

See the section "Configure PPP" in the section "Configure a Synchronous Serial Interface" later in this chapter for more information about PPP.

Configure the Addressing Method

You can control whether a user must specify an address when making a SLIP or PPP connection or whether the address is forced by the system. Using an address defined by the system is referred to as default addressing. Requiring the user to specify an address is called dynamic addressing. It is common to configure an asynchronous interface both to have a default address and to allow dynamic addressing.

This section describes how to do the following:

- Assign a Default Asynchronous Address
- Allow an Asynchronous Address to be Assigned Dynamically

Assign a Default Asynchronous Address

You can assign a permanent default asynchronous address to a line by performing the following task in interface configuration mode:

Task	Command
Assign a default IP address to the asynchronous interface.	**async default ip address** *ip-address*

Use the **no** form of this command to disable the default address.

The assigned default address is used when the user enters the **slip default** or **ppp default** EXEC command. The TACACS server validates the transaction (when enabled), and the line is put into network mode using the address that is in the configuration file. This feature is useful when the user is not required to know the IP address to gain access to a system; for example, users of a server that is available to many students on a campus.

Allow an Asynchronous Address to be Assigned Dynamically

When a line is configured for dynamic assignment of asynchronous addresses, the user enters the **slip** or **ppp** EXEC command and is prompted for an address or logical host name. The TACACS validates the address, when enabled, and the line is assigned the given address and put into asynchronous mode. Assigning asynchronous addresses dynamically is also useful when you want to assign set addresses to users. For example, an application on a personal computer that automatically dials in using SLIP and polls for electronic mail messages can be set up to dial in periodically and enter the required IP address and password.

To configure asynchronous dynamic addressing, perform the following task in interface configuration mode:

Task	Command
Allow the IP address to be assigned at login.	**async dynamic address**

The dynamic addressing features of the internetwork allow packets to get to their destinations and back regardless of the router or network they are sent from. For example, if a host such as a laptop computer moves from place to place, it can keep the same address no matter where it is dialing in from. For an example of configuring asynchronous dynamic addressing, see the section "Example of Asynchronous Routing and Dynamic Addressing" at the end of this chapter.

Configure DHCP

The Dynamic Host Configuration Protocol (DHCP) model consists of the following components:

- A DHCP server—A host-based DHCP server configured to accept and process requests for temporary IP addresses.
- A DHCP proxy-client—A Cisco router configured to arbitrate DHCP calls between the DHCP server and the DHCP client.
- The DHCP client—The end user who is dialing into the router on an asynchronous line using SLIP or PPP.

The DHCP client-proxy feature manages a pool of IP addresses available to PPP or SLIP dial-in clients without a known IP address. This pool allows a finite number of IP addresses to be reused quickly and efficiently by many clients. Additional benefits include the ability to maintain sessions, such as Telnet, even when a modem line fails. When the client is auto-dialed back into the server, the session can be resumed because the same IP address is reissued to the client by the server.

You can designate all of the router's asynchronous interfaces to use DHCP or can turn off DHCP on individual interfaces. Cisco's implementation of DHCP complies with RFC 1541, and is compliant with extended TACACS.

To enable DHCP on a router's asynchronous interfaces, perform the following tasks, starting in global configuration mode:

Task		Command
Step 1	Specify that the router use the DHCP client-proxy feature on all asynchronous interfaces.	**ip address-pool dhcp-proxy-client**
Step 2	(Optional) Specify the IP address of at least one and up to 10 DHCP servers for the proxy-client (the Cisco router) to use. DHCP servers provide temporary IP addresses.	**ip dhcp-server** [i*p-address* | *name*]

Task		Command
Step 3	(Optional) Enter interface configuration mode.	**interface async** *number*
Step 4	(Optional) Turn off DHCP on any asynchronous interfaces on the router, as needed.	**no peer default ip address pool**

Configure Dedicated or Interactive Mode

You can configure the asynchronous interface to be in dedicated network or interactive mode.

In dedicated mode, there is no user prompt or EXEC level, so no end-user commands are required to place the line into interface mode. When the interface is configured for dedicated mode, the user cannot change the encapsulation method, address, or other parameters.

To configure an asynchronous interface to be in dedicated network mode, perform the following task in interface configuration mode:

Task	Command
Place the asynchronous line into dedicated network mode.	**async mode dedicated**

For an example of placing an asynchronous interface into dedicated network mode, see the section "Example of a Dedicated Asynchronous Interface" at the end of this chapter.

Alternatively, you can configure an asynchronous line for interactive mode. In interactive mode, the line can be used to make any type of connection, depending on the EXEC command entered by the user. For example, depending on its configuration, the line could be used for Telnet connections, or SLIP or PPP encapsulation. Perform the following task in interface configuration mode to configure an asynchronous line for interactive mode:

Task	Command
Place the asynchronous line in interactive mode.	**async mode interactive**

Enable Asynchronous Routing

You can enable use of dynamic routing protocols on the asynchronous interface by performing the following task in interface configuration mode:

Task	Command
Configure an asynchronous interface for routing.	**async dynamic routing**

Connect to Remote Routers via PPP or SLIP

You can use an asynchronous device as a network interface connection to a remote router via the auxiliary port using the PPP or SLIP protocols. Refer to the *Cisco Access Connection Guide*.

Configure an ATM Interface

See the "Configuring ATM" chapter for information on how to configure an Asynchronous Transfer Mode (ATM) interface.

See also the section “Invoke ATM over a Serial Line” in the section “Configure a Synchronous Serial Interface” later in this chapter.

Configure Channelized E1

Support for channelized E1 is provided on the following platforms:

- Cisco 7000 series—by means of a MultiChannel Interface Processor (MIP) and a CxBus channelized E1 adapter (CX-MIP-CE1). The Cisco 7000 MIP can support one or two CX-MIP-CE1 adapters.
- Cisco 4000 series—by means of up to three channelized E1 controllers, each providing one physical interface (adapter) to the network when running as a channelized interface card. (When used to run ISDN PRI, only one Network Processor Module (NPM) can be used in the Cisco 4000 and two NPMs can be used in the Cisco 4500.)

Each E1 adapter can support a maximum of 30 channel groups. The Cisco 7000 MIP can support one or two adapters, providing a maximum of 60 channel groups per MIP. The Cisco 4000 can support one adapter, providing a maximum of 30 channel groups. Each channel group is presented to the system as a serial interface that can be configured individually. In effect, up to 30 E1 circuits are multiplexed to each hardware adapter.

Use the **show controllers e1** EXEC command to display current E1 status. This command provides a report for each physical interface configured to support channelized E1.

Channelized E1 supports the following WAN protocols:

- X.25
- LAPB
- Frame Relay
- PPP
- HDLC
- SMDS
- ATM-DXI

When a channelized E1 adapter is used for ISDN PRI, it can support DDR; when it is not used for ISDN PRI, it does not support DDR. Refer to the “Configuring ISDN” chapter of this manual for more information.

Channelized E1 Task List

Using channelized E1 controller and serial interface configuration commands, you can perform the tasks in the following sections to configure channelized E1:

- Configure the E1 Controller
- Define the Line Code
- Define the Framing Characteristics
- Define the E1 Channel Groups
- Configure the Channelized E1 Channel Groups

See the end of this chapter for “Interface Configuration Examples.”

Configure the E1 Controller

To configure the E1 physical characteristics, you first define the location of the controller in the router. A Cisco 7000 router can have up to four MIP and eight CX-MIP-CE1 interfaces. A Cisco 7010 router can have up to three MIP and six CX-MIP-CE1 interfaces. The Cisco 4000 series routers can support up to three network processor module (NPM) cards, each with one interface when running it as a channelized interface card. However, when the card is used to run ISDN PRI, only one NPM can be used in the Cisco 4000 and two NPMs can be used in the Cisco 4500.

Note You can define ISDN PRI groups and channel groups on the same controller. However, you cannot overlap timeslots or use the ISDN D-channel timeslot in a channel group.

Perform the following task in global configuration mode to define the E1 controller and to enter controller configuration mode:

Task	Command
Define the controller location in the Cisco 7000 series by slot and port number.	**controller e1** *slot*/*port*
or	
Define the controller location in the Cisco 4000 series by unit number, ranging from 0 through 2.	**controller e1** *number*

Define the Line Code

Perform the following task in controller configuration mode to define the line code as either alternate mark inversion (AMI) or HDB3:

Task	Command
Define the line code as either AMI or HDB3; HDB3 is the default.	**linecode** {**ami** | **hdb3**}

Contact your local telephone service provider to determine the line-code requirements of the physical E1 line. The E1 controller values must match the service provided by the telephone company.

Define the Framing Characteristics

Perform the following task in controller configuration mode to define the framing characteristics as either CRC4 or no-CRC4, or as the version of E1 framing used in Australia only:

Task	Command
Define the framing characteristics as either CRC4 or no-CRC4.	**framing** {**crc4** | **no-crc4**} [**australia**]

Contact your local telephone service provider to determine the framing requirements of the physical E1 line. The E1 controller values must match the service provided by the telephone company.

Define the E1 Channel Groups

You can define up to 30 channel groups for each E1 adapter. You must define the timeslots that belong with each channel group. Channel groups are numbered 0 to 30, and timeslots are numbered 1 to 31. Perform the following task in controller configuration mode to define the channel groups and timeslots:

Task	Command
Define the channel group number and, if needed, circuit speed.	**channel-group** *number* **timeslots** *range* [**speed** {**48** | **56** | **64**}]

Working with your local service provider, you can create channel-groups with from one to 31 timeslots. These timeslots can be in any order, contiguous or noncontiguous. Channel-group speeds can be 48 kbps, 56 kbps, or 64 kbps; the default is 64 kbps if the speed is not specified. The speed you choose must match the speed specified by your service provider. 7

Defining a channel group creates a serial interface; defining multiple channel groups creates an equal number of serial interfaces that you can configure independently. The channel group numbers for each E1 controller can be arbitrarily assigned.

Configure the Channelized E1 Channel Groups

After you define the E1 channel groups, you can configure each channel group as a serial interface. In other words, you can think of each channel group as a virtual serial interface. Subinterface configuration on the created interface is also supported. Perform the following task either in global configuration mode or controller configuration mode to enter interface configuration mode and configure the serial interface that corresponds to a channel group:

Task	Command
Define the serial interface for an E1 channel group.	**interface serial** *slot*/*port*:*channel-group* (Cisco 7000 series)
	interface serial *number*:*channel-group* (Cisco 4000 series)

E1 channel groups support local loopback. You can enable local loopback for specified individual channel groups with the **loopback local** command. Local loopback loops the entire specified channel group both toward the line and toward the router.

E1 channel groups do not respond to any remote loopback codes. That is, you cannot remotely loop an E1 channel group.

Configure Channelized T1

Support for channelized T1 (also referred to as *fractional* T1) is provided on the following platforms:

- Cisco 7000 series by means of a MultiChannel Interface Processor (MIP) and a CxBus channelized T1 adapter (CxCT1). The Cisco 7000 MIP can support one or two CxCT1 adapters.
- Cisco 4000 series by means of a single channelized T1 adapter.

Each T1 adapter can support a maximum of 24 DS0 channel groups. Each channel group is presented to the system as a serial interface that can be configured individually. The Cisco 7000 MIP can support one or two CxCT1 adapters, providing a maximum of 48 channel groups per MIP. The Cisco 4000 supports a one adapter, providing a maximum of 24 channel groups. In effect, up to 24 DS0 circuits are multiplexed to a single hardware adapter.

Use the **show controllers t1** EXEC command to display current T1 status. This command provides a report for each physical interface configured to support channelized T1.

Channelized T1 supports the following WAN protocols:

- X.25
- LAPB
- Frame Relay
- PPP
- HDLC
- SMDS
- ATM-DXI

When a channelized T1 adapter is used for ISDN PRI, it can support DDR; when it is not used for ISDN PRI, it does not support DDR. Refer to the "Configuring ISDN" chapter of this manual for more information.

The Cisco channelized T1 controllers require the use of a CSU when connected to a public network. This device should take a T1 signal from the public network and provide a T1 signal to the channelized T1 controller.

Channelized T1 Task List

Using channelized T1 controller and serial interface configuration commands, you can perform the tasks in the following sections to configure channelized T1:

- Configure the T1 Controller
- Define the Line Code
- Define the Framing Characteristics
- Define the Clock Source
- Define the T1 Channel Groups
- Configure the Channelized T1 Channel Groups

See the end of this chapter for "Interface Configuration Examples."

Configure the T1 Controller

To configure the T1 physical characteristics, you first define the physical location of the MIP and CxCT1 in the Cisco 7000 series router. A Cisco 7000 router can have up to four MIP and eight CxCT1 interfaces. A Cisco 7010 router can have up to three MIP and six CxCT1 interfaces. The Cisco 4000 series routers can support up to three network processor module (NPM) cards, each with one interface when running it as a channelized interface card. However, when the card is used to run ISDN PRI, only one NPM can be used in the Cisco 4000 and two NPMs can be used in the Cisco 4500.

Note You can define ISDN PRI groups and channel groups on the same controller. However, you cannot overlap timeslots or use the ISDN D-channel timeslot in a channel group.

Perform the following task in global configuration mode to define the T1 controller and to enter controller configuration mode:

Task	Command
Define the MIP and CxCT1 locations in the Cisco 7000 series by slot and port number.	**controller t1** *slot*/*port*
or	or
Define the controller location in the Cisco 4000 series by unit number, ranging from 0 through 2.	**controller t1** *number*

Define the Line Code

Perform the following task in controller configuration mode to define the line code as either alternate mark inversion (AMI) or bipolar 8 zero substitution (B8ZS):

Task	Command
Define the line code as either AMI or B8ZS; AMI is the default.	**linecode** {**ami** | **b8zs**}

Contact your local telephone service provider to determine the line-code requirements of the physical T1 line. The T1 controller values must match the service provided by the telephone company.

Define the Framing Characteristics

Perform the following task in controller configuration mode to define the framing characteristics as either super frame (SF) or extended super frame (ESF):

Task	Command
Define the framing characteristics as either SF or ESF; SF is the default.	**framing** {**sf** | **esf**}

Contact your local telephone service provider to determine the framing requirements of the physical T1 line. The T1 controller values must match the service provided by the telephone company.

Define the Clock Source

Under normal usage, skip this step. You must define the clock source only when connecting two devices back-to-back for testing purposes. The clock source normally comes from the T1 line rather than from the router interface, but when you connect two routers back-to-back for testing purposes, one device supplies an internal clock source. To define the clock source, perform the following task in controller configuration mode:

Task	Command
Define the clock source if you are connecting two cards back-to-back for testing purposes; the default source is the **line**.	**clock source** {**line** | **internal**}

Define the T1 Channel Groups

You can define up to 24 channel groups for each T1 adapter.You must define the timeslots that belong with each channel group. Channel groups are numbered 0 to 23, and timeslots are numbered 1 to 24. Perform the following task in controller configuration mode to define the channel groups and timeslots:

Task	Command
Define the channel group number and, if needed, circuit speed.	**channel-group** *number* **timeslots** *range* [**speed** {**48** | **56** | **64**}]

Working with your local service provider, you can create channel-groups with from one to 24 timeslots. These timeslots can be in any order, contiguous or noncontiguous. In the United States, channel-group speeds can be either 56 kbps or 64 kbps; the default is 56 kbps. If 64 kbps is used, it is recommended to be used with framing type of ESF and a linecode of B8ZS. The speed you select must match the speed provided by the telephone company.

Defining a channel group creates a serial interface; defining multiple channel groups creates an equal number of serial interfaces that you can configure independently. The channel group numbers for each T1 controller can be arbitrarily assigned.

Configure the Channelized T1 Channel Groups

After you define the T1 channel groups, you can configure each channel group as a serial interface. In other words, you can think of each channel group as a virtual serial interface. Subinterface configuration is also supported on the created serial interface. Perform the following task either in global configuration mode or controller configuration mode to enter interface configuration mode and configure the serial interface that corresponds to a channel group:

Task	Command
Define the serial interface for a T1 channel group.	**interface serial** *slot*/*port***:***channel-group* (Cisco 7000 series) **interface serial** *number***:***channel-group* (Cisco 4000 series)

Configure a Dialer Interface

See the chapter “Configuring DDR” for information about how to configure a dialer interface.

Configure an Ethernet Interface

Support for the Ethernet interface is supplied on one of the following Ethernet network interface cards or systems:

- The Multiport Communications Interface (MCI) card in the modular routers, which provides one Ethernet connector compatible with Ethernet Versions 1 and 2 and the IEEE 802.3 protocol.
- The Multiport Ethernet Controller (CSC-MEC) interface card in the modular routers, which provides two, four, or six high-speed Ethernet connectors compatible with Ethernet Versions 1 and 2 and the IEEE 802.3 protocol.
- An integrated Ethernet controller on the Cisco 2500 series and Cisco 3000 models.
- An integrated Ethernet controller on the Cisco 1003 model.
- On the Cisco 7000 series, the high-speed Ethernet interface processor (EIP) for two, four, or six AUI ports. The EIP ports are in compliance with Ethernet versions 1 and 2 and the IEEE 802.3 specifications.
- On the Cisco 7000 series, the Fast Ethernet Interface Processor (FEIP) provides a Fast Ethernet interface. The FEIP enables the router to communicate at speeds of 100 Mbps with network devices such as switches, servers, and other routers.

Use the **show interfaces**, **show controllers mci**, and **show controllers cbus** EXEC commands to display the Ethernet port numbers. These commands provide a report for each interface supported by the router.

Use the **show interfaces fastethernet** command to display the Fast Ethernet slots and ports. The FEIP defaults to half-duplex mode and media type 10BaseTX.

The Fast Ethernet encapsulation methods are the same as the Ethernet encapsulation methods. See the section “Etherenet Encapsulation Methods.”

Ethernet Interface Task List

Perform the tasks in the following sections to configure features on an Ethernet interface. The first task is required; the remaining tasks are optional.

