

# Configuring LAN Interfaces

---

Use the information in this chapter to configure LAN interfaces supported on Cisco routers and access servers.

This chapter describes the processes for configuring LAN interfaces. It contains these sections:

- Configure an Ethernet or Fast Ethernet Interface
- Configure a Fiber Distributed Data Interface (FDDI)
- Configure a Hub Interface
- Configure a LAN Extender Interface
- Configure a Token Ring Interface

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

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

---

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

---

## Configure an Ethernet or Fast Ethernet Interface

Cisco supports both 10 Mbps Ethernet and 100 Mbps Fast Ethernet.

Support for the 10 Mbps and 100 Mbps Ethernet interface is supplied on various Ethernet network interface cards or systems.

The Fast Ethernet NP-1FE Module, for example, provides the following benefits:

- VLAN routing—Virtual LAN (VLAN) support enables network managers to group users logically rather than by physical location. The high performance of the underlying Cisco 4700, combined with the feature-rich NP-1FE, makes it an ideal combination for a low-density, higher-performance application such as inter-VLAN routing.

- High-speed interconnections—The Fast Ethernet interface enables network managers to implement Fast-Ethernet routing solutions for optimal cost and performance across a wide range of applications, including campus or enterprise backbones and data centers. It is also a low-cost way to provide Fast-Ethernet access to traditional low-speed WAN services.
- Local area network aggregation—The Cisco 4500 or the Cisco 4700 can support as many as 12 Ethernet, 4 Token Ring, or 1 FDDI segment. ISDN interfaces are also supported.

With the Catalyst 3000 or Catalyst 5000 system, the Fast Ethernet processor can be used to aggregate up to twelve 10-Mbps LANs and give them high-speed access to such Layer 3 routing services as providing firewalls and maintaining access lists.

Refer to the *Cisco Product Catalog* for specific platform and hardware compatibility information.

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 or access server.

Use the **show interface fastethernet** command to display interface statistics, and use the **show controller fastethernet** to display the information about the Fast Ethernet controller chip. The output shows statistics, including information about initialization block information, transmit ring, receive ring and errors.

## Ethernet and Fast Ethernet Interface Configuration Task List

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

- Specify an Ethernet or Fast Ethernet Interface
- Specify an Ethernet Encapsulation Method
- Specify the Media and Connector Type (Cisco 4000)
- Extend the 10BaseT Capability (Cisco 4000 and Cisco 4500 only) (Does not apply to the FastEthernet interface)
- Configure the 100VG-AnyLAN Port Adapter

## Specify an Ethernet or Fast 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.	<b>interface ethernet</b> <i>number</i>
Begin interface configuration for the Cisco 7200 and 7500 series.	<b>interface ethernet</b> <i>slotport</i>
Begin interface configuration for Cisco 7500 series.	<b>interface ethernet</b> <i>slotport-adapter/port</i>
Begin interface configuration for the Cisco 4000 series with a Fast Ethernet NIM installed.	<b>interface fastethernet</b> <i>number</i>
Specify a Fast Ethernet interface and enter interface configuration mode on the Cisco 7200 series or the Cisco 7500 series.	<b>interface fastethernet</b> <i>slotport</i>
Specify a Fast Ethernet interface and enter interface configuration mode on the Cisco 7500.	<b>interface fastethernet</b> <i>slotport-adapter/port</i>

Use the **show interfaces fastethernet** command to display the Fast Ethernet slots and ports. The Fast Ethernet NIM and the FEIP default to half-duplex mode.

## Specify an Ethernet Encapsulation Method

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 routing protocol you are using, the type of Ethernet media connected to the router or access server and the routing or bridging application you configure.

Establish Ethernet encapsulation of IP packets by performing one of the following tasks in interface configuration mode:

Task	Command
Select ARPA Ethernet encapsulation.	<b>encapsulation arpa</b>
Select SAP Ethernet encapsulation.	<b>encapsulation sap</b>
Select SNAP Ethernet encapsulation.	<b>encapsulation snap</b>

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

## Specify the Media and Connector Type (Cisco 4000)

You can specify that the Ethernet network interface module (NIM) on the Cisco 4000 uses either the default of an AUI and a 15-pin connector, or 10BaseT and an RJ45 connector. To do so, perform one of the following tasks in interface configuration mode:

Task	Command
Select a 15-pin Ethernet connector.	<b>media-type aui</b>
Select an RJ45 Ethernet connector.	<b>media-type 10baset</b>

## Extend the 10BaseT Capability (Cisco 4000 and Cisco 4500 only)

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.