- Specify an Ethernet Interface
- Configure Ethernet Encapsulation
- Configure the Ethernet Network Interface Module on the Cisco 4000
- Extend the 10BaseT Capability

Specify an Ethernet Interface

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

Task	Command
Begin interface configuration.	**interface ethernet** *number*
Begin interface configuration for the Cisco 7000 series.	**interface ethernet** *slot*/*port*

Configure Ethernet Encapsulation

Currently, there are three common Ethernet encapsulation methods:

- The standard ARPA Ethernet Version 2.0 encapsulation, which uses a 16-bit protocol type code (the default encapsulation method)
- SAP IEEE 802.3 encapsulation, in which the type code becomes the frame length for the IEEE 802.2 LLC encapsulation (destination and source Service Access Points, and a control byte)
- The SNAP method, as specified in RFC 1042, which allows Ethernet protocols to run on IEEE 802.2 media

The encapsulation method you use depends upon the type of Ethernet media connected to the router and the routing or bridging application you configure. Establish Ethernet encapsulation by performing one of the following tasks in interface configuration mode:

Task	Command
Select ARPA Ethernet encapsulation.	**encapsulation arpa**
Select SAP Ethernet encapsulation.	**encapsulation sap**
Select SNAP Ethernet encapsulation.	**encapsulation snap**

For an example of selecting Ethernet encapsulation, see the section “Example of Enabling Ethernet Encapsulation” at the end of this chapter. See also the chapters describing specific protocols or applications.

Configure the Ethernet Network Interface Module on the Cisco 4000

You can specify the type of Ethernet Network Interface Module configuration on the Cisco 4000. To do so, perform one of the following tasks in interface configuration mode:

Task	Command
Select a 15-pin Ethernet connector.	**media-type aui**
Select an RJ45 Ethernet connector.	**media-type 10baset**

Extend the 10BaseT Capability

On a Cisco 4000 or Cisco 4500, you can extend the twisted-pair 10BaseT capability beyond the standard 100 meters by reducing the squelch (signal cutoff time). This feature applies only to the LANCE controller 10BaseT interfaces. LANCE is the AMD controller chip for the Cisco 4000 and Cisco 4500 Ethernet interface.

To reduce squelch, perform the first task that follows in interface configuration mode. You can later restore the squelch by performing the second task.

Task	Command
Reduce the squelch.	**squelch reduced**
Return squelch to normal.	**squelch normal**

Configure a Fiber Distributed Data Interface (FDDI)

The Fiber Distributed Data Interface (FDDI) is an ANSI-defined standard for timed 100-Mbps token passing over fiber-optic cable. An FDDI network consists of two counter token-passing fiber-optic rings. On most networks, the primary ring is used for data communication and the secondary ring is used as a hot standby. The FDDI standard sets a total fiber length of 200 kilometers. (The maximum circumference of the FDDI network is only half the specified kilometers because of the *wrapping* or looping back of the signal that occurs during fault isolation.)

The FDDI standard allows a maximum of 500 stations with a maximum distance between active stations of two kilometers when interconnecting them with multimode fiber or ten kilometers when interconnected via single mode fiber, both of which are supported by our FDDI interface controllers. The FDDI frame can contain a minimum of 17 bytes and a maximum of 4500 bytes. Our implementation of FDDI supports Station Management (SMT) Version 7.3 of the X3T9.5 FDDI specification, offering a single MAC dual-attach interface that supports the fault-recovery methods of the dual attachment stations (DASs). The mid-range platforms also support single attachment stations (SASs).

Support for FDDI is supplied on one of our FDDI interface cards, as follows:

- The CSC-FCI interface card, which operates with the standard modular router controller complex
- The CSC-C2/FCIT interface card, which operates with the ciscoBus2 controller complex
- On the Cisco 4000 series, the high-speed multimode-to-multimode, single mode-to-single mode, multimode-to-single mode, or single mode-to-multimode FDDI DAS Network Interface Module (NIM), and also the multimode FDDI SAS NIM.
- On the Cisco 7000 series, the high-speed multimode-to-multimode, single mode-to-single mode, multimode-to-single mode, or single mode-to-multimode FDDI Interface Processor (FIP)

We also provide support for some of the FDDI MIB variables as described in RFC 1285, "FDDI Management Information Base," published in January 1992 by Jeffrey D. Case of the University of Tennessee and SNMP Research, Inc. One such variable that we support is *snmpFddiSMTCFState*.

FDDI interface configuration is not required for dual homing. The FDDI interface recognizes that it is attached to two M ports on the concentrators and automatically supports dual homing.

Using Connection Management (CMT) Information

Connection Management (CMT) is an FDDI process that handles the transition of the ring through its various states (off, on, active, connect, and so on) as defined by the X3T9.5 specification. The FIP provides CMT functions in microcode.

A partial sample output of the **show interfaces fddi** command follows, along with an explanation of how to interpret the CMT information in the output.

```
Phy-A state is active, neighbor is B, cmt signal bits 08/20C, status ALS
Phy-B state is active, neighbor is A, cmt signal bits 20C/08, status ILS
CFM is thru A, token rotation 5000 usec, ring operational 0:01:42
Upstream neighbor 0800.2008.C52E, downstream neighbor 0800.2008.C52E
```

The **show interfaces fddi** example shows that Physical A (Phy-A) completed CMT with its neighbor. The state is active and the display indicates a Physical B-type neighbor.

The sample output indicates cmt signal bits 08/20C for Phy-A. The transmit signal bits are 08. Looking at the PCM state machine, 08 indicates that the port type is A, the port compatibility is set, and the LCT duration requested is short. The receive signal bits are 20C, which indicate the neighbor type is B, port compatibility is set, there is a MAC on the port output, and so on.

The neighbor is determined from the received signal bits, as follows:

Bit Positions	9	8	7	6	5	4	3	2	1	0
Value Received	1	0	0	0	0	0	1	1	0	0

Interpreting the bits in the diagram above, the received value equals 0x20C. Bit positions 1 and 2 (0 1) indicate a Physical B-type connection.

The transition states displayed indicate that the CMT process is running and actively trying to establish a connection to the remote physical connection. The CMT process requires state transition with different signals being transmitted and received before moving on to the state ahead as indicated in the PCM state machine. The ten bits of CMT information are transmitted and received in the Signal State. The NEXT state is used to separate the signaling performed in the Signal State. Therefore, in the preceding sample output, the NEXT state was entered 11 times.

Note The display line showing transition states is not generated if the FDDI interface has been shut down, or if the **cmt disconnect** command has been issued, or if the **fddi if-cmt** command has been issued. (The **fddi if-cmt** command applies to the AGS+ and Cisco 7000 only.)

The CFM state is thru A in the sample output, which means this interface's Phy-A has successfully completed CMT with the Phy-B of the neighbor and Phy-B of this interface has successfully completed CMT with the Phy-A of the neighbor.

The display (or nondisplay) of the upstream and downstream neighbor does not affect the ability to route data. Since the upstream neighbor is also its downstream neighbor in the sample, there are only two stations in the ring: the network server and the router at address 0800.2008.C52E.

FDDI Task List

Perform the tasks in the following sections to configure an FDDI interface. The first task is required; the remaining tasks are optional.

- Specify an FDDI
- Enable FDDI Bridging Encapsulation
- Set the Token Rotation Time
- Set the Transmission Valid Timer
- Control the Transmission Timer
- Modify the C-Min Timer
- Modify the TB-Min Timer
- Modify the FDDI Timeout Timer
- Control SMT Frame Processing
- Enable Duplicate Address Checking
- Set the Bit Control
- Control the CMT Microcode
- Start and Stop FDDI

- Control the FDDI SMT Message Queue Size
- Preallocate Buffers for Bursty FDDI Traffic

Specify an FDDI

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

Task	Command
Begin interface configuration	**interface fddi** *number*
Begin interface configuration for the Cisco 7000 series.	**interface fddi** *slot/port*

Enable FDDI Bridging Encapsulation

Our FDDI by default uses the SNAP encapsulation format defined in RFC 1042. It is not necessary to define an encapsulation method for this interface when using the CSC-FCI interface card or FIP.

The CSC-C2/FCIT interface card and FIP fully support transparent and translational bridging for the following configurations:

- FDDI to FDDI
- FDDI to Ethernet
- FDDI to Token Ring

Enabling FDDI bridging encapsulation places the CSC-C2/FCIT interface or FIP into encapsulation mode when doing bridging. In transparent mode, the FCIT interface or FIP interoperates with earlier versions of the CSC-FCI encapsulating interfaces when performing bridging functions on the same ring. When using the CSC-C2/FCIT interface card or FIP, you can specify the encapsulation method by performing the following task in interface configuration mode:

Task	Command
Specify the encapsulation method for the CSC-C2/FCIT interface card or FIP.	**fddi encapsulate**

When you are translationally bridging, you have to route routable protocols and translationally bridge the rest (such as LAT).

The CSC-FCI interfaces are always in encapsulating bridge mode, so disabling applies only to CSC-C2/FCIT interfaces.

Note Bridging between dissimilar media presents several problems that can prevent communications. These problems include bit-order translation (or use of MAC addresses as data), maximum transfer unit (MTU) differences, frame status differences, and multicast address usage. Some or all of these problems might be present in a multimedia-bridged LAN and might prevent communication. These problems are most prevalent when bridging between Token Rings and Ethernets or between Token Rings and FDDI nets. This is because of the different way Token Ring is implemented by the end nodes.

We are currently aware of problems with the following protocols when bridged between Token Ring and other media: AppleTalk, DECnet, IP, Novell IPX, Phase IV, VINES, and XNS. Further, the following protocols might have problems when bridged between FDDI and other media: Novell IPX and XNS. We recommend that these protocols be routed whenever possible.

Set the Token Rotation Time

You can set the FDDI token rotation time to control ring scheduling during normal operation and to detect and recover from serious ring error situations. To do so, perform the following task in interface configuration mode:

Task	Command
Set the FDDI token rotation time.	**fddi token-rotation-time** *microseconds*

The FDDI standard restricts the allowed time to be greater than 4000 microseconds and less than 165,000 microseconds. As defined in the X3T9.5 specification, the value remaining in the token rotation timer (TRT) is loaded into the token holding timer (THT). Combining the values of these two timers provides the means to determine the amount of bandwidth available for subsequent transmissions.

Set the Transmission Valid Timer

You can set the transmission timer to recover from a transient ring error by performing the following task in interface configuration mode:

Task	Command
Set the FDDI valid transmission timer.	**fddi valid-transmission-time** *microseconds*

Control the Transmission Timer

You can set the FDDI control transmission timer to control the FDDI TL-Min time, which is the minimum time to transmit a Physical Sublayer or PHY line state before advancing to the next Physical Connection Management or PCM state as defined by the X3T9.5 specification. To do so, perform the following task in interface configuration mode:

Task	Command
Set the FDDI control transmission timer.	**fddi tl-min-time** *microseconds*

Modify the C-Min Timer

You can modify the C-Min timer on the PCM from its default value of 1600 microseconds by performing the following task in interface configuration mode:

Task	Command
Set the C-Min timer on the PCM.	**fddi c-min** *microseconds*

Modify the TB-Min Timer

You can change the TB-Min timer in the PCM from its default value of 100 milliseconds. To do so, perform the following task in interface configuration mode:

Task	Command
Set TB-Min timer in the PCM.	**fddi tb-min** *milliseconds*

Modify the FDDI Timeout Timer

You can change the FDDI timeout timer in the PCM from its default value of 100 milliseconds. To do so, perform the following task in interface configuration mode:

Task	Command
Set the timeout timer in the PCM.	**fddi t-out** *milliseconds*

Control SMT Frame Processing

You can disable and reenable SMT frame processing for diagnostic purposes. To do so, perform one of the following tasks in interface configuration mode:

Task	Command
Disable SMT frame processing.	no fddi smt-frames
Enable SMT frame processing.	**fddi smt-frames**

Enable Duplicate Address Checking

You can enable the duplicate address detection capability on the FDDI. If the FDDI finds a duplicate address, it displays an error message and shuts down the interface. To enable duplicate address checking, perform the following task in interface configuration mode:

Task	Command
Enable duplicate address checking capability.	**fddi duplicate-address-check**

Set the Bit Control

You can set the FDDI bit control to control the information transmitted during the Connection Management (CMT) signaling phase. To do so, perform the following task in interface configuration mode:

Task	Command
Set the FDDI bit control.	**fddi cmt-signal-bits** *signal-bits* [**phy-a** | **phy-b**]

Control the CMT Microcode

You can control whether the CMT onboard functions are on or off. The CSC-FCI and CSC-C2/FCIT interface cards and FIP provide CMT functions in microcode. These functions are separate from those provided on the processor card and are accessed through EXEC commands.

The default is for the FCIT and FIP CMT functions to be on. A typical reason to disable is when you work with new FDDI equipment and have problems bringing up the ring. If you disable the CMT microcode, the following actions occur:

- The FCIT or FIP CMT microcode is disabled.
- The main system code performs the CMT function while debugging output is generated.

To disable the CMT microcode, perform the following task in interface configuration mode:

Task	Command
Disable the FCIT CMT functions.	no fddi if-cmt

Start and Stop FDDI

In normal operation, the FDDI interface is operational once the interface is connected and configured. You can start and stop the processes that perform the CMT function and allow the ring on one fiber to be stopped. To do so, perform either of the following tasks in EXEC mode:

Task	Command
Start CMT processes on FDDI ring.	**cmt connect** [*interface-name* [**phy-a** | **phy-b**]]
Stop CMT processes on FDDI ring.	**cmt disconnect** [*interface-name* [**phy-a** | **phy-b**]]

Do not do either of the preceding tasks during normal operation of FDDI; they are performed during interoperability tests.

Control the FDDI SMT Message Queue Size

You can set the maximum number of unprocessed FDDI Station Management (SMT) frames that will be held for processing. Setting this number is useful if the router you are configuring gets bursts of messages arriving faster than the router can process them. To set the number of frames, perform the following task in global configuration mode:

Task	Command
Set SMT message queue size.	**smt-queue-threshold** *number*

Preallocate Buffers for Bursty FDDI Traffic

The FCI card preallocates three buffers to handle bursty FDDI traffic (for example, NFS bursty traffic). You can change the number of preallocated buffers by performing the following task in interface configuration mode:

Task	Command
Preallocate buffers to handle bursty FDDI traffic.	**fddi burst-count**

Configure a High-Speed Serial Interface (HSSI)

The High-Speed Serial Interface (HSSI) consists of the following components:

- The CSC-HSCI controller card, which is ciscoBus-resident.
- The CSC-HSA, which is a back-panel applique.

  The controller card provides a single, full-duplex, synchronous serial interface capable of transmitting and receiving data at up to 52 megabits per second (Mbps). The HSSI is an approved standard (ANSI/EIA RS-613) providing connectivity to T3 (DS-3), E3, SMDS (at a DS-3 route), and other high-speed wide-area services through a DSU or line termination unit.

  - The high-speed, full-duplex, synchronous serial interface is supported only on our modular network server products.
  - The ciscoBus card can query the appliques to determine their types. However, it does so only at system startup, so the appliques must be attached when the system is started. Issue a **show controllers cbus** command to determine how the HSSI card has identified them. The command also will show the capabilities of the card and report controller-related failures.
- On the Cisco 7000 series, the HSSI Interface Processor (HIP), which provides a single HSSI network interface for the Cisco 7000. The network interface resides on a modular interface processor that provides a direct connection between the high-speed Cisco Extended Bus (CxBus) and an external network.

HSSI Task List

Perform the tasks in the following sections to configure an HSSI interface. The first task is required; the remaining tasks are optional.

- Specify an HSSI
- Specify HSSI Encapsulation
- Invoke ATM on an HSSI Line
- Convert HSSI to Clock Master

Specify an HSSI

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

Task	Command
Begin interface configuration.	**interface hssi** *number*
Begin interface configuration for the Cisco 7000 series.	**interface hssi** *slot/port*

Specify HSSI Encapsulation

The HSSI supports the serial encapsulation methods, except for X.25-based encapsulations. The default method is HDLC. You can define the encapsulation method by performing the following task in interface configuration mode:

Task	Command
Configure HSSI encapsulation.	**encapsulation** {**atm-dxi** | **hdlc** | **frame-relay** | **ppp** | **sdlc-primary** | **sdlc-secondary** | **smds** | **stun**}

For information about PPP, see the section “Configure PPP” in the section “Configure a Synchronous Serial Interface” later in this chapter.

Invoke ATM on an HSSI Line

If you have an ATM DSU, you can invoke ATM over a HSSI line.You do so by mapping an ATM virtual path identifier (VPI) and virtual channel identifier (VCI) to a DXI frame address. ATM-DXI encapsulation defines a data exchange interface that allows a DTE (such as a router) and a DCE (such as an ATM DSU) to cooperate to provide a User - Network Interface (UNI) for ATM networks.

To invoke ATM over a serial line, perform the following tasks in interface configuration mode:

Task		Command
Step 1	Specify the encapsulation method.	**encapsulation atm-dxi**
Step 2	Map a given VPI and VCI to a DXI frame address.	**atm-dxi map** *protocol address vpi vci* [**broadcast**]

You can also configure the **atm-dxi map** command on a serial interface.

To configure an ATM interface using an AIP card, see the chapter “Configuring ATM.”

Convert HSSI to Clock Master

You can convert the HSSI interface into a 45-MHz clock master by performing the following task in interface configuration mode:

Task	Command
Convert the HSSI interface into a 45-MHz clock master.	**hssi internal-clock**

Configure a Hub Interface

The Cisco 2500 series includes routers that have hub functionality for an Ethernet interface. The hub is a multiport repeater. The advantage of an Ethernet interface over a hub is that the hub provides a star-wiring physical network configuration while the Ethernet interface provides 10BaseT physical network configuration. The router models with hub ports and their configurations are as follows:

- Cisco 2505—1 Ethernet (8 ports) and 2 serial
- Cisco 2507—1 Ethernet (16 ports) and 2 serial
- Cisco 2516—1 Ethernet (14 ports), 2 serial, and 1 ISDN BRI

We provide SNMP management of the Ethernet hub as specified in RFC 1516.

To configure hub functionality on an Ethernet interface, perform the tasks in the following sections. The first task is required; the remaining are optional.

- Enable a Hub Port
- Disable or Enable Automatic Receiver Polarity Reversal
- Disable or Enable the Link Test Function
- Enable Source Address Control

See the end of this chapter for “Examples of Hub Configuration.”