## Configure a Fiber Distributed Data Interface (FDDI)

---

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.	<b>squelch reduced</b>
Return squelch to normal.	<b>squelch normal</b>

## Configure the 100VG-AnyLAN Port Adapter

The 100VG-AnyLAN port adapter (PA-100VG) is available on Cisco 7200 series routers and on Cisco 7500 series routers.

The PA-100VG provides a single interface compatible with and specified by IEEE 802.12 to support 100 Mbps over Category 3 or Category 5 unshielded twisted-pair (UTP) cable with RJ-45 terminators. The PA-100VG supports 802.3 Ethernet packets and can be monitored with the IEEE 802.12 Interface MIB.

To configure the PA-100VG port adapter, perform the following tasks beginning in global configuration mode:

Task	Command
<b>Step 1</b> Specify a 100VG-AnyLAN interface and enter interface configuration.	<b>interface <i>vg-anylan slot/port-adapter/port</i></b> (Cisco 7500)
	<b>interface <i>vg-anylan slot/port</i></b> (Cisco 7200)
<b>Step 2</b> Specify the IP address and subnet mask to the interface.	<b>ip address <i>ip-address mask</i></b>
<b>Step 3</b> Configure the frame type. Currently, only Ethernet frames are supported. The frame type defaults to Ethernet.	<b>frame-type ethernet</b>

---

**Note** The port number for the 100VG-AnyLAN port adapter is always 0.

---

Configuring the PA-100VG interface is similar to configuring an Ethernet or Fast Ethernet interface. To display information about the 100VG-AnyLAN port adapter, use the **show interfaces *vg-anylan* EXEC** command.

## 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. FDDI is not supported on access servers.

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).

Refer to the *Cisco Product Catalog* for specific information on platform and interface compatibility. For installation and configuration information, refer to the installation and configuration publication for the appropriate interface card or port adapter.

## Source-Route Bridging over FDDI on Cisco 4000-M, Cisco 4500-M, and Cisco 4700-M Routers

Source-route bridging (SRB) is supported on the FDDI interface to the Cisco 4000-M, Cisco 4500-M, and Cisco 4700-M routers. For instructions on configuring autonomous FDDI SRB or fast-switching SRB over FDDI, refer to the “Configuring Source-Route Bridging” chapter of the *Bridging and IBM Networking Configuration Guide*.

## Particle-Based Switching of Source-Route Bridge Packets on Cisco 7200 Series Routers

Source-route bridging (SRB) is supported over Fiber Distributed Data Interface (FDDI).

Particle-based switching is supported for SRB packets (over FDDI and Token Ring) by default.

Particle-based switching adds scatter-gather capability to SRB to improve performance. Particles represent a communications data packet as a collection of noncontiguous buffers. The traditional Cisco IOS packet has a packet type control structure and a single contiguous data buffer. A particle packet has the same packet type control structure, but also maintains a queue of particle type structures, each of which manages its own block.

The scatter-gather architecture used by particle-based switching provides the following advantages:

- Allows drivers to use memory more efficiently (especially when using media that has a large maximum transmission unit [MTU]). For example, Token Ring buffers could be 512 bytes rather than 16 KB.
- Allows concurrent use of the same region of memory. For example, on IP multicast a single packet is received and sent out on multiple interfaces simultaneously.
- Allows insertion or deletion of memory at any location in a packet (not just at the beginning or end).

For information about configuring SRB over FDDI, refer to the “Configure Source-Route Bridging” chapter of the Cisco IOS *Bridging and IBM Networking Configuration Guide*.

## 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 Cisco 7500 only.)

---

The CFM state is through 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 Configuration 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
- Enable Full-Duplex Mode on the FDDI
- 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	<b>interface fddi</b> <i>number</i>
Begin interface configuration for the Cisco 7200 or Cisco 7500 series.	<b>interface fddi</b> <i>slot/port</i>

## Enable FDDI Bridging Encapsulation

Cisco 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 FIP.