Enable a Hub Port

To enable a hub port, perform the following tasks in global configuration mode:

Task		Command
Step 1	Specify the hub number and the hub port (or range of hub ports) and enter hub configuration mode.	**hub ethernet** *number port* [*end-port*]
Step 2	Enable the hub ports.	**no shutdown**

Disable or Enable Automatic Receiver Polarity Reversal

On Ethernet hub ports only, the hub ports can invert, or correct, the polarity of the received data if the port detects that the received data packet waveform polarity is reversed due to a wiring error. This receive circuitry polarity correction allows the hub to repeat subsequent packets with correct polarity. When enabled, this function is executed once after reset of a link fail state.

Automatic receiver polarity reversal is enabled by default. To disable this feature on a per-port basis, perform the following task in hub configuration mode:

Task	Command
Disable automatic receiver polarity reversal.	**no auto-polarity**

To reenable automatic receiver polarity reversal on a per-port basis, perform the following task in hub configuration mode:

Task	Command
reenable automatic receiver polarity reversal.	**auto-polarity**

Disable or Enable the Link Test Function

The link test function applies to Ethernet hub ports only. The Ethernet ports implement the link test function as specified in the 802.3 10BaseT standard. The hub ports will transmit link test pulses to any attached twisted pair device if the port has been inactive for more than 8 to 17 milliseconds.

If a hub port does not receive any data packets or link test pulses for more than 65 to 132 milliseconds and the link test function is enabled for that port, that port will enter link fail state and be disabled from transmit and receive functions. The hub port will be reenabled when it receives four consecutive link test pulses or a data packet.

The link test function is enabled by default. To allow the hub to interoperate with 10BaseT twisted-pair networks that do not implement the link test function, the hub's link test receive function can be disabled on a per-port basis. To do so, perform the following task in hub configuration mode:

Task	Command
Disable the link test function.	**no link-test**

To reenable the link test function on a hub port connected to an Ethernet interface, perform the following task in hub configuration mode:

Task	Command
Enable the link test function.	**link-test**

Enable Source Address Control

On an Ethernet hub port only, you can configure a security measure such that the port accepts packets only from a specific MAC address. For example, suppose your workstation is connected to port 3 on a hub, and source address control is enabled on port 3. Your workstation has access to the network because the hub accepts from port 3 any packet bearing your workstation's MAC address. Any packets arriving with a different MAC address cause the port to be disabled. The port is reenabled after 1 minute and the MAC address of incoming packets is checked again.

To enable source address control on a per-port basis, perform the following task in hub configuration mode:

Task	Command
Enable source address control.	**source-address** [*mac-address*]

If you omit the optional MAC address, the hub remembers the first MAC address it receives on the selected port, and allows only packets from the learned MAC address.

See the examples of establishing source address control at the end of this chapter in "Examples of Hub Configuration."

Configure an ISDN Basic BRI, MBRI, or PRI Interface

ISDN Primary Rate Interface (PRI) is supported only on the Cisco 7000 on the channelized T1 card and E1 card (the MultiChannel Interface Processor [MIP] card). ISDN PRI over T1 offers 23 B-channels and 1 D-channel. The E1 support provides 30 B-channel and 1 D-channel. To configure an ISDN PRI interface, see the chapter "Configuring ISDN."

The Integrated Services Digital Network (ISDN) Basic Rate Interface (BRI) is supported on the Cisco 1003 model, Cisco 2500 series, Cisco 3000 series, and Cisco 4000 series routers. ISDN Multiport BRI (MBRI) is supported on the Cisco 4000 and Cisco 4500 only, which have a multichannel NIM. The multichannel card supports one or two BRI port adapters, providing either 4 or 8 ports, respectively.

For information on how to configure ISDN, see the chapter entitled "Configuring ISDN." For command information, refer to the chapter entitled "DDR Commands" in the *Router Products Command Reference* publication.

Note Any router configured for ISDN support must be connected to the same switch type on all its ISDN interfaces.

Configure a LAN Extender Interface

The Cisco 1001 and Cisco 1002 LAN Extenders are two-port chassis that connects a remote Ethernet LAN to a core router at a central site (see Figure 6-2). The LAN Extender is intended for small networks at remote sites.

The remote site can have one Ethernet network. The core router can be a Cisco 2500 series, Cisco 4000 series, Cisco 4500 series, Cisco 4700 series, Cisco 7000 series, or AGS+ router running Cisco IOS Release 10.2(2) or later, which support the LAN Extender host software. The connection between the LAN Extender and the core router is made via a short leased serial line, typically a 56-kbps or 64-kbps line. However, the connection can also be via T1 or E1 lines.

Figure 6-1 Cisco 1000 Series LAN Extender Connection to a Core Router

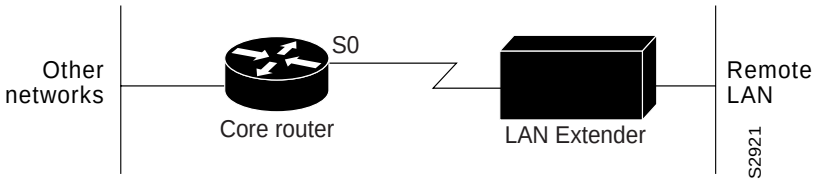
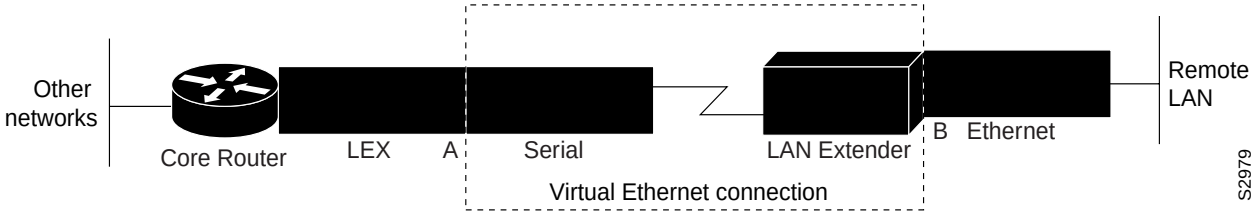


Figure 6-2 is an expanded view of Figure 6-1 that shows all the components of the LAN Extender connection to a core router. On the left is the core router, which is connected to the LAN Extender as well as to other networks. In the core router, you configure a LAN Extender interface, which is a logical interface that connects the core router to the LAN Extender chassis. In the core router, you also configure a serial interface, which is the physical interface that connects the core router to the LAN Extender. You then bind, or associate, the LAN Extender interface to the physical serial interface.

Figure 6-2 shows the actual physical connection between the core router and the LAN Extender. The serial interface on the core router is connected by a leased serial line to a serial port on the LAN Extender. This creates a virtual Ethernet connection, which is analogous to having inserted an Ethernet interface processor into the core router.

Figure 6-2 Expanded View of Cisco 1000 Series LAN Extender Connection



Although there is a physical connection between the core router and the LAN Extender, what you actually manage is a remote Ethernet LAN. Figure 6-3 shows the connection you are managing, which is a LAN Extender interface connected to an Ethernet network. The virtual Ethernet connection (the serial interface and LAN Extender) has been removed from the figure, and points A and B, which in Figure 6-2 were separated by the virtual Ethernet connection, are now adjacent. All LAN Extender interface configuration tasks described in this chapter apply to the interface configuration shown in Figure 6-3.

Figure 6-3 LAN Extender Interface Connected to an Ethernet Network

To install a LAN Extender at a remote site, refer to the *Cisco 1000 Series Hardware Installation* publication.

After the LAN Extender has been installed at the remote site, you need to obtain its MAC address. Each LAN Extender is preconfigured with a permanent (burned-in) MAC address. The address is assigned at the factory; you cannot change it. The MAC address is printed on the LAN Extender's

packing box. (If necessary, you can also display the MAC address with the **debug ppp negotiation** command.) The first three octets of the MAC address (the vendor code) are always the hexadecimal digits 00.00.0C.

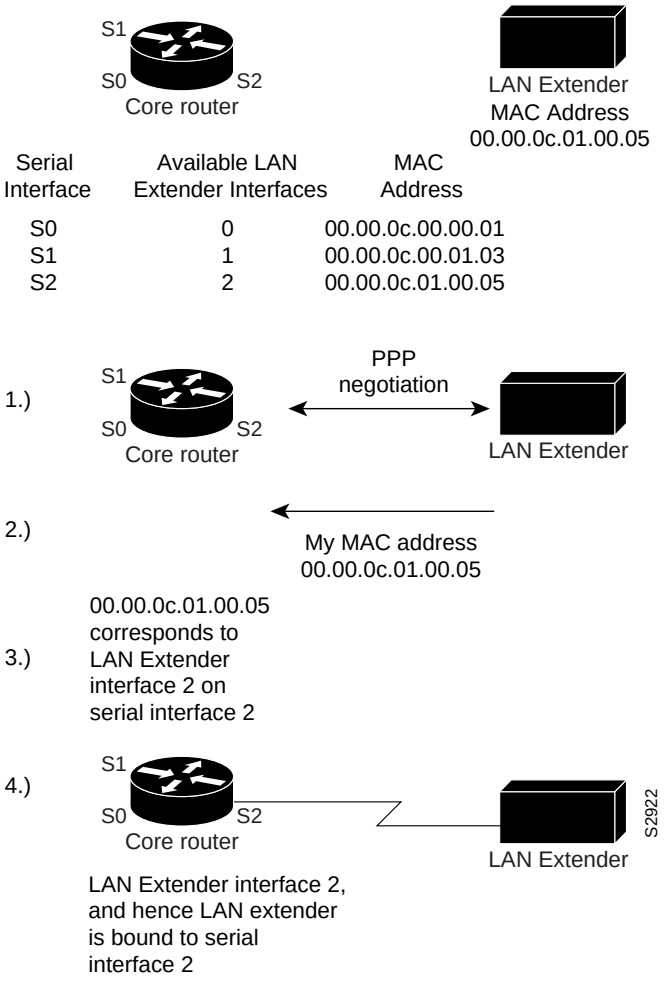
You can upgrade software for the LAN Extender on the host router with a TFTP server that is local to the host router.

The LAN Extender and core router communicate using the Point-to-Point Protocol (PPP). Before you can configure the LAN Extender from the core router, you must first enable PPP encapsulation on the serial interface to which the LAN Extender is connected.

You then configure the LAN Extender from the core router—either a Cisco 4000 series or Cisco 7000 series router—as if it were simply a network interface board. The LAN Extender cannot be managed or configured from the remote Ethernet LAN or via a Telnet session.

To configure the LAN Extender, you configure a logical LAN Extender interface on the core router and assign the MAC address from your LAN Extender to that interface. Subsequently, during the PPP negotiation on the serial line, the LAN Extender sends its preconfigured MAC address to the core router. The core router then searches for an available (preconfigured) LAN Extender interface, seeking one to which you have already assigned that MAC address. If the core router finds a match, it binds, or associates, that LAN Extender interface to the serial line on which that MAC address was negotiated. At this point, the LAN Extender interface is created and is operational. If the MAC address does not match one that is configured, the connection request is rejected. Figure 6-4 illustrates this binding process.

Figure 6-4 Binding a Serial Line to a LAN Extender Interface



LAN Extender Interface Configuration Task List

To configure a LAN Extender interface, perform the tasks described in the following sections. The first task is required; the remainder are optional.

- Configure and Create a LAN Extender Interface
- Define Packet Filters
- Control Priority Queuing
- Control the Sending of Commands to the LAN Extender
- Shut Down and Restart the LAN Extender's Ethernet Interface
- Restart the LAN Extender
- Download a Software Image to the LAN Extender
- Troubleshoot the LAN Extender

To monitor the LAN Extender interface, see the section "Monitor and Maintain the Interface" later in this chapter. See the end of this chapter for configuration examples.

Configure and Create a LAN Extender Interface

To configure and create a LAN Extender interface, you configure the LAN Extender interface itself and the serial interface to which the LAN Extender is physically connected. The order in which you configure these two interface interfaces does not matter. However, you must first configure both interfaces in order for the LAN Extender interface to bind (associate) to the serial interface.

To create and configure a LAN Extender interface, perform the following tasks:

Task		Command
Step 1	Configure a LAN Extender interface in global configuration mode and enter interface configuration mode.	**interface lex** *number*
Step 2	Assign the burned-in MAC address from your LAN Extender to the LAN Extender interface.	**lex burned-in-address** *ieee-address*
Step 3	Assign a protocol address to the LAN Extender interface.	**ip address** *ip-address mask*
Step 4	Return to global configuration mode.	**exit**
Step 5	Configure a serial interface in global configuration mode and enter interface configuration mode.	**interface serial** *number*
Step 6	Enable PPP encapsulation on the serial interface in interface configuration mode.	**encapsulation ppp**
Step 7	(Optional) Enable Stacker compression.	**ppp compress stac**
Step 8	Exit interface configuration mode.	**Ctrl-Z**
Step 9	Save the configuration to memory.	write memory

Note that there is no correlation between the number of the serial interface and the number of the LAN Extender interface. These interfaces can have the same or different numbers.

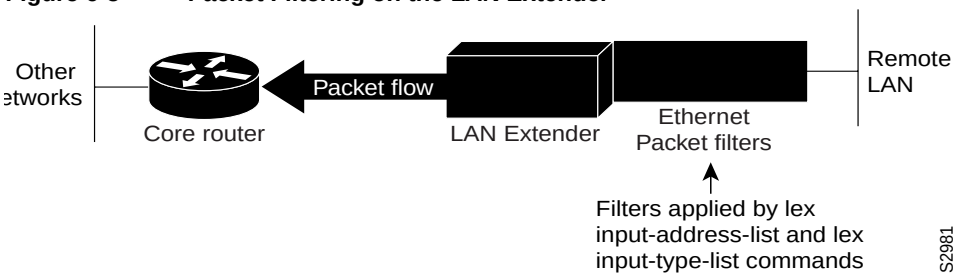
Note Do not configure the MTU to a value other than the default value when you are configuring a LAN Extender interface.

Define Packet Filters

You can configure specific administrative filters that filter frames based on their source MAC address. The LAN Extender forwards packets between a remote LAN and a core router. It examines frames and transmits them through the internetwork according to the destination address, and it does not forward a frame back to its originating network segment.

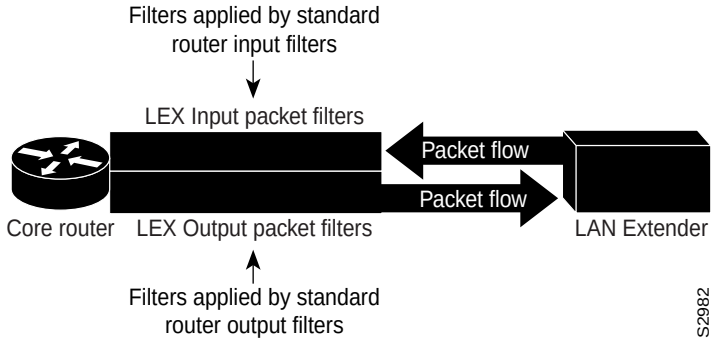
You define filters on the LAN Extender interface in order to control which packets from the remote Ethernet LAN are permitted to pass to the core router. (See Figure 6-5.) These filters are applied only on traffic passing from the remote LAN to the core router. Filtering on the LAN Extender interface is actually performed in the LAN Extender, not on the core router. This means that the filtering is done using the LAN Extender CPU, thus off-loading the function from the core router. This process also saves bandwidth on the WAN, because only the desired packets are forwarded from the LAN Extender to the core router. Whenever possible, you should perform packet filtering on the LAN Extender.

Figure 6-5 Packet Filtering on the LAN Extender



You can also define filters on the core router to control which packets from the LAN Extender interface are permitted to pass to other interfaces on the core router. (See Figure 6-6.) You do this using the standard filters available on the router. This means that all packets are sent across the WAN before being filtered and that the filtering is done using the core router's CPU.

Figure 6-6 Packet Filtering on the Core Router



The major reason to create access lists on a LAN Extender interface is to prevent traffic that is local to the remote Ethernet LAN from traversing the WAN and reaching the core router. You can filter packets by MAC address, including vendor code, and by Ethernet type code. To define filters on the LAN Extender interface, perform the tasks described in one or both of the following sections:

- Filter by MAC Address and Vendor Code
- Filter by Protocol Type

Note When setting up administrative filtering, remember that there is virtually no performance penalty when filtering by vendor code, but there can be a performance penalty when filtering by protocol type.

When defining access lists, keep the following points in mind:

- You can assign only one vendor code access list and only one protocol type access list to an interface.
- The conditions in the access list are applied to all outgoing packets from the LAN Extender.
- The entries in an access list are scanned in the order you enter them. The first entry that matches the outgoing packet is used.

- An implicit “deny everything” entry is automatically defined at the end of an access list unless you include an explicit “permit everything” entry at the end of the list. This means that unless you have an entry at the end of an access list that explicitly permits all packets that do no match any of the other conditions in the access list, these packets will not be forwarded out the interface.
- All new entries to an existing list are placed at the end of the list. You cannot add an entry to the middle of a list.
- If you do not define any access lists on an interface, it is as if you had defined an access lists with only a “permit all” entry. All traffic passes across the interface.

Filter by MAC Address and Vendor Code

You can create access lists to administratively filter MAC addresses. These access lists can filter groups of MAC addresses, including those with particular vendor codes. There is no noticeable performance loss in using these access lists, and the lists can be of indefinite length. You can filter groups of MAC addresses with particular vendor codes by performing the tasks that follow:

Step 1 Create a vendor code access list.

Step 2 Apply an access list to an interface.

To create a vendor code access list, perform the following task in global configuration mode:

Task	Command
Create an access list to filter frames by canonical (Ethernet-ordered) MAC address.	**access-list** *access-list-number* {**permit** | **deny**} *address mask*

Note Token Ring and FDDI networks swap their MAC address bit ordering, but Ethernet networks do not. Therefore, an access list that works for one medium might not work for others.

Once you have defined an access list to filter by a particular vendor code, you can assign this list to a particular LAN Extender interface so that the interface will then filter based on the MAC source addresses of packets received on that LAN Extender interface. To apply the access list to an interface, perform the following task in interface configuration mode:

Task	Command
Assign an access list to an interface for filtering by MAC source addresses.	**lex input-address-list** *access-list-number*

For an example of creating an access list and applying it to a LAN Extender interface, see the section “Examples of LAN Extender Interface Access List” in the section “Interface Configuration Examples” at the end of this chapter.

Filter by Protocol Type

You can filter by creating a type-code access list and applying it to a LAN Extender interface.

The LAN Extender interface can filter only on bytes 13 and 14 of the Ethernet frame. In Ethernet packets, these two bytes are the type field. For a list of Ethernet type codes, refer to the “Ethernet Type Codes” appendix in the *Router Products Command Reference* publication. In 802.3 packets, these two bytes are the length field.

To filter by protocol type, perform the following tasks:

Step 1 Create a protocol-type access list.

Step 2 Apply the access list to an interface.

Note Type-code access lists can have an impact on system performance; therefore, keep the lists as short as possible and use wildcard bit masks whenever possible.

To create a protocol-type access list, perform the following task in global configuration mode:

Task	Command
Create an access list to filter frames by protocol type.	**access-list** *access-list-number* {**permit** | **deny**} *type-code wild-mask*

To apply an access list to an interface, perform the following task in interface configuration mode:

Task	Command
Add a filter for Ethernet- and SNAP-encapsulated packets on input.	**lex input-type-list** *access-list-number*

For an example of creating an access list and applying it to a LAN Extender interface, see the section "Examples of LAN Extender Interface Access List" in the section "Interface Configuration Examples" at the end of this chapter.

Control Priority Queuing

Priority output queuing is an optimization mechanism that allows you to set priorities on the type of traffic passing through the network. Packets are classified according to various criteria, including protocol and subprotocol type. Packets are then queued on one of four output queues. For more information about priority queuing, refer to the "Managing the System" chapter.

To control priority queuing on a LAN Extender interface, perform the following tasks:

- Set the priority by protocol type.
- Assign a priority group to an interface.

To establish queuing priorities based on the protocol type, perform the following task in global configuration mode:

Task	Command
Establish queuing priorities based on the protocol type.	**priority-list** *list* **protocol** *protocol* {**high** | **medium** | **normal** | **low**} or **priority-list** *list* **protocol bridge** {**high** | **medium** | **normal** | **low**} **list** *list-number*

You then assign a priority list to an interface. You can assign only one list per interface. To assign a priority list to a LAN Extender interface, perform the following task in interface configuration mode:

Task	Command
Assign a priority list to a LAN Extender interface, thus activating priority output queuing on the LAN Extender.	**lex priority-group** *group*

Control the Sending of Commands to the LAN Extender

Each time the core router sends a command to the LAN Extender, the LAN Extender responds with an acknowledgment. The core router waits for the acknowledgment for a predetermined amount of time. If it does not receive an acknowledgment in this time period, the core router resends the command.

By default, the core router waits 2 seconds for an acknowledgment from the LAN Extender. You might want to change this interval if your connection to the LAN Extender requires a different amount time.To determine whether commands to the LAN Extender are timing out, use the **debug lex rcmd** privileged EXEC command. To change this interval, perform the following task in interface configuration mode:

Task	Command
Set the amount of time that the core router waits to receive an acknowledgment from the LAN Extender.	**lex timeout** *milliseconds*

By default, the core router sends each command ten times before giving up. The core router displays an error message when it gives up sending commands to the LAN Extender. To change this default, perform the following task in interface configuration mode:

Task	Command
Set the number of times the core router sends a command to the LAN Extender before giving up.	**lex retry-count** *number*

Shut Down and Restart the LAN Extender's Ethernet Interface

From the core router, you can shut down the LAN Extender's Ethernet interface. This stops traffic on the remote Ethernet LAN from reaching the core router, but leaves the LAN Extender interface that you created intact.

Note that logically it makes no sense to shut down the serial interface on the LAN Extender. There are no commands that might allow you to do this.

To shut down the LAN Extender's Ethernet interface, perform the following task in interface configuration mode:

Task	Command
Shut down the LAN Extender's Ethernet interface.	**shutdown**

To restart the LAN Extender's Ethernet interface, perform the following task in interface configuration mode:

Task	Command
Restart the LAN Extender's Ethernet interface.	**no shutdown**

Restart the LAN Extender

To reboot the LAN Extender and reload the software, perform the following task in privileged EXEC mode:

Task	Command
Halt operation of the LAN Extender and have it perform a cold restart.	**clear controller lex** *number* [**prom**]
Halt operation of the LAN Extender on a Cisco 7000.	**clear controller lex** *slot*/*port* [**prom**]

Download a Software Image to the LAN Extender

When the LAN Extender is powered on, it runs the software image that is shipped with the unit. You can download a new software image from Flash memory on the core router or from a TFTP server to the LAN Extender.

To download a software image to the LAN Extender, perform one of the following tasks in privileged EXEC mode:

Task	Command
Download a software image from a TFTP server.	**copy tftp lex** *number*
Download a software image from Flash memory.	**copy flash lex** *number*

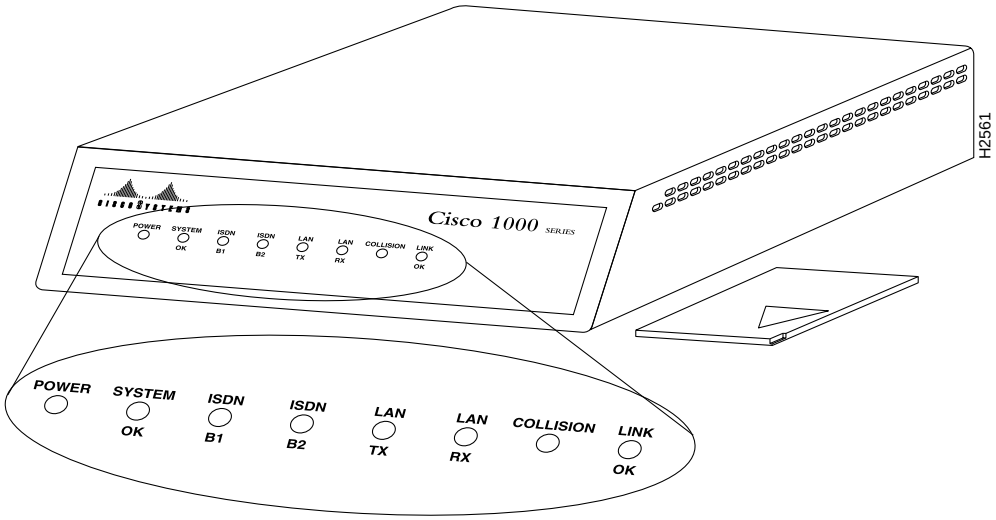
Troubleshoot the LAN Extender

The primary means of troubleshooting the LAN Extender is by using the light emitting diodes (LEDs) that are present on the chassis. This section will help you assist the remote user at the LAN Extender site who can observe the LEDs.

The key to problem solving is to try to isolate the problem to a specific subsystem. By comparing what the system is doing to what it should be doing, the task of isolating a problem is greatly simplified.

The Cisco 1000 series LAN Extender uses multiple LEDs to indicate its current operating condition. By observing the LEDs, any fault conditions that the unit is encountering can be observed. The system LEDs are located on the front panel of your LAN Extender (see Figure 6-7).

Figure 6-7 LAN Extender LEDs



When there is a problem with the LAN Extender, a user at the remote site should contact you and report the condition of the LEDs located on the front panel of the LAN Extender. You can then use this information to diagnose or verify the operation of the system. Table 6-1 explains the LEDs.

Table 6-1 LED Trouble Indicators

LED	Condition	Meaning
POWER	On Steady	The POWER LED indicates that 12 Volts DC is being supplied to the LAN Extender.
	Off	If the POWER LED is off, power is not reaching the unit. Verify that the power supply is plugged into the wall receptacle, and that the cable from the power supply to the unit is connected.
SYSTEM OK	On Steady	The SYSTEM OK LED is lit when the unit passes the power on diagnostics. This indicates proper operation.
	Blinking	The system will blink while running its startup diagnostics and then will go to a steady on position. Blinking after the start-up diagnostics indicates that a system error has been encountered. Contact your system administrator who will have you disconnect and then reconnect the power to recycle your LAN Extender. If the blinking continues, check your WAN connection and the RX and TX LEDs.
	Off	An error condition has occurred. Contact your system administrator who will ask you to disconnect the power cord and then reconnect it to re-establish power to your LAN Extender.
SERIAL TX and SERIAL RX	Flicker	The serial line is transmitting and receiving packets normally.

LED	Condition	Meaning
	Blinking	A line fault has been detected. The LEDs will go on for several seconds and then they will blink a certain number of times to indicate a particular error. The LEDs will blink at a rate of one to two blinks per second. The following are the errors that can be encountered: 1 blink = The serial line is down. 2 blinks = No clock signal was received. 3 blinks = An excessive number of cyclic redundancy check (CRC) errors has been received. 4 blinks = The line is noisy. 5 blinks = A loopback condition has occurred. 6 blinks = The PPP link has failed. Contact your system administrator.
LAN TX and LAN RX	Flicker	The Ethernet LAN connection is transmitting and receiving data normally.
COLLISION		Data collisions are being detected.
LINK OK	Steady	This indicates the Ethernet link is up and functioning.

For more complete network troubleshooting information, refer to the *Troubleshooting Internetworking Systems* publication.

Configure a Loopback Interface

You can specify a software-only interface called a loopback interface that emulates an interface that is always up. 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 for establishing a Telnet session from the router's console to its auxiliary port when all other interfaces are down. In applications where other routers will 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 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 does double duty as the Null0 interface.

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.	**interface loopback** *number*

Task	Command
Begin interface configuration for the Cisco 7000 series.	**interface loopback** *slot*/*port*

See the section “Run Interface Loopback Diagnostics” later in this chapter.

Configure a Null Interface

The router 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 redirects**.

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.	**interface null 0**

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 Synchronous Serial Interface

Support for the synchronous serial interface is supplied on the following serial network interface cards or systems:

- The Multiport Communications Interface (CSC-MCI), a single card that provides up to two high-speed synchronous serial port connectors that support RS-232, V.35, RS-449, and X.21 connections
- The Serial Port Communications Interface (CSC-SCI), a single card that provides up to four high-speed serial ports that support RS-232, V.35, RS-449, and X.21 connections
- The high-speed synchronous serial interface on the Cisco 2500 series and Cisco 3000 series
- The four-port serial NIM on the Cisco 4000, available in two configurations:
  - A universal cable support (5/1 applique)
  - G.703 support (G.703 applique)
- On the Cisco 7000 series, the Fast Serial Interface Processor (FSIP) for four or eight channel-independent, synchronous serial ports that support full-duplex operation at DS-1 (1.544 Mbps) and E-1 (2.048 Mbps) speeds. Each port supports any of the available interface types (RS-232, RS-449, V.35, X.21, RS-530, and G.703), and each can be configured individually to operate with either internal or external timing signals.

The MCI and SCI cards can query the appliques to determine their types for use in reports displayed by the EXEC **show** commands. However, they do so only at system startup, so the appliques must be attached when the system is started. Use the **show interfaces** and **show controllers mci** EXEC commands to display the serial port numbers. These commands provide a report for each interface the router supports.

Synchronous Serial Task List

Perform the tasks in the following sections to configure a synchronous serial interface. The first task is required; the remaining tasks are optional.

- Specify a Synchronous Serial Interface
- Specify Synchronous Serial Encapsulation
- Configure PPP
- Configure Compression of PPP Data
- Configure Compression of LAPB Data
- Configure Compression of HDLC Data
- Invoke ATM over a Serial Line
- Configure the CRC
- Use the NRZI Line-Coding Format
- Enable the Internal Clock
- Invert the Transmit Clock Signal
- Set Transmit Delay
- Configure DTR Signal Pulsing
- Ignore DCD and Monitor DSR as Line Up/Down Indicator
- Configure the Clock Rate on DCE Appliques
- Specify the Serial Network Interface Module Timing
- Specify G.703 Interface Options

Specify a Synchronous Serial Interface

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

Task	Command
Begin interface configuration.	**interface serial** *number*
Begin interface configuration for the Cisco 7000 series.	**interface serial** *slot*/*port*
Begin interface configuration for a channelized T1 or E1 interface.	**interface serial** *slot*/*port*:*channel-group* (Cisco 7000 series) **interface serial** *number*:*channel-group* (Cisco 4000 series)

Specify Synchronous Serial Encapsulation

By default, synchronous serial lines use the High-Level Data Link Control (HDLC) serial encapsulation method, which provides the synchronous framing and error detection functions of HDLC without windowing or retransmission. The synchronous serial interfaces support the following serial encapsulation methods:

- Asynchronous Transfer Mode-Data Exchange Interface (ATM-DXI)
- High-Level Data Link Control (HDLC)
- Frame Relay
- Point-to-Point Protocol (PPP)
- Synchronous Data Link Control (SDLC)
- Switched Multimegabit Data Services (SMDS)
- Cisco Serial Tunnel (STUN)
- X.25-based encapsulations

You can define the encapsulation method by performing the following task in interface configuration mode:

Task	Command
Configure synchronous serial encapsulation.	**encapsulation** {**atm-dxi** | **hdlc** | **frame-relay** | **ppp** | **sdlc-primary** | **sdlc-secondary** | **smds** | **stun** | **x25**}

Encapsulation methods are set according to the type of protocol or application you configure on your router. ATM-DXI is described in this chapter in the section "Invoke ATM over a Serial Line." PPP is described in the next section, "Configure PPP." The remaining methods are defined in their respective chapters describing the protocols or applications. Serial encapsulation methods are also discussed in the *Router Products Command Reference* publication in the chapter entitled "Interface Commands" under the **encapsulation** command.

Configure PPP

The Point-to-Point Protocol (PPP), described in RFCs 1331 and 1332, is a method of encapsulating network layer protocol information over point-to-point links. You can configure PPP on the following types of interfaces:

- Asynchronous serial
- HSSI
- ISDN
- Synchronous serial

The current implementation of PPP supports option 3, authentication using CHAP or PAP, option 4, Link Quality Monitoring and option 5, Magic Number configuration options. The software always sends option 5 and will negotiate for options 3 and 4 if so configured. All other options are rejected.

We support the following upper-layer protocols: AppleTalk, Bridging, CLNS, DECnet, IP, IPX, VINES, and XNS.

The software provides PPP as an encapsulation method. It also provides the Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP) on serial interfaces running PPP encapsulation. The following sections describe the tasks to configure these features.

PPP Magic Number Support

Magic Number support is available on all serial interfaces. When using PPP, PPP will always attempt to negotiate for Magic Numbers, which are used to detect looped-back nets. The link might or might not be taken down upon looped-back detection, depending on the use of the **down-when-looped** command.

Enable PPP Encapsulation

You can enable PPP on serial lines to encapsulate IP and SLIP datagrams. To do so, perform the following task in interface configuration mode:

Task	Command
Enable PPP encapsulation.	encapsulation ppp

PPP echo requests are used as keepalives to minimize disruptions to the end users of your network. The **no keepalive command** can be used to disable echo requests.

Enable CHAP or PAP Authentication

Access control using Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) is available on all serial interfaces that use PPP encapsulation. The authentication feature reduces the risk of security violations on your router. You can configure either CHAP or PAP for the interface.

Note To use CHAP, you must be running PPP encapsulation.

When CHAP is enabled is enabled on an interface, the local router sends a CHAP packet and the remote device (a PC, workstation, or router) attempting to connect to the local router is requested, or "challenged," to respond.

The challenge consists of an ID, a random number, and either the host name of the local router or the name of the user on the remote device. This challenge is transmitted to the remote device.

The required response consists of two parts:

- An encrypted version of the ID, a secret password (or *secret*), and the random number
- Either the host name of the remote device or the name of the user on the remote device

When the local router receives the challenge response, it verifies the secret by looking up the name given in the response and performing the same encryption operation as indicated in the response. The secret passwords must be identical on the remote device and the local router.

Because the secret is never transmitted, other devices are prevented from stealing it and gaining illegal access to the system. Without the proper response, the remote device cannot connect to the local router.

CHAP transactions occur only at the time a link is established. The local router does not request a password during the rest of the call. (The local router can, however, respond to such requests from other devices during a call.)

To use CHAP, you must perform the following tasks:

Step 1 Enable CHAP on the interface.

Once you have enabled CHAP, the local router requires a password from remote devices. If the remote device does not support CHAP, no traffic will be passed to that device.

Step 2 Configure server host name or username authentication.

Configure the secret or password for each remote system with which authentication is required. As an alternative, you can also configure TACACS.

Perform the following tasks in interface configuration mode:

Task		Command
Step 1	Enable CHAP or PAP.	**ppp authentication** {**chap** | **pap**} [**if-needed**] or **ppp authentication** {**chap** | **pap**} [*list-name*]
Step 2	Configure host authentication or configure TACACS.	**username** *name* **password** *secret* **ppp use-tacacs [single-line]**

The optional keyword **if-needed** can be used only with TACACS or extended TACACS. The optional argument *list-name* can only be used with AAA/TACACS+. CHAP is specified in RFC 1334. It is an additional authentication phase of the PPP Link Control Protocol.

Caution If you use a *list-name* that has not been configured with the **aaa authentication ppp** command, you disable PPP on this line.

To specify the password to be used in CHAP or PAP caller identification, perform the following task in global configuration mode:

Task	Command
Configure authentication.	**username** *name* **password** *secret*[1]

1. This command is documented in the "System Management Commands" chapter of the *Router Products Command Reference* publication.

Make sure that this password does not contain spaces or underscores.

For an example of CHAP, see the section "Example of CHAP with an Encrypted Password" at the end of this chapter. CHAP and PAP are specified in the IETF RFC 1334, "The PPP Authentication Protocols." CHAP is specified as an additional authentication phase of the PPP Link Control Protocol.

Enable Link Quality Monitoring (LQM)

Link Quality Monitoring (LQM) is available on all serial interfaces running PPP. LQM will monitor the link quality, and if the quality drops below a configured percentage, the link will be taken down. The percentages are calculated for both the incoming and outgoing directions. The outgoing quality is calculated by comparing the total number of packets and bytes sent with the total number of packets and bytes received by the peer. The incoming quality is calculated by comparing the total number of packets and bytes received with the total number of packets and bytes sent by the peer.