FIP fully supports transparent and translational bridging for the following configurations:

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

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

Task	Command
Specify the encapsulation method for the FIP.	<b>fddi encapsulate</b>

When you are doing translational bridging, you have to route routable protocols and use translational bridging for the rest (such as LAT).

---

**Note** Bridging between dissimilar media presents several problems that can prevent communications. These problems include bit-order translation (using 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 in networks that bridge between Token Rings and Ethernet networks or between Token Rings and FDDI because of the different ways 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.

### Enable Full-Duplex Mode on the FDDI

To enable full-duplex mode on the PA-F/FD-SM and PA-F/FD-MM port adapters, perform the following task in interface configuration mode:

Task	Command
Enable full-duplex on the FDDI interface of the PA-F/FD-SM and PA-F/FD-MM port adapter.	<b>full-duplex</b> or <b>no half-duplex</b>

### 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.	<b>fdi token-rotation-time</b> <i>microseconds</i>

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.	<b>fdi valid-transmission-time</b> <i>microseconds</i>

## 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.	<b>fddi tl-min-time</b> <i>microseconds</i>

## 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.	<b>fddi c-min</b> <i>microseconds</i>

## 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.	<b>fddi tb-min</b> <i>milliseconds</i>

## 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.	<b>fddi t-out</b> <i>milliseconds</i>

## 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.	<b>no fddi smt-frames</b>
Enable SMT frame processing.	<b>fddi smt-frames</b>

## 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.	<b>fddi duplicate-address-check</b>

### 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.	<b>fdi cmt-signal-bits</b> <i>signal-bits</i> [ <b>phy-a</b>   <b>phy-b</b> ]

### Control the CMT Microcode

You can control whether the CMT onboard functions are on or off. The FIP provides 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 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 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.	<b>no fddi if-cmt</b>

### 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.	<b>cmt connect</b> [ <i>interface-name</i> [ <b>phy-a</b>   <b>phy-b</b> ]]
Stop CMT processes on FDDI ring.	<b>cmt disconnect</b> [ <i>interface-name</i> [ <b>phy-a</b>   <b>phy-b</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.	<b>smt-queue-threshold</b> <i>number</i>

## 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.	<b>fdi burst-count</b>

## 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
- Enable SNMP Illegal Address Trap

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

### Enable a Hub Port

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

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

### 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.	<b>no auto-polarity</b>

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

Task	Command
Reenable automatic receiver polarity reversal.	<b>auto-polarity</b>

## 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 reenbled 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.	<b>no link-test</b>

To reenble 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.	<b>link-test</b>

## 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 any packet from port 3 with your workstation's MAC address. Any packets arriving with a different MAC address cause the port to be disabled. The port is reenbled 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.	<b>source-address</b> <i>[mac-address]</i>

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 “Hub Configuration Examples.”

## Enable SNMP Illegal Address Trap

To enable the router to issue an SNMP trap when an illegal MAC address is detected on an Ethernet hub port, perform the following tasks in hub configuration mode:

Task	Command
<b>Step 1</b> Specify the hub number and the hub port (or range of hub ports) and enter hub configuration mode.	<b>hub ethernet</b> <i>number port [end-port]</i>
<b>Step 2</b> Enable the router to issue an SNMP trap when an illegal MAC address is detected on the hub port.	<b>snmp trap illegal-address</b>

You may need to set up a host receiver for this trap type (snmp-server host) for a Network Management System (NMS) to receive this trap type. The default is no trap. For an example of configuring a SNMP trap for an Ethernet hub port, see the section “Hub Configuration Examples” at the end of this chapter.

## Configure a LAN Extender Interface

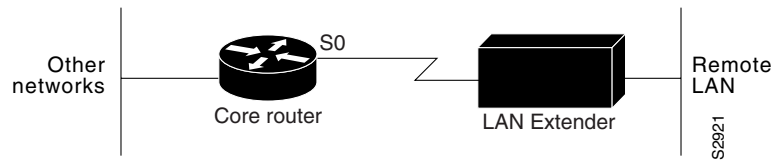
The Cisco 1001 and Cisco 1002 LAN Extenders are two-port chassis that connect a remote Ethernet LAN to a core router at a central site (see Figure 19). The LAN Extender is intended for small networks at remote sites. Overview information for LAN extender interfaces is provided in these sections:

- Connect a LAN Extender to a Core Router
- Install a LAN Extender at a Remote Site
- Discover the MAC Address
- Upgrade Software for the LAN Extender
- Configure the LAN Extender

### Connect a LAN Extender to a Core Router

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 7500 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 19 Cisco 1000 Series LAN Extender Connection to a Core Router**

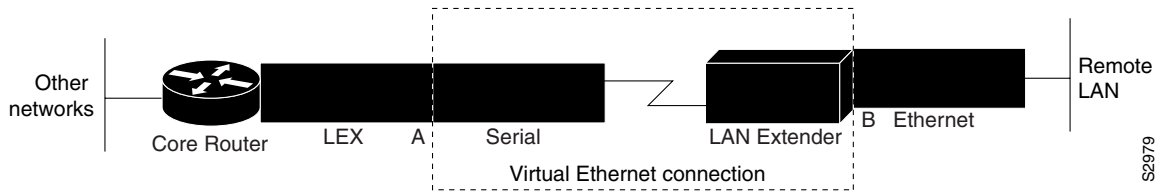


**Expanded View of the Connection to a Core Router**

Figure 20 is an expanded view of Figure 19 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 20 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 20 Expanded View of Cisco 1000 Series LAN Extender Connection**



**Management of the LAN Extender Interface**

Although there is a physical connection between the core router and the LAN Extender, what you actually manage is a remote Ethernet LAN. Figure 21 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 20 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 21.

**Figure 21 LAN Extender Interface Connected to an Ethernet Network**



## Install a LAN Extender at a Remote Site

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

## Discover the MAC Address

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.

## Upgrade Software for the LAN Extender

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

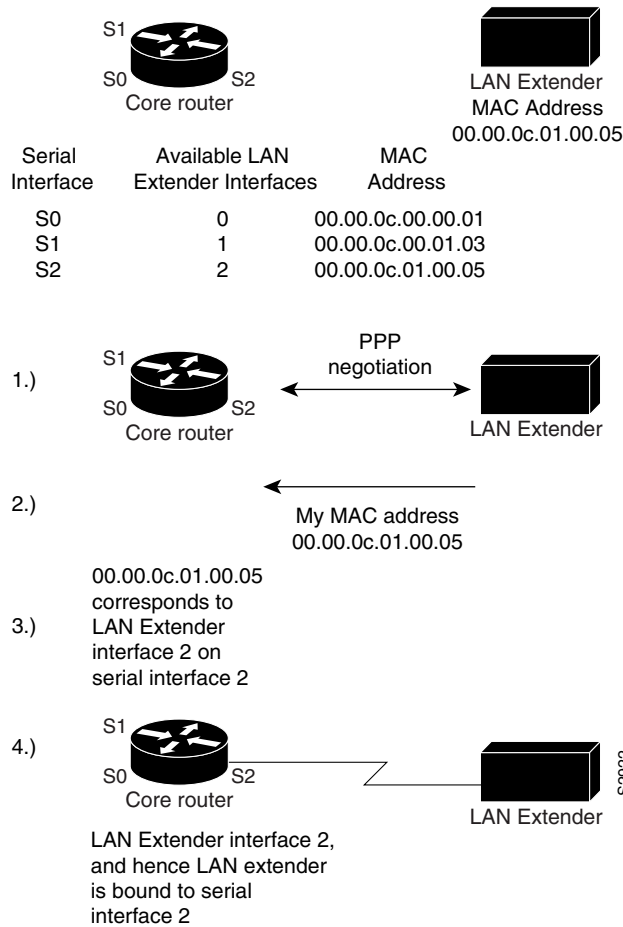
The LAN Extender and core router communicate using the Point-to-Point Protocol (PPP). Before you can configure the LAN Extender from the core router, you must first enable PPP encapsulation on the serial interface to which the LAN Extender is connected.

## Configure the LAN Extender

You 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 22 illustrates this binding process.

**Figure 22 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” in the “Overview of Interface Configuration” chapter. For configuration examples, see the “Enable a LAN Extender Interface Example” and the “LAN Extender Interface Access List Examples” sections at the end of this chapter.