When LQM is enabled, Link Quality Reports (LQRs) are sent every keepalive period. LQRs are sent in place of keepalives. All incoming keepalives are responded to properly. If LQM is not configured, keepalives are sent every keepalive period and all incoming LQRs are responded to with an LQR.

LQR is specified in the IETF RFC-1333, "PPP Link Quality Monitoring," by William A. Simpson of Computer Systems Consulting Services.

To enable LQM on the interface, perform the following task in interface configuration mode:

Task	Command
Enable LQM on the interface.	**ppp quality** *percentage*

The *percentage* argument specifies the link quality threshold. That percentage must be maintained, or the link is deemed to be of poor quality and taken down.

Configure Compression of PPP Data

You can configure point-to-point software compression on serial interfaces that use PPP encapsulation. Compression reduces the size of a PPP frame via lossless data compression. The compression algorithm used is a predictor algorithm (the RAND algorithm), which uses a compression dictionary to predict what the next character in the frame will be.

PPP encapsulations support both predictor and Stacker compression algorithms.

Compression is performed in software and might significantly affect system performance. We recommend that you disable compression if CPU load exceeds 65%. To display the CPU load, use the **show process cpu** EXEC command.

If the majority of your traffic is already compressed files, it is recommended that you not use compression.

To configure compression over PPP, perform the following tasks in interface configuration mode:

Task		Command
Step 1	Enable encapsulation of a single protocol on the serial line.	**encapsulation ppp**
Step 2	Enable compression.	**compress** [**predictor** | **stac**] or **ppp compress** [**predictor** | **stac**]

Configure Compression of LAPB Data

You can configure point-to-point software compression on serial interfaces that use LAPB or multi-LAPB encapsulation. Compression reduces the size of a LAPB or multi-LAPB frame via lossless data compression. The compression algorithm used is a predictor algorithm (the RAND algorithm), which uses a compression dictionary to predict what the next character in the frame will be.

Compression is performed in software and may significantly affect system performance. We recommend that you disable compression if CPU load exceeds 65%. To display the CPU load, use the **show process cpu** EXEC command.

Predictor compression is recommended when the bottleneck is the load on the router; Stacker compression is recommended when the bottleneck is line bandwidth. Compression is not recommended if the majority of your traffic is already compressed files. Compression is also not recommended for line speeds greater than T1; the added processing time slows performance on such lines.

If the majority of your traffic is already compressed files, it is recommended that you not use compression.

To configure compression over LAPB, perform the following tasks in interface configuration mode:

Task		Command
Step 1	Enable encapsulation of a single protocol on the serial line.	**encapsulation lapb**
Step 2	Enable compression.	**compress** [**predictor** | **stac**]

To configure compression over multi-LAPB, perform the following tasks in interface configuration mode:

Task		Command
Step 1	Enable encapsulation of multiple protocols on the serial line.	**encapsulation lapb multi**
Step 2	Enable compression.	**compress** [**predictor** | **stac**]

When you are using compression, adjust the MTU for the serial interface and the LAPB N1 parameter as in the following example, to avoid informational diagnostics regarding excessive MTU or N1 sizes:

```
interface serial 0
encapsulation lapb
compress predictor
mtu 1509
lapb n1 12072
```

For information about configuring X.25 TCP/IP header compression and X.25 payload compression, see the chapter “Configuring X.25 and LAPB.”

Configure Compression of HDLC Data

You can configure point-to-point software compression on serial interfaces that use HDLC encapsulation. Compression reduces the size of a HDLC frame via lossless data compression. The compression algorithm used is a Stacker (LZS) algorithm.

Compression is performed in software and might significantly affect system performance. We recommend that you disable compression if CPU load exceeds 65%. To display the CPU load, use the **show process cpu** EXEC command.

If the majority of your traffic is already compressed files, you should not use compression.

To configure compression over HDLC, perform the following tasks in interface configuration mode:

Task		Command
Step 1	Enable encapsulation of a single protocol on the serial line.	**encapsulation hdlc**
Step 2	Enable compression.	**compress stac**

Invoke ATM over a Serial Line

If you have an ATM DSU, you can invoke ATM over a serial line.You do so by mapping an ATM virtual path identifier (VPI) and virtual channel identifier (VCI) to a DXI frame address. ATM-DXI encapsulation defines a data exchange interface that allows a DTE (such as a router) and a DCE (such as an ATM DSU) to cooperate to provide a User - Network Interface (UNI) for ATM networks.

To invoke ATM over a serial line, perform the following tasks in interface configuration mode:

Task		Command
Step 1	Specify the encapsulation method.	**encapsulation atm-dxi**
Step 2	Map a given VPI and VCI to a DXI frame address.	**atm-dxi map** *protocol address vpi vci* [**broadcast**]

You can also configure the **atm-dxi map** command on a HSSI interface.

To configure an ATM interface using an AIP card, see the chapter "Configuring ATM."

Configure the CRC

The cyclic redundancy check (CRC) on a serial interface defaults to a length of 16 bits. To change the length of the CRC to 32 bits on an FSIP or HIP of the Cisco 7000 series only, complete the following task in interface configuration mode:

Task	Command
Set the length of the CRC.	**crc** *size*

Use the NRZI Line-Coding Format

Many serial interfaces—including the Hitachi-based interfaces in the Cisco 2500 series, the 4T processor modules, and the Cisco 3000 series with dual serial ports, and all FSIP interface types on the Cisco 7000— support nonreturn-to-zero (NRZ) and nonreturn-to-zero inverted (NRZI) format. This is a line-coding format that is required for serial connections in some environments. NRZ encoding is most common. NRZI encoding is used primarily with RS-232 connections in IBM environments.

The default configuration for all serial interfaces is NRZ format. The default is **no nrzi-encoding**. To enable NRZI format, complete the following task in interface configuration mode:

Task	Command
Enable NRZI encoding format.	**nrzi-encoding**

Enable the Internal Clock

When a DTE does not return a transmit clock, use the following interface configuration command on the Cisco 7000 series to enable the internally generated clock on a serial interface:

Task	Command
Enable the internally generated clock on a serial interface.	**transmit-clock-internal**

Invert the Transmit Clock Signal

Delays between the SCTE clock and data transmission indicate that the transmit clock signal might not be appropriate for the interface rate and length of cable being used. Different ends of the wire may have variances that differ slightly. Invert the clock signal to compensate for these factors by completing the following task in interface configuration mode on a Cisco 7000 series:

Task	Command
Invert the clock signal on an interface.	**invert-transmit-clock**

Set Transmit Delay

It is possible to send back-to-back data packets over serial interfaces faster than some hosts can receive them. You can specify a minimum dead time after transmitting a packet to alleviate this condition. This setting is available for serial interfaces on the MCI and SCI interface cards and for the HSSI. Perform one of the following tasks, as appropriate for your system, in interface configuration mode:

Task	Command
Set the transmit delay on the MCI and SCI synchronous serial interfaces.	**transmitter-delay** *microseconds*
Set the transmit delay on the HSSI.	**transmitter-delay** *hdlc-flags*

Configure DTR Signal Pulsing

You can configure pulsing DTR signals on all serial interfaces.When the serial line protocol goes down (for example, because of loss of synchronization) the interface hardware is reset and the DTR signal is held inactive for at least the specified interval. This function is useful for handling encrypting or other similar devices that use the toggling of the DTR signal to resynchronize.To configure DTR signal pulsing, perform the following task in interface configuration mode:

Task	Command
Configure DTR signal pulsing.	**pulse-time** *seconds*

Ignore DCD and Monitor DSR as Line Up/Down Indicator

This task applies to Quad Serial NIM interfaces on the Cisco 4000 series and Hitachi-based serial interfaces on the Cisco 2500 series and Cisco 3000 series.

By default, when the serial interface is operating in DTE mode, it monitors the Data Carrier Detect (DCD) signal as the line up/down indicator. By default, the attached DCE device sends the DCD signal. When the DTE interface detects the DCD signal, it changes the state of the interface to up.

In some configurations, such as an SDLC multidrop environment, the DCE device sends the Data Set Ready (DSR) signal instead of the DCD signal, which prevents the interface from coming up. To tell the interface to monitor the DSR signal instead of the DCD signal as the line up/down indicator, perform the following task in interface configuration mode:

Task	Command
Configure the serial interface to monitor the DSR signal as the line up/down indicator.	**ignore-dcd**

Configure the Clock Rate on DCE Appliques

You can configure the clock rate for appliques (connector hardware) on the serial interface of the MCI and SCI cards to an acceptable bit rate. To do so, perform the following task in interface configuration mode:

Task	Command
Configure the clock rate on serial interfaces.	**clock rate** *bps*

Specify the Serial Network Interface Module Timing

On Cisco 4000 series routers, you can specify the serial Network Interface Module timing signal configuration. When the board is operating as a DCE and the DTE provides terminal timing (SCTE or TT), you can configure the DCE to use SCTE from the DTE. When running the line at high speeds and long distances, this strategy prevents phase shifting of the data with respect to the clock.

To configure the DCE to use SCTE from the DTE, perform the following task in interface configuration mode:

Task	Command
Configure the DCE to use SCTE from the DTE.	**dce-terminal-timing enable**

When the board is operating as a DTE, you can invert the TXC clock signal it gets from the DCE that the DTE uses to transmit data. Invert the clock signal if the DCE cannot receive SCTE from the DTE, the data is running at high speeds, and the transmission line is long. Again, this prevents phase shifting of the data with respect to the clock.

To configure the interface so that the router inverts the TXC clock signal, perform the following task in interface configuration mode:

Task	Command
Specify timing configuration to invert TXC clock signal.	**dte-invert-txc**

Specify G.703 Interface Options

This section describes the optional tasks for configuring a G.703-E1 interface on a Cisco 4000 router or Cisco 7000 series router.

- Enable Framed Mode
- Enable CRC4 Generation
- Use Time Slot 16 for Data
- Specify a Clock

Enable Framed Mode

G.703-E1 interfaces have two modes of operation: framed and unframed. By default, G.703-E1 interfaces are configured for unframed mode. To enable framed mode, perform the following task in interface configuration mode:

Task	Command
Enable framed mode.	**timeslot** *start-slot* - *stop-slot*

To restore the default, use the **no** form of this command or set the starting time slot to 0.

Enable CRC4 Generation

By default, the G.703-E1 CRC4 is not generated. To enable generation of the G.703-E1 CRC4, which is useful for checking data integrity while operating in framed mode, perform the following task in interface configuration mode:

Task	Command
Enable CRC4 generation.	**crc4**

Use Time Slot 16 for Data

By default, time slot 16 is used for signaling. It can also be used for data. To control the use of time slot 16 for data, perform the following task in interface configuration mode:

Task	Command
Specify that time slot 16 is used for data.	**ts16**

Specify a Clock

A G.703-E1 interface can clock its transmitted data from either its internal clock or from a clock recovered from the line's receive data stream. By default, the applique uses the line's receive data stream. To control which clock is used, perform the following task in interface configuration mode:

Task	Command
Specify the clock used for transmitted data.	**clock source** {**line** | **internal**}

Configure a Token Ring Interface

Support for the Token Ring interface is supplied on one of our Token Ring network interface cards:

- The 4/16-Mbps Token Ring cards, which interconnect network servers to IEEE 802.5 and IBM-compatible Token Ring media at speeds of 4 or 16 Mbps. The 4/16-Mbps cards are the CSC-C2CTR,CSC-R16 (or CSC-R16M), CSC-1R, and CSC-2R (dual Token Ring card).
- On the Cisco 7000 series, the high-speed Token Ring Interface Processor (TRIP) that has two or four DB-9 ports and interconnects network servers to IEEE 802.5 and IBM-compatible Token Ring media.

The Token Ring interface supports both routing (Layer 3 switching) and source-route bridging (Layer 2 switching). The use of routing and bridging is on a per-protocol basis. For example, IP traffic could be routed while SNA traffic is bridged. The routing support interacts correctly with source-route bridges.

Support for the Token Ring MIB variables is provided as described in RFC 1231, "IEEE 802.5 Token Ring MIB," by K. McCloghrie, R. Fox, and E. Decker, May 1991. The mandatory Interface Table and Statistics Table are implemented, but the optional Timer Table of the Token Ring MIB is not. The Token Ring MIB has been implemented for the TRIP.

Use the **show interfaces**, **show controllers token**, and **show controllers cbus** EXEC commands to display the Token Ring numbers. These commands provide a report for each ring supported by the router.

Note If the system receives an indication of a cabling problem from a Token Ring interface, it puts that interface into a reset state and does not attempt to restart it. It functions this way because periodic attempts to restart the Token Ring interface have a drastic impact on the stability of protocol routing tables. Once you have replugged the cable into the MAU, restart the interface by typing the command **clear interface tokenring** *number,* where *number* is the interface number.

The Token Ring interface by default uses the SNAP encapsulation format defined in RFC 1042. It is not necessary to define an encapsulation method for this interface.

Token Ring Task List

Perform the tasks in the following sections configure a Token Ring interface. The first task is required; the remaining tasks are optional, unless you are configuring a Token Ring on the CSC-1R or CSC-2R, in which case you must also select a Token Ring speed.

- Specify a Token Ring Interface
- Select the Token Ring Speed
- Enable Early Token Release
- Configure PCbus Token Ring Interface Management

Specify a Token Ring Interface

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

Task	Command
Begin interface configuration.	**interface tokenring** *number*
Begin interface configuration for the Cisco 7000 series.	**interface tokenring** *slot/port*

Select the Token Ring Speed

The Token Ring interface on the CSC-1R and CSC-2R can run at either 4 or 16 Mbps. These Token Ring interfaces do not default to any particular ring speed; you must select the speed the first time you use them.

Caution Configuring a ring speed that is wrong or incompatible with the connected Token Ring causes the ring to beacon, which effectively takes the ring down and makes it nonoperational.

Configure the ring speed on the CSC-1R or CSC-2R Token Ring interfaces by performing the following task in interface configuration mode:

Task	Command
Select the ring speed.	**ring-speed** *speed*

Enable Early Token Release

Our Token Ring interfaces support early token release, a method whereby the interface releases the token back onto the ring immediately after transmitting rather than waiting for the frame to return. This feature can help to increase the total bandwidth of the Token Ring. To configure the interface for early token release, perform the following task in interface configuration mode:

Task	Command
Enable early token release.	**early-token-release**

Configure PCbus Token Ring Interface Management

The Token Ring interface on the AccessPro PC card can be managed by a remote LAN manager over the PCbus interface. Currently, the LanOptics Hub Networking Management software running on an IBM compatible PC is supported.

To enable LanOptics Hub Networking Management of a PCbus Token Ring interface, perform the following task in interface configuration mode:

Task	Command
Enable PCbus LAN management.	**local-lnm**

Configure a Tunnel Interface

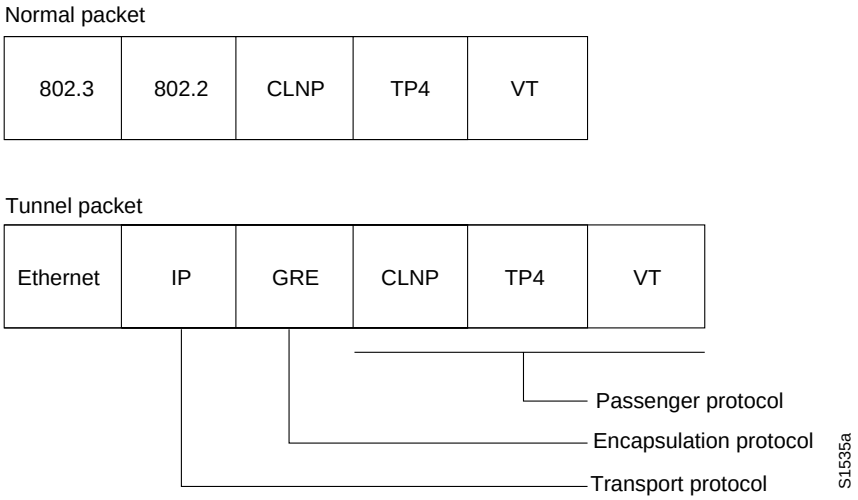
Tunneling provides a way to encapsulate arbitrary packets inside of a transport protocol. This feature is implemented as a virtual interface to provide a simple interface for configuration. The tunnel interface is not tied to specific “passenger” or “transport” protocols, but rather, it is an architecture 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 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
- Transport protocol, which is the protocol used to carry the encapsulated protocol (IP only)

Figure 6-8 illustrates IP tunneling terminology and concepts.