## 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
<b>Step 1</b> Configure a LAN Extender interface in global configuration mode and enter interface configuration mode. or Configure a LAN Extender on a Cisco 7000.	<b>interface</b> <i>lex number</i>  <b>interface</b> <i>lex slot/port</i>
<b>Step 2</b> Assign the burned-in MAC address from your LAN Extender to the LAN Extender interface.	<b>lex burned-in-address</b> <i>ieee-address</i>
<b>Step 3</b> Assign a protocol address to the LAN Extender interface.	<b>ip address</b> <i>ip-address mask</i>
<b>Step 4</b> Return to global configuration mode.	<b>exit</b>
<b>Step 5</b> Configure a serial interface in global configuration mode and enter interface configuration mode.	<b>interface</b> <i>serial number</i>
<b>Step 6</b> Enable PPP encapsulation on the serial interface in interface configuration mode.	<b>encapsulation</b> <i>ppp</i>
<b>Step 7</b> Exit interface configuration mode.	<b>Ctrl-Z</b>
<b>Step 8</b> Save the configuration to memory.	<b>copy</b> <i>running-config startup-config</i>

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.

---

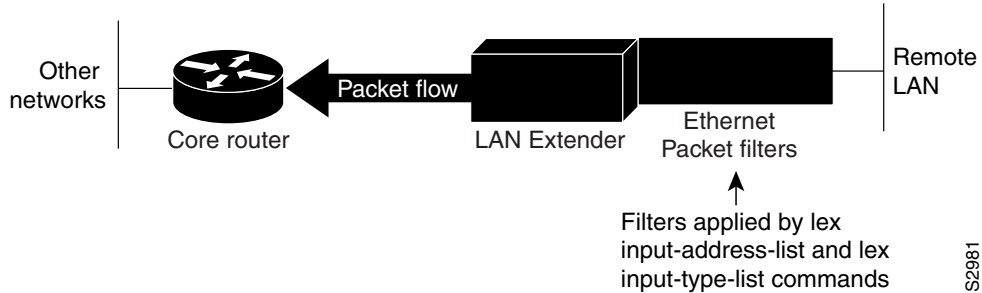
## Define Packet Filters

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 23.) 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

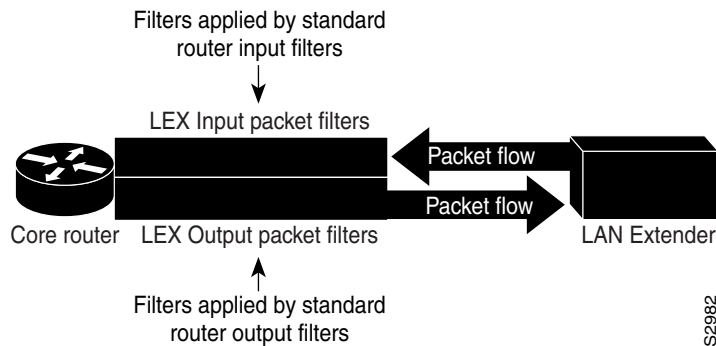
also saves bandwidth on the WAN, because only the desired packets are forwarded from the LAN Extender to the core router. Whenever possible, you should perform packet filtering on the LAN Extender.

**Figure 23 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 24.) 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 24 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 not 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.	<b>access-list</b> <i>access-list-number</i> { <b>permit</b>   <b>deny</b> } <i>address</i> <i>mask</i>

**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.	<b>lex input-address-list</b> <i>access-list-number</i>

For an example of creating an access list and applying it to a LAN Extender interface, see the section “LAN Extender Interface Access List Examples” in the section “LAN 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 *Bridging and IBM Networking Command Reference*. 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.	<b>access-list</b> <i>access-list-number</i> { <b>permit</b>   <b>deny</b> } <i>type-code wild-mask</i>

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.	<b>ip input-type-list</b> <i>access-list-number</i>

For an example of creating an access list and applying it to a LAN Extender interface, see the section “LAN Extender Interface Access List Examples” in the section “LAN 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 System Performance” 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.	<b>priority-list</b> <i>list</i> <b>protocol</b> <i>protocol</i> { <b>high</b>   <b>medium</b>   <b>normal</b>   <b>low</b> } or <b>priority-list</b> <i>list</i> <b>protocol</b> <b>bridge</b> { <b>high</b>   <b>medium</b>   <b>normal</b>   <b>low</b> } <b>list</b> <i>list-number</i>