Figure 6-8 IP Tunneling Terminology and Concepts



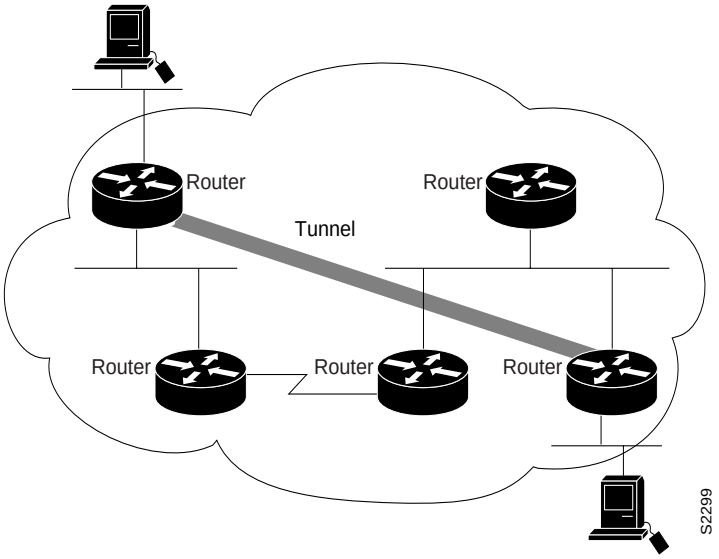
To understand the process of tunneling, consider connecting two AppleTalk networks with a non-AppleTalk backbone, such as IP. The relatively high bandwidth consumed by the broadcasting of Routing Table Maintenance Protocol (RTMP) data packets can severely hamper the backbone's network performance. This 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

There 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 6-9)
- To connect discontinuous subnetworks
- To allow virtual private networks across wide-area networks (WANs)

Figure 6-9 Providing Workarounds for Networks with Limited Hop Counts



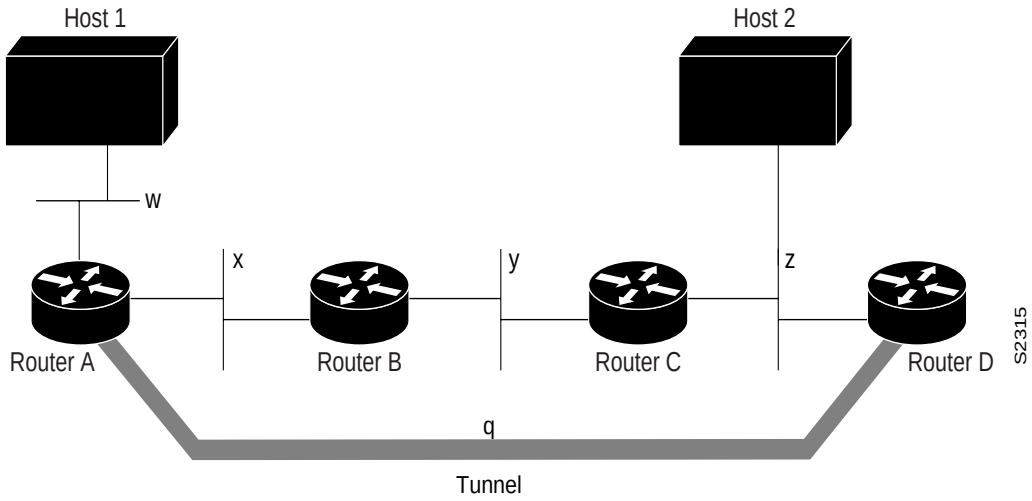
If the path between two computers has more than 15 hops, they cannot communicate with each other, but it is possible to hide some of the hops inside the network with a tunnel.

Special Considerations

The following are considerations and precautions to observe when configuring tunneling:

- Encapsulation and decapsulation at the tunnel endpoints are slow operations; currently only processor switching is supported.
- Be cautious in your configuration and take into account security and topology issues. Be careful not to violate access control lists. You can configure a tunnel with a source and destination that is not restricted by firewall access 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 6-10, 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 6-10 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 Task List

If you want to configure IP tunneling, you must perform at least the first three tasks in the following sections. The remaining tunnel configuration tasks are optional.

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

For commands that monitor IP tunnels, see the section "Monitor and Maintain the Interface" later in this chapter. For examples of configuring tunnels, see the section "Examples of IP Tunneling" 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.	**interface tunnel** *number*
Begin interface configuration for the Cisco 7000 series.	**interface tunnel** *slot/port*

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.	**tunnel source** {*ip-address* | *type number*}

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.	**tunnel destination** {*hostname* | *ip-address*}

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.	**tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre ip** | **nos**}

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 to interoperate with Cayman GatorBoxes. You can have our routers at either end of the tunnel, or you can have a GatorBox at one end and our router at the other end. Use Distance Vector Multicast Routing Protocol (DVMRP) mode when a router connects to a mrouted router 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 our routers at both ends of the tunnel connection. When using 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
Step 1	Enable tunneling on the interface.	**interface tunnel** *number*
Step 2	Assign a cable range to an interface.	**appletalk cable-range** *start-end [network.node]*[1]
Step 3	Set a zone name for the connected AppleTalk network.	**appletalk zone** *zone-name*[2]
Step 4	Specify the interface out which the encapsulated packets will be sent, or specify the router's IP address.	**tunnel source** {*ip-address* | *type*}
Step 5	Specify the IP address of the router at the far end of the tunnel.	**tunnel destination** {*ip-address* | *hostname*}
Step 6	Enable GRE tunneling.	**tunnel mode gre ip**

1. This command is documented in the "AppleTalk Commands" chapter of the *Router Products Command Reference* publication.
2. This command is documented in the "AppleTalk Commands" chapter of the *Router Products Command Reference* publication.

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 routers will drop 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.	**tunnel checksum**

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.

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

Task	Command
Configure a tunnel identification key.	**tunnel key** *key-number*

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.	**tunnel sequence-datagrams**

Understand Subinterfaces

Configuring multiple virtual interfaces, or subinterfaces, on a single physical interface allows greater flexibility and connectivity on the network. A subinterface is a mechanism that allows a single physical interface to support multiple logical interfaces or networks. That is, several logical interfaces or networks can be associated with a single hardware interface. Subinterfaces are implemented in various WAN and LAN protocols, including ATM, Frame Relay, SMDS, X.25, and Novell IPX. For more information about using subinterfaces, refer to the appropriate protocol chapter.

Note A router can support a maximum of 255 interfaces and subinterfaces.

Configure Features Available on Any Interface

The following sections describe optional tasks that you can perform on any type of interface:

- Add a Description for an Interface
- Configure MOP
- Control Interface Hold-Queue Limits
- Set Bandwidth
- Set Interface Delay
- Adjust Timers
- Limit Transmit Queue Size
- Adjust Maximum Packet Size or MTU Size

Add a Description for an Interface

You can add a description about an interface to help you remember what is attached to it. This description is meant solely as a comment to help identify what the interface is being used for. The description will appear in the output of the following commands: **show configuration**, **write terminal**, and **show interfaces**. When you add a description for a T1 controller interface, it will appear in the output of the **show controllers t1** and **write terminal** commands.

To add a description for any interface but a T1 controller interface, perform the following task in interface configuration mode. To add a description for a T1 controller interface, perform the following task in controller configuration mode:

Task	Command
Add a description for an interface.	**description** *string*

For examples of adding interface descriptions, see the section "Examples of Interface Descriptions" at the end of this chapter.

Configure MOP

You can enable MOP on an interface by performing the following task in interface configuration mode:

Task	Command
Enable MOP.	**mop enabled**

You can enable an interface to send out periodic MOP system identification messages on an interface by performing the following task in interface configuration mode:

Task	Command
Enable MOP message support.	**mop sysid**

Control Interface Hold-Queue Limits

Each interface has a hold-queue limit. This limit is the number of data packets that the interface can store in its hold queue before rejecting new packets. When the interface empties one or more packets from the hold queue, it can accept new packets again. You can specify the hold-queue limit of an interface in interface configuration mode as follows:

Task	Command
Specify the maximum number of packets allowed in the hold queue.	**hold-queue** *length* {**in** | **out**}

Set Bandwidth

Higher-level protocols use bandwidth information to make operating decisions. For example, IGRP uses the minimum path bandwidth to determine a routing metric. TCP adjusts initial retransmission parameters based on the apparent bandwidth of the outgoing interface. Perform the following task in interface configuration mode to set a bandwidth value for an interface:

Task	Command
Set a bandwidth value.	**bandwidth** *kilobits*

The bandwidth setting is a routing parameter only; it does not affect the physical interface.

Set Interface Delay

Higher-level protocols might use delay information to make operating decisions. For example, IGRP can use delay information to differentiate between a satellite link and a land link. To set a delay value for an interface, perform the following task in interface configuration mode:

Task	Command
Set a delay value for an interface.	**delay** *tens-of-microseconds*

Setting the delay value sets an informational parameter only; you cannot adjust the actual delay of an interface with this configuration command.

Adjust Timers

To adjust the frequency of update messages, perform the following task in interface configuration mode:

Task	Command
Adjust the frequency with which the router sends messages to itself (Ethernet and Token Ring) or to the other end (HDLC-serial and PPP-serial links) to ensure that a network interface is alive for a specified interface.	**keepalive** [*seconds*]

You also can configure the *keepalive* interval, the frequency at which the router sends messages to itself (Ethernet and Token Ring) or to the other end (HDLC-serial, PPP-serial) to ensure that a network interface is alive. The interval in some previous software versions was 10 seconds; it is now adjustable in one-second increments down to one second. An interface is declared down after three update intervals have passed without receiving a keepalive packet.

When adjusting the keepalive timer for a very low bandwidth serial interface, large packets can delay the smaller keepalive packets long enough to cause the line protocol to go down. You might need to experiment to determine the best value.

Limit Transmit Queue Size

You can control the size of the transmit queue available to a specified interface on the MCI and SCI cards. To limit the size, perform the following task in interface configuration mode:

Task	Command
Limit the size of the transmit queue.	**tx-queue-limit** *number*

Adjust Maximum Packet Size or MTU Size

Each interface has a default maximum packet size or maximum transmission unit (MTU) size. This number generally defaults to 1500 bytes. On serial interfaces, the MTU size varies, but cannot be set smaller than 64 bytes. To adjust the maximum packet size, perform the following task in interface configuration mode:

Task	Command
Adjust the maximum packet size or MTU size.	**mtu** *bytes*

Configure Dial Backup Service

The dial backup service provides protection against WAN downtime by allowing you to configure a backup serial line via a circuit-switched connection.

To configure dial backup, associate a secondary serial interface as a backup to a primary serial interface. This feature requires that an external modem, CSU/DSU device, or ISDN terminal adapter (TA) attached to a circuit-switched service be connected on the secondary serial interface. The external device must be capable of responding to a DTR signal (DTR active) by auto-dialing a connection to a preconfigured remote site.

The dial backup software keeps the secondary line inactive (DTR inactive) until one of the following conditions is met:

- The primary line goes down.
- The transmitted traffic load on the primary line exceeds a defined limit.

These conditions are defined using the interface configuration commands described later in this section.

When the software detects a lost Carrier Detect signal from the primary line device or finds that the line protocol is down, it activates DTR on the secondary line. At that time, the modem, CSU/DSU, or ISDN TA must be set to dial the remote site. When that connection is made, the routing protocol defined for the serial line will continue the job of transmitting traffic over the dialup line.

You can also configure the dial backup feature to activate the secondary line based upon traffic load on the primary line.

The software monitors the traffic load and computes a five-minute moving average. If this average exceeds the value you set for the line, the secondary line is activated, and depending upon how the line is configured, some or all of the traffic will flow onto the secondary dialup line.

You can also specify a value that defines when the secondary line should be disabled and the amount of time the secondary line can take going up or down.

To configure dial backup, perform the following tasks in interface configuration mode:

Task		Command
Step 1	Select a serial interface as a backup line. Select a serial interface on a Cisco 7000.	**backup interface** *type* **backup interface** *type slot*/*port*
Step 2	Enter the load as a percentage of the primary line's available bandwidth.	**backup load** {*enable-threshold* | **never**} {*disable-load* | **never**}
Step 3	Define how much time should elapse before a secondary line is set up or taken down (after a primary line transitions).	**backup delay** {*enable-delay* | **never**} {*disable-delay* | **never**}

See examples of configuring dial backup service in the sections "Examples of Dial Backup Service When the Primary Line Goes Down," "Examples of Dial Backup Service When the Primary Line Reaches Threshold," and "Examples of Dial Backup Service When the Primary Line Exceeds Threshold" at the end of this chapter. See also the chapter "Configuring DDR."

Configure Loopback Detection

When an interface has a backup interface configured, it is often desirable that the backup interface be enabled when the primary interface is either down or in loopback. By default, the backup is only enabled if the primary interface is down. By using the **down-when-looped** command, the backup interface will also be enabled if the primary interface is in loopback. To achieve this condition, perform the following task in interface configuration mode:

Task	Command
Configure an interface to tell the system it is down when loopback is detected.	**down-when-looped**

If testing an interface with the loopback command, you should not have loopback detection configured, or packets will not be transmitted out the interface that is being tested.

Understand Online Insertion and Removal (OIR)

The Cisco 7000 series Online Insertion and Removal (OIR) feature allows you to remove and replace CxBus interface processors while the system is on line. You can shut down the interface processor before removal and restart it after insertion without causing other software or interfaces to shut down.

Note Do not remove or install more that one interface processor at one time. After a removal or installation, observe the LEDs before continuing.

You do not need to notify the software that you are going to remove or install an interface processor. When the route processor is notified by the system that an interface processor has been removed or installed, it stops routing and scans the system for a configuration change. All interface processors are initialized, and each interface type is verified against the system configuration; then the system runs diagnostics on the new interface. There is no disruption to normal operation during interface processor insertion or removal.

Note Only the Cisco 7000 series supports OIR.

Only an interface of a type that has been configured previously will be brought on line; others require configuration. If a newly installed interface processor does not match the system configuration, the interface is left in an administratively down state until the system operator configures the system with the new interfaces.

Hardware (MAC-level) addresses for all interfaces on the Cisco 7000 are stored on an electronically erasable programmable read-only memory (EEPROM) component in the Route Processor (RP) instead of on the individual interface boards. An address allocator in the EEPROM contains a sequential block of 40 addresses (5 interface slots times a maximum of 8 possible ports per slot). Each address is assigned to a specific slot and port address in the chassis, regardless of how the interfaces are configured. This allows interfaces to be replaced online without requiring the system to update routing tables and data structures. Regardless of the types of interfaces installed, the hardware addresses do not change unless you replace the system RP. If you do replace the RP, the hardware addresses of *all* ports change to those specified in the address allocator on the new RP.

Understand Fast, Autonomous, and SSE Switching Support

Switching is the process by which packets in a router are forwarded. Our routers support four kinds of switching: process switching, fast switching, autonomous switching, and silicon switching. For more information about switching and about which platforms, interfaces, and protocols support which types of switching, refer to the "Switching" appendix in the *Router Products Command Reference* publication.

Monitor and Maintain the Interface

You can perform the tasks in the following sections to monitor and maintain the interfaces:

- Monitor Interface Status
- Monitor the Interface Port
- Monitor the T1 or E1 Controller
- Monitor the LAN Extender Interface
- Monitor and Maintain a Hub
- Monitor IP Tunnels
- Clear and Reset the Interface
- Shut Down and Restart the Interface
- Run Interface Loopback Diagnostics

Monitor Interface Status

The software contains commands that you can enter at the EXEC prompt to display information about the interface including the version of the software and the hardware, the controller status, and statistics about the interfaces. The following table lists some of the interface monitoring tasks. (The full list of **show** commands can be displayed by entering the **show ?** command at the EXEC prompt.) These commands are fully described in the *Router Products Command Reference* publication.

Perform the following commands in EXEC mode:

Task	Command
Display the status of the asynchronous interface.	**show async status**
Display compression statistics on a serial interface.	**show compress**
Display current internal status information for the interface controller cards.	**show controllers {bri | cbus | fddi | lance | mci | serial | token}**
For the Cisco 7000.	**show controllers {cxbus | fddi | serial | t1 | token}**
Display the number of packets of each protocol type that have been sent through the interface.	**show interfaces** [*type* {*number*}] [*first*] [*last*] [**accounting**]
For the Cisco 7000.	**show interfaces** [*type slot*/*port*] [**accounting**]
Display the number of packets of each protocol type that have been sent through the asynchronous serial line.	**show interfaces async** [*number*] [**accounting**]
Display the current contents of the routing information field (RIF) cache.	**show rif**

Task	Command
Display the hardware configuration, software version, the names and sources of configuration files, and the boot images.	**show version**[1]

1. This command is documented in the "System Image, Microcode Image, and Configuration File Load Commands" chapter of the *Router Products Command Reference* publication.

Monitor the Interface Port

This section applies to the Cisco 7000 series only. The port adapter cable connected to each port determines the electrical interface type and mode of the port. The default mode of the ports is DCE, which allows you to perform a loopback test on any port without having to attach a port adapter cable. Although DCE is the default, there is no default clock rate set on the interfaces. When there is no cable attached to a port, the software actually identifies the port as "Universal, Cable Unattached" rather than either as a DTE or DCE interface.

Use the **show controller cxbus** command to show information about the interface port. The following example shows an interface port (2/0) that has an RS-232 DTE cable attached and a second port (2/1) that does not have a cable attached:

```
Cisco 7000# show controller cxbus

Switch Processor 7, hardware version 11.1 microcode version 1.4
 512 Kbytes of main memory, 128 Kbytes cache memory, 299 1520 byte buffers
 Restarts: 0 line down, 0 hung output, 0 controller error
FSIP 2, hardware version 3, microcode version 1.0
 Interface 16 - Serial2/0, electrical interface is RS-232 DTE
  31 buffer RX queue threshold, 101 buffer TX queue limit, buffer size 1520
  Transmitter delay is 0 microseconds
 Interface 17 -Serial2/1, electrical interface is Universal (cable unattached)
  31 buffer RX queue threshold, 101 buffer TX queue limit, buffer size 1520
```

To change the electrical interface type or mode of a port online, replace the serial adapter cable and use software commands to restart the interface and, if necessary, reconfigure the port for the new interface. At system startup or restart, the FSIP polls the interfaces and determines the electrical interface type of each port (according to the type of port adapter cable attached). However, it does not necessarily repoll an interface when you change the adapter cable online. To ensure that the system recognizes the new interface type, shut down and reenable the interface after changing the cable.

Monitor the T1 or E1 Controller

This section applies to the Cisco 7000 series only. Because the T1 or E1 line itself is viewed as the controller, perform the following task in EXEC mode to display information about activity on the T1 or E1 line.

Task	Command
Display information about the T1 line.	**show controller t1**
Display information about the E1 line.	**show controller e1**

Alarms, line conditions, and other errors are displayed. The data is updated every 10 seconds. Every 15 minutes, the cumulative data is stored and retained for 24 hours. This means at any one time, up to 96 15-minute accumulations are counted in the data display.

Monitor the LAN Extender Interface

To monitor the LAN Extender interface, the Ethernet interface that resides on the LAN Extender, the serial interface that resides on the LAN Extender, or the serial interface connected to the LAN Extender, perform one or more of the following tasks at the EXEC prompt:

Task	Command
Display hardware and software information about the LAN Extender.	**show controllers lex** [*number*]
Display information on the Cisco 7000 series.	**show controllers lex** [*slot*/*port*]
Display statistics about the LAN Extender interface.	**show interfaces lex** *number* [**ethernet** | **serial**]
Display statistics about the serial interface on the host router that is physically connected to the LAN Extender.	**show interfaces serial** *number* [**accounting**]
Display statistics on the Cisco 7000 series.	**show interfaces serial** *slot*/*port* [**accounting**]

For more complete network troubleshooting information, refer to the *Troubleshooting Internetworking Systems* publication.

Monitor and Maintain a Hub

You can perform the tasks in the following sections to monitor and maintain the hub:

- Shut Down the Hub Port
- Reset the Hub or Clear the Hub Counters
- Monitor the Hub

Shut Down the Hub Port

To shut down or disable a hub port, perform the following tasks, beginning in global configuration mode:

Task	Command
Specify the hub number and the hub port (or range of hub ports) and place you in hub configuration mode.	**hub ethernet** *number port* [*end-port*]
Shut down the hub port.	**shutdown**

See the examples of shutting down a hub port at the end of this chapter in "Examples of Hub Configuration."

Reset the Hub or Clear the Hub Counters

To reset the hub or clear the hub counters, perform one of the following tasks in EXEC mode:

Task	Command
Reset and reinitialize the hub hardware.	**clear hub ethernet** *number*
Clear the hub counters displayed by the **show hub** command.	**clear hub counters** [**ethernet** *number* [*port* [*end-port*]]]

Monitor the Hub

To display hub information, perform the following task in EXEC mode:

Task	Command
Display hub statistics.	**show hub** [**ethernet** *number* [*port* [*end-port*]]]

Monitor IP Tunnels

Complete any of the following tasks in EXEC mode to monitor the IP tunnels you have configured:

Task	Command
List tunnel interface information.	**show interfaces tunnel** *unit* [**accounting**]
List the routes that go through the tunnel.	**show** *protocol* **route**[1]
List the route to the tunnel destination.	**show ip route**[2]

1. This command is documented in a separate chapter for each protocol. For example, the command **show clns route** is documented in the "ISO CLNS Commands" chapter of the *Router Products Command Reference* publication.
2. This command is documented in the "IP Commands" chapter of the *Router Products Command Reference* publication.

Clear and Reset the Interface

To clear the interface counters shown with the **show interfaces** command, enter the following command at the EXEC prompt:

Task	Command
Clear the interface counters.	**clear counters** [*type number*] [**ethernet** | **serial**]
Clear interface counters for the Cisco 7000.	**clear counters** [*type slot/port*]

The command clears all the current interface counters from the interface unless the optional arguments are specified to clear only a specific interface type from a specific slot and port number.

Note This command will not clear counters retrieved using SNMP, but only those seen with the EXEC **show interfaces** command.

Complete the following tasks in EXEC mode to clear and reset interfaces. Under normal circumstances, you do not need to clear the hardware logic on interfaces.

Task	Command
Reset the hardware logic on an interface.	**clear interface** *type number*
Reset the hardware logic on an asynchronous serial line.	**clear line** [*number*][1]
Clear the entire Token Ring RIF cache.	**clear rif-cache**

1. This command is documented in the *Cisco Access Connection Guide*.

Shut Down and Restart the Interface

You can disable an interface. Doing so disables all functions on the specified interface and marks the interface as unavailable on all monitoring command displays. This information is communicated to other network servers through all dynamic routing protocols. The interface will not be mentioned in any routing updates. On serial interfaces, shutting down an interface causes the DTR signal to be dropped. On Token Ring interfaces, shutting down an interface causes the interface to deinsert from the ring. On FDDIs, shutting down an interface causes the optical bypass switch, if present, to go into bypass mode.

To shut down an interface and then restart it, perform the following tasks in interface configuration mode:

Command	Task
Shut down an interface.	**shutdown**
Reenable an interface.	**no shutdown**

To check whether an interface is disabled, use the EXEC command **show interfaces**. An interface that has been shut down is shown as administratively down in the **show interfaces** command display. See examples in the section "Examples of Interface Shutdown" at the end of this chapter.

One reason to shut down an interface is if you want to change the electrical interface type or mode of a Cisco 7000 series port online. You replace the serial adapter cable and use software commands to restart the interface, and if necessary, reconfigure the port for the new interface. At system startup or restart, the FSIP polls the interfaces and determines the electrical interface type of each port (according to the type of port adapter cable attached). However, it does not necessarily repoll an interface when you change the adapter cable online. To ensure that the system recognizes the new interface type, shut down using the **shutdown** command, and reenable the interface after changing the cable. Refer to your hardware documentation for more details.

Run Interface Loopback Diagnostics

You can use a loopback test on lines to detect and distinguish equipment malfunctions between line and modem or Channel Service Unit/Digital Service Unit (CSU/DSU) problems on the network server. If correct data transmission is not possible when an interface is in loopback mode, the interface is the source of the problem. The DSU might have similar loopback functions you can use to isolate the problem if the interface loopback test passes. If the device does not support local loopback, this function will have no effect.

You can specify hardware loopback tests on the Ethernet and synchronous serial interfaces, and all Token Ring interfaces (except the CSC-R 4-megabit card) that are attached to CSU/DSUs and that support the local loopback signal. The CSU/DSU acts as a Data Communications Equipment (DCE) device; the router acts as a Data Terminal Equipment (DTE) device. The local loopback test generates a CSU loop—a signal that goes through the CSU/DSU to the line, then back through the CSU/DSU to the router. The **ping** command can also be useful during loopback operation.

The loopback tests are available on the following interfaces:

- High-Speed Serial Interface (HSSI), including the High-Speed Communications Interface (HSCI) card ribbon cable
- Cisco Multiport Communications Interface (MCI) and Cisco Serial Communication Interface (SCI) synchronous serial interfaces
- MCI and Cisco Multiport Ethernet Controller (MEC) Ethernet interfaces; an Ethernet loopback server is also provided on the Ethernet interfaces

- Ethernet loopback server
- Channelized E1 interfaces (local loopback only)
- Channelized T1 interfaces (local and remote loopback)
- The FDDI (CSC-FCI) card
- Token Ring interfaces

The following sections describe each test.

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

Enable Loopback Testing on the HSSI

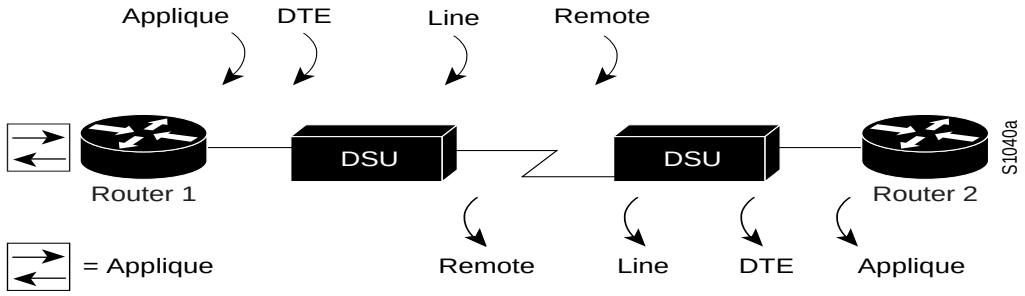
The HSSI allows you to do the following:

- Enable Loopback Test of the HSSI Applique
- Enable Loopback Test to the DTE
- Enable Loopback Test through the CSU/DSU
- Enable Loopback Test over Remote DS-3 Link
- Enable HSSI Externally Requested Loopback
- Perform HSCI Card Ribbon Cable Loopback Test

These tests apply only when the device supports them and are used to check the data communications channels. The tests are usually performed at the line port rather than the DTE port of the remote CSU/DSU.

The internal loopback concepts are illustrated in Figure 6-11.

Figure 6-11 HSSI Loopback Testing



Enable Loopback Test of the HSSI Applique

You can configure an internal loop on the HSSI applique by performing the following task in interface configuration mode:

Task	Command
Loop internally on the HSSI applique.	**loopback applique**

Once enabled, the **loopback applique** command loops the packets on the applique, thereby establishing a loopback inside the router. This command is useful for sending pings to yourself to check the functionality of the applique. The HSSI applique (HSA card) uses an internal 44.736-MHz crystal clock during the applique loopback to drive its internal circuits. Refer to your hardware installation and maintenance publication for more information.

This command is functionally equivalent to entering the **loopback** command with no arguments; however, when the HSCI card is installed, the configuration displayed after the **write terminal** command is entered will show loopback applique set.

Enable Loopback Test to the DTE

You can loop packets to DTE within the CSU/DSU at the DTE interface, when the device supports this function. Doing so is useful for testing the DTE-to-DCE cable. To loop the packets to DTE, perform the following task in interface configuration mode:

Task	Command
Loop packets to DTE internally.	**loopback dte**

Enable Loopback Test through the CSU/DSU

You can loop packets completely through the CSU/DSU to configure a CSU loop, when the device supports this feature. Doing so is useful for testing the DCE device (CSU/DSU) itself. To configure a CSU loop, perform the following task in interface configuration mode:

Task	Command
Loop packets completely through the CSU/DSU.	**loopback line**

Enable Loopback Test over Remote DS-3 Link

You can loop packets through the CSU/DSU, over the Digital signal level 3 (DS-3) link, and to the remote CSU/DSU and back. To do so, perform the following task in interface configuration mode:

Task	Command
Loop packets through the CSU/DSU to a remote CSU/DSU over the DS-3 link.	**loopback remote**

This command applies only when the device supports the remote function. It is used for testing the data communication channels. The loopback usually is performed at the line port, rather than the DTE port, of the remote CSU/DSU.

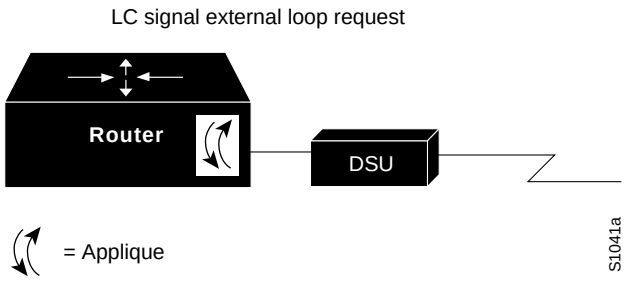
Enable HSSI Externally Requested Loopback

The HSA applique on the HSSI contains an LED that indicates the LA, LB, and LC signals transiting through the devices. The CSU/DSU uses the LC signal to request a loopback from the router. The CSU/DSU might want to do this so that its own network management diagnostics can independently check the integrity of the connection between the CSU/DSU and the router.

When the CSU/DSU asserts the LC signal and the router enables the external loopback, the connection is blocked by the loopback and the router no longer has access to the data communication channel.

Figure 6-12 illustrates the extent of the signal during an external loopback request.

Figure 6-12 HSSI External Loopback Request



By default, this feature is disabled on the router. To enable this feature to support those CSU/DSUs that support this function, perform the following task in interface configuration mode:

Task	Command
Enable a two-way internal and external loopback request on HSSI from DSU/CSU.	**hssi external-loop-request**

If your CSU/DSU does not support this feature, it should not be enabled in the router. This prevents spurious line noise from accidentally tripping the external loopback request line, which would interrupt the normal data flow.

Perform HSCI Card Ribbon Cable Loopback Test

A useful diagnostic is available that allows fault isolation of possible defects on the HSCI card. This diagnostic is not part of the normal system diagnostics, but is offered to help technicians test for controller defects at installation or when the system is upgraded. The diagnostic involves recabling the HSCI card and then entering a diagnostic script. The tasks required to perform this diagnostic are described in the hardware installation and maintenance publication for your router.

Enable Loopback on MCI and SCI Serial Cards

The MCI and SCI serial interface cards support the loopback function when a CSU/DSU or equivalent device is attached to the router. To enable loopback mode on them, perform the following task in interface configuration mode:

Task	Command
Enable loopback through a CSU/DSU to configure a CSU loop on the MCI and SCI synchronous serial interfaces.	**loopback**

Enable Loopback on MCI and MEC Ethernet Cards

The Ethernet interfaces on the MCI and MEC cards support loopback mode. To enable loopback mode on them, perform the following task in interface configuration mode:

Task	Command
Enable loopback to verify that the interface receives back every packet it sends.	**loopback**

Configure the Ethernet Loopback Server

The router software provides an Ethernet loopback server that supports Digital Equipment Corporation (Digital), Intel, and Xerox systems specified by the "blue book," a joint specification written by Digital, Intel, and Xerox that defines the Ethernet protocol. The loopback server responds to forward data loopback messages sent either to the server's MAC address or to the broadcast address. Currently, the Ethernet loopback server does not respond to the loopback assistance multicast address.

Use the Ethernet loopback server to test communications between your internetworking products and Digital systems that do not support the IP **ping** command, such as DECnet-only VMS systems.

To originate a loop test on your VMS system with a Cisco server, use the Digital Network Control Program (NCP) command **Loop Circuit**. For more information about the **Loop Circuit** command, consult the DECnet VAX documentation. Cisco network servers support all options that can be specified by the VMS hosts.

Troubleshooting Channelized E1 and Channelized T1

When troubleshooting channelized T1 or E1, you must first decide if the problem is with a particular channel group, or with the T1 or E1 line.

If the problem is with a single channel group, you have an potential interface problem.

If the problem is with the T1 or E1 line, or with all channel groups, you have a potential controller problem.

Troubleshooting Channelized T1/E1 Controllers

When you troubleshoot E1 or T1 controllers, first check that the configuration is correct. The framing type and line code should match to what the service provider has specified. Then check channel group and PRI-group configurations, especially to verify that the timeslots and speeds are what the service provider has specified.

At this point, the **show controller t1** or **show controller e1** commands should be used to check for T1 or E1 errors. Use the command several times to determine if error counters are increasing, or if the line status is continually changing. If this is occurring, you need to work with the service provider.

Note Cisco routers do not have CSU capability and do not react to any remote loopback codes at the T1 or E1 level.

Channelized T1 Controller Loopbacks

For the T1 controller, two loopbacks are available for testing.

- Local loopback
- Remote loopback

Local Loopback. The local loopback loops the controller both toward the router, and toward the line. Since the loopback is done internally to the router, the controller should transition to the UP state within approximately 10 seconds, and no further T1 errors should be detected.

All channel groups will be looped back; if the encapsulation on that channel group supports loopbacks (for example, HDLC and PPP), you can test that channel group by pinging the interface address. For example, if you have assigned an IP address to the serial interface defined for a channel group, you can ping that IP address.

To place the controller into local loopback, perform the following task in controller configuration mode.

Task	Command
Loop the T1 controller toward the router and toward the line.	**loopback local** (controller)

To test a channel group, perform the following task in EXEC mode:

Ping the interface address.	**ping** *protocol protocol-address*

Check errors if any. by performing the following task in EXEC mode:

Check errors.	**show controller t1**

If any errors occur, or the controller fails to change to the UP state, please contact the Cisco TAC.

Since the controller local loopback is bidirectional, the service provider can test the line integrity using a T1 BERT test set.

Remote Loopback. The second T1 controller loopback is a remote loopback. This loopback can be used only if the *entire* T1 goes to a remote CSU. This is not the case with 99.9% of channelized T1. When the **loopback remote** controller command is executed, an inband CSU loop-up code will be sent over the entire T1, which will attempt to loop up the remote CSU. To place the controller in remote loopback, perform the following task in controller configuration mode:

Task	Command
Place the T1 controller in remote loopback.	**loopback remote** (controller)

Note If controller loopbacks are used, they will disrupt service for all channel groups on that interface.

Channelized E1 Controller Loopback

For the E1 controller, only the local loopback is available. Local loopback operates the same as the local loopback on the T1 controller, forming a bidirectional loopback, both toward the router, and toward the line.To place the E1 controller in local loopback, perform the following task in controller configuration mode:

Task	Command
Place the E1 controller in local loopback toward the router and toward the line.	**loopback** (controller)

All channel groups will be looped back; if the encapsulation on that channel group supports loopbacks (for example, HDLC and PPP), you can test that channel group by pinging the interface address. For example, if you have assigned an IP address to the serial interface defined for a channel group, you can ping that IP address.

To place the controller into local loopback, perform the following task in controller configuration mode.

Task	Command
Loop the T1 controller toward the router and toward the line.	**loopback local** (controller)

To test a channel group, perform the following task in EXEC mode:

Ping the interface address.	**ping** *protocol protocol-address*

Check errors if any. by performing the following task in EXEC mode:

Check errors.	**show controller t1**

If any errors occur, it is most likely a hardware problem; please contact the Cisco TAC. At the same time, the service provider can test the line by using a T1 BERT test set.

Troubleshooting Channelized T1/E1 Channel Groups

Each channelized T1 or channelized E1 channel group is treated as a separate serial interface. To troubleshoot channel groups, first verify configurations and check everything that is normally checked for serial interfaces. You can verify that the timeslots and speed are correct for the channel group by checking for CRC errors and aborts on the incoming line.

Channelized T1/E1 Channel Group Loopbacks

Note None of the Cisco channelized interfaces will react to any loop codes. To loop a channelized interface requires that the configuration command be entered manually.

Two loopbacks are available for channel groups:

- Interface local loopback
- Interface remote loopback

Interface Local Loopback. Interface local loopback is a bi-directional loopback, which will loopback toward the router and toward the line. The entire set of timeslots for the channel group are looped back. The service provider can use a BERT test set to test the link from the central office to your local router, or the remote router can test using pings to their local interface (which will go from the remote site, looped back at your local site, and return to the interface on the remote site).

To place the serial interface (channel group) into local loopback, perform the following task in interface configuration mode:

Task	Command
Place the serial interface (channel group) in local loopback,	**loopback local**

Interface Remote Loopback. Remote loopback is the ability to put the remote DDS CSU/DSU in loopback. It will work only with channel groups that have a single DS0 (1 timeslot), and with equipment that works with a latched CSU loopback as specified in AT&T specification TR-TSY-000476, "OTGR Network Maintenance Access and Testing." To place the serial interface (channel group) in remote loopback, perform the following task in interface configuration mode:

Task	Command
Place the serial interface (channel group) in remote loopback.	**loopback remote** (interface)

Using the **loopback remote** interface command sends a latched CSU loopback command to the remote CSU/DSU. The router must detect the response code, at which time the remote loopback is verified.