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.	<b>lex priority-group</b> <i>group</i>

## 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.	<b>lex timeout</b> <i>milliseconds</i>

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.	<b>lex retry-count</b> <i>number</i>

## 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.

## Configure a LAN Extender Interface

---

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.	<b>shutdown</b>

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.	<b>no shutdown</b>

## 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.	<b>clear controller lex number [prom]</b>
Halt operation of the LAN Extender on a Cisco 7000.	<b>clear controller lex slot/port [prom]</b>

## 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 a TFTP server or from Flash memory on the core router 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.	<b>copy tftp lex number</b>
Download a software image from Flash memory.	<b>copy flash lex number</b>

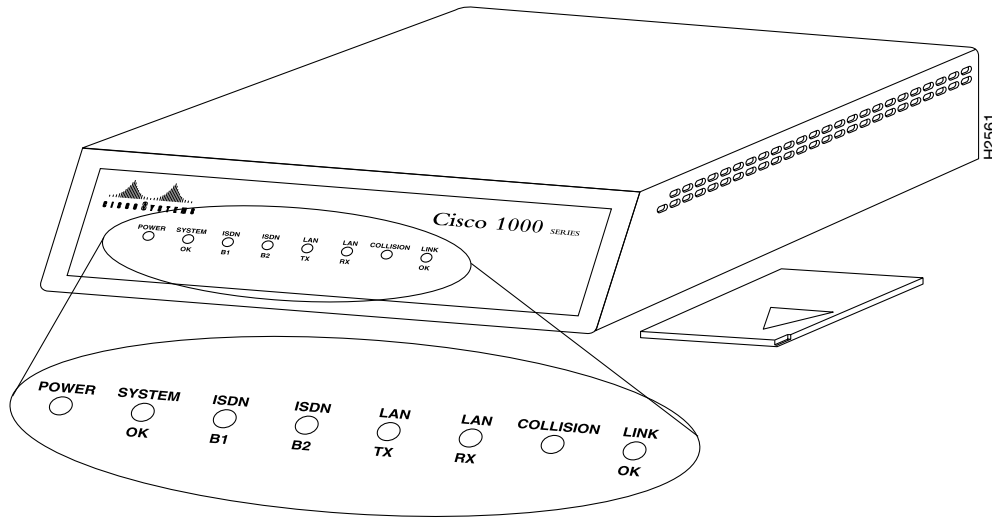
## Troubleshoot the LAN Extender

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 25).

Figure 25 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. explains the LEDs.

Table 10 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.

**Table 10 LED Trouble Indicators (Continued)**

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 serial link is up and functioning.

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

## Configure a Token Ring Interface

Cisco supports various Token Ring interfaces. Refer to the *Cisco Product Catalog* for information about platform and hardware compatibility.

The Token Ring interface supports both routing (Layer 3 switching) and source-route bridging (Layer 2 switching). Routing and bridging function on a per-protocol basis. For example, IP traffic could be routed while SNA traffic is bridged. Routing features enhance source-route bridges.

The Token Ring MIB variables support the specification 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 that Cisco IOS software supports.

---

**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 drastically affect the stability of routing tables. Once you have again plugged the cable into the MAU, restart the interface by entering the **clear interface tokenring** command, where the *number* argument is the interface number.

---

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

## Particle-Based Switching of Source-Route Bridge Packets on Cisco 7200 Series Routers

Particle-based switching is supported for SRB packets (over FDDI and Token Ring) by default.

Particle-based switching adds scatter-gather capability to SRB to improve performance. Particles represent a communications data packet as a collection of noncontiguous buffers. The traditional Cisco IOS packet has a packet type control structure and a single contiguous data buffer. A particle packet has the same packet type control structure, but it also maintains a queue of particle type structures, each of which manages its own block.

The scatter-gather architecture used by particle-based switching provides the following advantages:

- Allows drivers to use memory more efficiently (especially when using media that has a large maximum transmission unit [MTU]). For example, Token Ring buffers could be 512 bytes rather than 16 KB.
- Allows concurrent use of the same region of memory. For example, on IP multicast a single packet is received and sent out on multiple interfaces simultaneously.
- Allows insertion or deletion of memory at any location in a packet (not just at the beginning or end).