Enable Loopback on the CSC-FCI FDDI Card

You can place the FDDI (CSC-FCI) into loopback mode by performing the following task in interface configuration mode:

Task	Command
Enable loopback to verify that the FDDI (CSC-FCI) interface receives back every packet it sends.	**loopback**

Enable Loopback on Token Ring Cards

You can place all of the Token Ring interface cards except the 4-MB CSC-R card into loopback mode by performing the following task in interface configuration mode:

Task	Command
Enable loopback to verify that the Token Ring interface receives back every packet it sends.	**loopback**

Interface Configuration Examples

Use the configuration examples in this section to help you understand some aspects of interface configuration. More complex and realistic examples appear in the chapters that describe special interface configuration and routing and bridging configuration.

- Examples of Enabling Interface Configuration
- Example of a Dedicated Asynchronous Interface
- Example of Restricting Access on the Asynchronous Interface
- Example of Asynchronous Routing and Dynamic Addressing
- Examples of Channelized T1 Controller and Interface

- Example of Enabling Ethernet Encapsulation
- Example of Enabling a LAN Extender Interface
- Examples of LAN Extender Interface Access List
- Examples of DHCP
- Example of CHAP with an Encrypted Password
- Examples of IP Tunneling
- Examples of Interface Descriptions
- Examples of Interface Shutdown
- Examples of Dial Backup Service When the Primary Line Goes Down
- Examples of Dial Backup Service When the Primary Line Reaches Threshold
- Examples of Dial Backup Service When the Primary Line Exceeds Threshold
- Examples of Hub Configuration

Examples of Enabling Interface Configuration

The following example illustrates how to begin interface configuration on a serial interface. It assigns Point-to-Point (PPP) encapsulation to serial interface 0.

```
interface serial 0
encapsulation ppp
```

The same example on a Cisco 7000 requires the following commands:

```
interface serial 1/0
encapsulation ppp
```

Example of a Dedicated Asynchronous Interface

The following example assigns an IP address to an asynchronous interface and places the line in dedicated network mode:

```
interface async 1
async default ip address 182.32.7.51
async mode dedicated
```

Example of Restricting Access on the Asynchronous Interface

The following example assumes that users are restricted to certain servers designated as asynchronous servers, but that normal terminal users can access anything on the local network.

```
! access list for normal connections
access-list 1 permit 131.108.0.0 0.0.255.255
!
access-list 2 permit 131.108.42.55
access-list 2 permit 131.108.111.1
access-list 2 permit 131.108.55.99
!
line 1
speed 19200
flow hardware
modem inout
interface async 1
```

```
async mode interactive
async dynamic address
ip access-group 1 out
ip access-group 2 in
```

Example of Asynchronous Routing and Dynamic Addressing

The following example shows a simple configuration that allows routing and dynamic addressing. In this configuration, the router will act as either a telecommuting server or a router, depending on whether the user specifies **/routing** in the EXEC **slip** or **ppp** command.

```
interface async 1
async dynamic routing
async dynamic address
async mode interactive
```

Examples of Channelized T1 Controller and Interface

This example applies only to a Cisco 7000 series. It configures the router to acknowledge a T1 line and its circuits. Four different circuits are defined for the second CxCT1 attached to the MIP in slot 4.

```
controller t1 4/1
framing esf
linecode b8zs
channel-group 0 timeslots 1
channel-group 8 timeslots 5,7,12-15, 20 speed 64
channel-group 12 timeslots 2
channel-group 23 timeslots 24
```

The following example configures circuit 0 for Point-to-Point (PPP) encapsulation:

```
interface serial 4/1:0
ip address 131.108.13.1 255.255.255.0
encapsulation ppp
```

The following example configures circuit 8 for IP routing and disables IP route cache:

```
interface serial 4/1:8
ip address 131.108.1.1 255.255.255.0
no ip routecache
```

The following example configures circuit 12 for Frame Relay encapsulation and subinterface support:

```
interface serial 4/1:12
encapsulation frame-relay
!
interface serial 4/1:12.1
ip address 1.1.1.1 255.0.0.0
!
interface serial 4/1:12.2
ip address 2.2.2.2 255.0.0.0
```

The following example configures circuit 23 for IP routing and enables autonomous switching:

```
interface serial 4/1:23
ip address 3.3.3.3 255.0.0.0
ip routecache cbus
```

Example of Enabling Ethernet Encapsulation

These commands enable standard Ethernet Version 2.0 encapsulation on the Ethernet interface processor in slot 4 on port 2 of a Cisco 7000:

```
interface ethernet 4/2
encapsulation arpa
```

Example of Enabling a LAN Extender Interface

The following simple example configures and creates a LAN Extender interface. In this example, the MAC address of the LAN Extender is 0000.0c00.0001.

```
interface serial 4
encapsulation ppp
interface lex 0
lex burned-in-address 0000.0c00.0001
ip address 131.108.172.21 255.255.255.0
```

Examples of LAN Extender Interface Access List

This section provides the following examples:

- Example of Filtering by MAC Address
- Example of Filtering by Ethernet Type Code

Example of Filtering by MAC Address

The following is an example that controls which traffic from Macintosh computers on the remote Ethernet LAN reaches the core router:

```
access-list 710 permit 0800.0298.0000 0000.0000.FFFF
access-list 710 deny 0800.0276.2917 0000.0000.0000
access-list 710 permit 0800.0000.0000 0000.FFFF.FFFF
interface lex 0
lex input-address-list 710
```

The first line of this access list permits traffic from any Macintosh whose MAC address starts with 0800.0298. The remaining two octets in the MAC address can be any value because the mask for these octets is FFFF (“don’t care” bits).

The second line specifically rejects all traffic originating from a Macintosh with the MAC address of 0800.0276.2917. Note that none of the mask bits are “don’t care” bits.

The third line specifically permits all traffic from other Macintoshes whose MAC addresses start with 0800. Note that in the mask, the “don’t care” bits are the rest of the address.

At the end of the list is an implicit “deny everything” entry, meaning that any address that does not match an address or address group on the list is rejected.

Example of Filtering by Ethernet Type Code

Using the same configuration as in the previous section, you could allow only the Macintosh traffic by Ethernet type code with the following access list:

```
access-list 220 permit 0x809B 0x0000
interface lex 0
lex input-type-list 220
```

This access list permits only those messages whose protocol number matches the masked protocol number in the first line. The implicit last entry in the list is a “deny everything” entry.

Examples of DHCP

The following global configuration example enables DHCP proxy-client status on all asynchronous interfaces on the communication server:

```
ip address-pool dhcp-proxy-client
```

The following global configuration example illustrates how to designate DHCP servers for use on your network. You can specify up to four severs using IP addresses or names. If you do not designate servers, the IP limited broadcast address of 255.255.255.255 is used by default for transactions with any and all discovered DHCP servers.

```
ip dhcp-server jones smith wesson
```

The following interface configuration example illustrates how to disable DHCP proxy-client functionality on asynchronous interface 1:

```
async interface
interface 1
peer default ip address pool
```

Example of CHAP with an Encrypted Password

The following configuration examples enable CHAP on interface serial 0 of three routers.

Configuration of Router yyy

```
hostname yyy
interface serial 0
encapsulation ppp
ppp authentication chap
username xxx password secretxy
username zzz password secretzy
```

Configuration of Router xxx

```
hostname xxx
interface serial 0
encapsulation ppp
ppp authentication chap
username yyy password secretxy
username zzz password secretxz
```

Configuration of Router zzz

```
hostname zzz
interface serial 0
encapsulation ppp
ppp authentication chap
username xxx password secretxz
username yyy password secretzy
```

When you look at the configuration file, the passwords will be encrypted and the display will look similar to the following:

```
hostname xxx
```

```
interface serial 0
encapsulation ppp
ppp authentication chap
username yyy password 7 121F0A18
username zzz password 7 1329A055
```

Examples of IP Tunneling

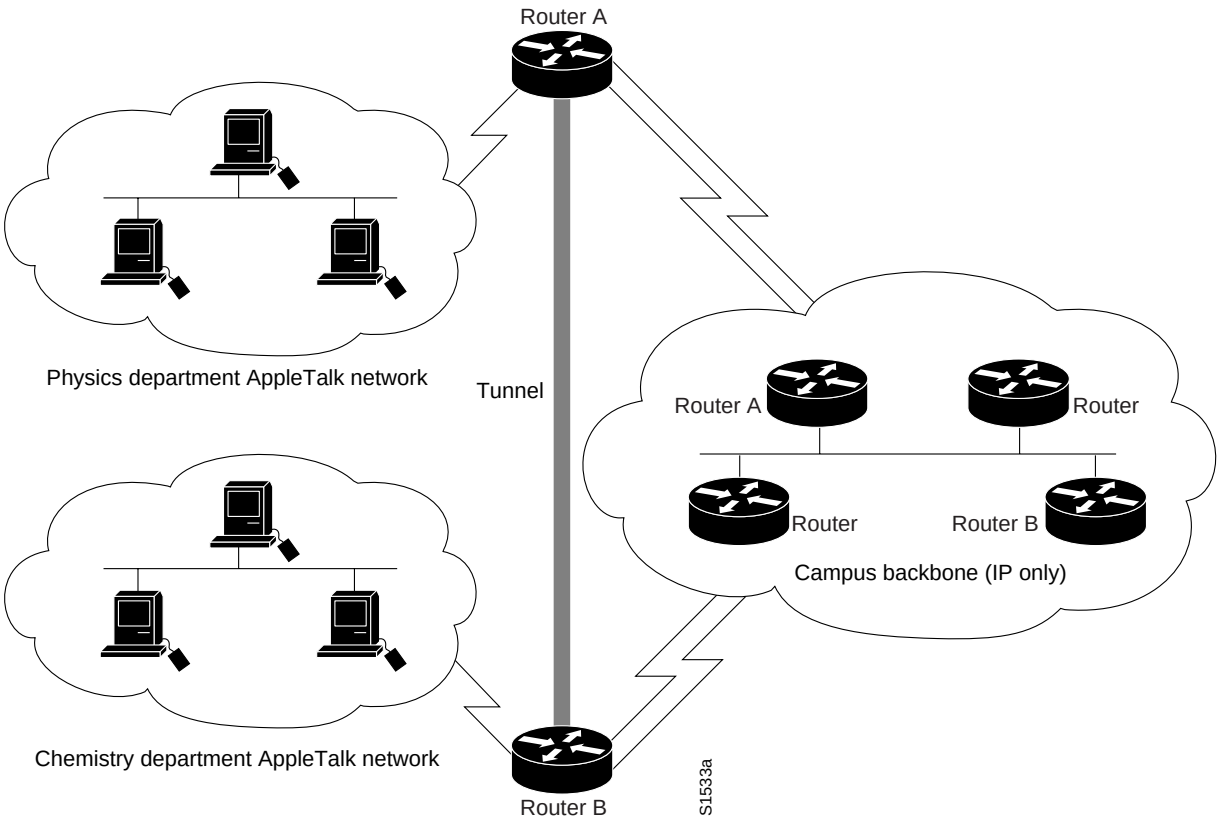
The following example shows an IP tunneling configuration with commented (!) explanations:

```
!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 encapsulator 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
```

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

Figure 6-13 is an example of connecting multiprotocol subnetworks across a single-protocol backbone. The configurations of Router A and Router B follow.

Figure 6-13 Connecting Multiprotocol Subnetworks across a Single-Protocol Backbone



Router A

```
interface ethernet 0
description physics department AppleTalk lan
AppleTalk cable-range 4001-4001 32
!
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
```

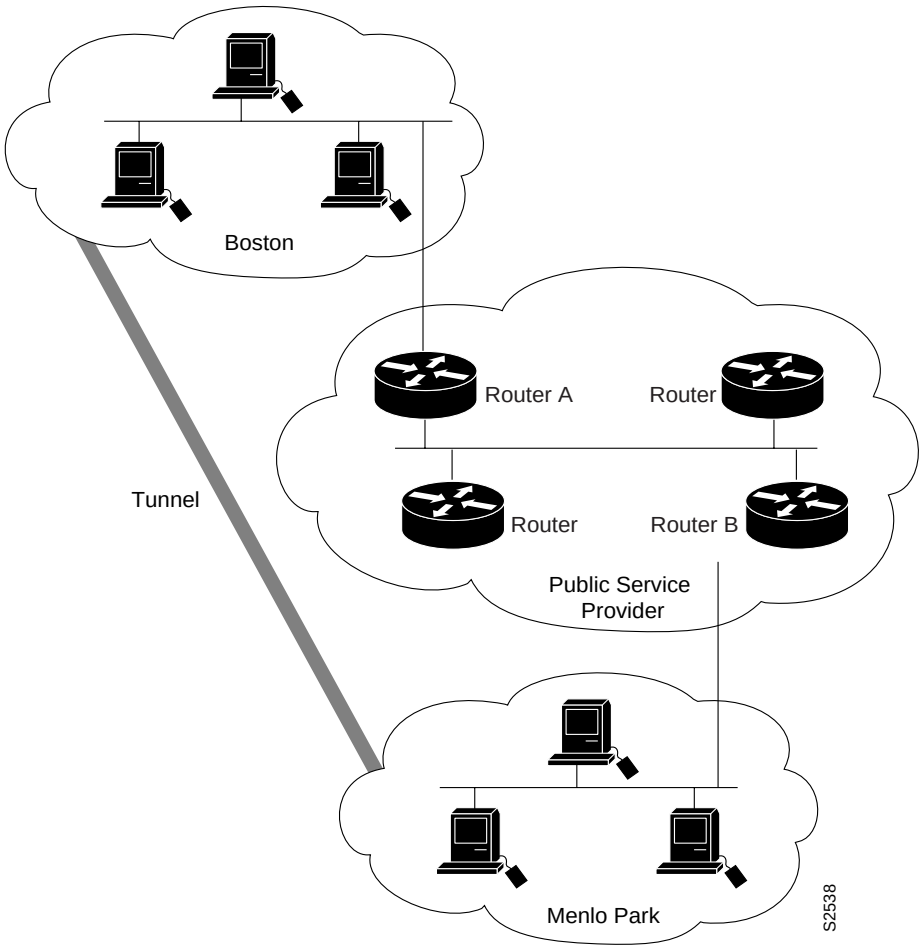
Router B

```
interface ethernet 0
description chemistry department appletalk lan
AppleTalk cable-range 9458-9458 3
!
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
```

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

Figure 6-14 is an example of routing a private IP network and a Novell network across a public service provider.

Figure 6-14 Creating Virtual Private Networks across WANs



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
```

Examples of Interface Descriptions

The following example illustrates how to add a description about an interface that will appear in configuration files and monitoring command displays.

```
interface ethernet 0
description First Ethernet in network 1
ip address 101.13.15.78 255.255.255.0
```

The following example for a Cisco 7000 describes an administration network attached to the Ethernet processor in slot 2, port 4:

```
interface ethernet 2/4
description 2nd floor administration net
```

Examples of Interface Shutdown

The following example turns off the Ethernet interface in slot 2 at port 4:

```
interface ethernet 2/4
shutdown
```

The following example turns the interface back on:

```
interface ethernet 2/4
no shutdown
```

The following example illustrates how to shut down a Token Ring interface:

```
interface tokenring 0
shutdown
```

The following example shuts down a T1 circuit number 23 running on a Cisco 7000:

```
interface serial 4/0:23
shutdown
```

The following next example shuts down the entire T1 line physically connected to a Cisco 7000:

```
controller t1 4/0
shutdown
```

Examples of Dial Backup Service When the Primary Line Goes Down

The following example configures serial 1 as a secondary line that activates only when the primary line (serial 0) goes down. The secondary line will not be activated because of load on the primary.

```
interface serial 0
backup interface serial 1
backup delay 30 60
```

The secondary line is configured to activate 30 seconds after the primary line goes down and to remain on for 60 seconds after the primary line is reactivated.

The same example on the Cisco 7000 would be as follows:

```
interface serial 1/1
backup interface serial 2/2
backup delay 30 60
```

Examples of Dial Backup Service When the Primary Line Reaches Threshold

The following example configures the secondary line (serial 1) to be activated only when the load of the primary line reaches a certain threshold:

```
interface serial 0
backup interface serial 1
backup load 75 5
```

In this case, the secondary line will not be activated when the primary goes down. The secondary line will be activated when the load on the primary line is greater than 75 percent of the primary's bandwidth. The secondary line will then be brought down when the aggregate load between the primary and secondary lines fits within 5 percent of the primary bandwidth.

The same example on the Cisco 7000 would be as follows:

```
interface serial 1/1
backup interface serial 2/2
backup load 75 5
```

Examples of Dial Backup Service When the Primary Line Exceeds Threshold

The following example configures the secondary line to activate once the traffic threshold on the primary line exceeds 25 percent:

```
interface serial 0
backup interface serial 1
backup load 25 5
backup delay 10 60
```

Once the aggregate load of the primary and the secondary lines return to within 5 percent of the primary bandwidth, the secondary line is deactivated. The secondary line waits 10 seconds after the primary goes down before activating, and remains active for 60 seconds after the primary returns and becomes active again.

The same example on the Cisco 7000 is as follows:

```
interface serial 1/1
backup interface serial 2/2
backup load 25 5
backup delay 10 60
```

Examples of Hub Configuration

The following sections provide examples of hub configuration:

- Examples of Hub Port Startup

- Examples of Source Address for an Ethernet Hub Port Configuration
- Examples of Hub Port Shutdown

Examples of Hub Port Startup

The following example configures port 1 on hub 0 of Ethernet interface 0:

```
hub ethernet 0 1
no shutdown
```

The following example configures ports 1 through 8 on hub 0 of Ethernet interface 0:

```
hub ethernet 0 1 8
no shutdown
```

Examples of Source Address for an Ethernet Hub Port Configuration

The following example configures the hub to allow only packets from MAC address 1111.2222.3333 on port 2 of hub 0:

```
hub ethernet 0 2
source-address 1111.2222.3333
```

The following example configures the hub to remember the first MAC address received on port 2, and allow only packets from that learned MAC address:

```
hub ethernet 0 2
source-address
```

Examples of Hub Port Shutdown

The following example shuts down ports 3 through 5 on hub 0:

```
hub ethernet 0 3 5
shutdown
```

The following example shuts down port 3 on hub 0:

```
hub ethernet 0 3
shutdown
```