For information about configuring SRB over FDDI, refer to the “Configure Source-Route Bridging” chapter of the *Bridging and IBM Networking Configuration Guide*.

## Token Ring Interface Configuration Task List

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

- Specify a Token Ring Interface
- 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.	<b>interface tokenring</b> <i>number</i>
Begin interface configuration for the Cisco 7200 or Cisco 7500 series.	<b>interface tokenring</b> <i>slot/port</i>
Begin interface configuration for the Cisco 7500 series.	<b>interface tokenring</b> <i>slot/port-adapter/port</i>

### Enable Early Token Release

Cisco 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.	<b>early-token-release</b>

### 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.	<b>local-lnm</b>

## LAN Interface Configuration Examples

This section provides examples to illustrate configuration tasks described in this chapter. These examples are included:

- Enable Interface Configuration Examples
- Enable Ethernet Encapsulation Example
- PA-VG100 Port Adapter Configuration Example
- Hub Configuration Examples
- Enable a LAN Extender Interface Example
- LAN Extender Interface Access List Examples
- Packet OC-3 Interface Configuration Examples

### Enable Interface Configuration Examples

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
```

### Configure Specific IP Addresses for an Interface Example

This example shows how to configure the access server so that it will use the default address pool on all interfaces except interface 7, on which it will use an address pool called lass:

```
ip address-pool local
ip local-pool lass 172.30.0.1
async interface
```

```
interface 7
peer default ip address lass
```

## Enable Ethernet Encapsulation Example

These commands enable standard Ethernet Version 2.0 encapsulation on the Ethernet interface processor in slot 4 on port 2 of a Cisco 7500:

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

## PA-VG100 Port Adapter Configuration Example

Following is an example of a basic configuration for the PA-VG100 port adapter interface in slot 1 on a Cisco 7500 series router. In this example, IP routing is enabled on the router, so an IP address and subnet mask are assigned to the interface.

```
configure terminal
interface vg-anylan 1/0/0
ip address 1.1.1.10 255.255.255.0
no shutdown
exit
exit
```

## Hub Configuration Examples

The following sections provide examples of hub configuration:

- Hub Port Startup Examples
- Source Address for an Ethernet Hub Port Configuration Examples
- Hub Port Shutdown Examples
- Enable SNMP Illegal Address Trap for Hub Port Example

### Hub Port Startup Examples

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
```

### Source Address for an Ethernet Hub Port Configuration Examples

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
```

### Hub Port Shutdown Examples

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
```

### Enable SNMP Illegal Address Trap for Hub Port Example

The following example specifies the gateway IP address, enables an SNMP trap to be issued to the host 172.69.40.51 when a MAC address violation is detected on hub ports 2, 3, or 4, and specifies that interface Ethernet 0 is the source for all traps on the router. The community string is defined as the string *public* and the read/write parameter is set.

```
ip route 0.0.0.0 0.0.0.0 172.22.10.1
snmp-server community public rw
snmp-server trap-source ethernet 0
snmp-server host 172.69.40.51 public
hub ethernet 0 2 4
snmp trap illegal-address
```

### Enable a LAN Extender Interface Example

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
```

### LAN Extender Interface Access List Examples

This section provides these examples of LAN extender interface configuration:

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

#### Filtering by MAC Address Example

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.

### Filtering by Ethernet Type Code Example

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.

## Packet OC-3 Interface Configuration Examples

The examples in this section include a simple configuration and a configuration for two routers back to back.

### Packet OC-3 Configuration with Default Values Accepted

In the following example, the default framing, MTU, and clock source are accepted, and the interface is configured for the IP protocol:

```
interface posi 3/0
ip address 172.18.2.3 255.0.0.0
```

### Two Routers Connected Back to Back

To connect two routers, attach the cable between the Packet OC-3 port on each. By default, the POSIP uses loop timing mode. For back-to-back operation, only one of the POSIPs may be configured to supply its internal clock to the line.

In the following example, two routers are connected back to back through their Packet OC-3 interfaces:

#### First router

```
interface posi 3/0
ip address 170.1.2.3 255.0.0.0
no keepalive
pos internal-clock
```

#### Second router

```
interface posi 3/0
```

```
ip address 170.1.2.4 255.0.0.0  
no keepalive
```

The following example shuts down the entire T1 line physically connected to a Cisco 7500:

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