Configuring Serial Interfaces

Use the information in this chapter to configure serial interfaces.

For information on configuring an Asynchronous Transfer Mode (ATM) interface, see the "Configuring ATM Access over a Serial Interface" chapter in the *Wide-Area Networking Configuration Guide*.

See also the section "Invoke ATM over a Serial Line" in the section "Configure a Synchronous Serial Interface" in 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 serial 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.

These sections are included in this chapter:

- Configure a High-Speed Serial Interface (HSSI)
- Configure a Synchronous Serial Interface
- Configure a Packet OC-3 Interface
- Configure Serial Interfaces for CSU/DSU Service Modules
- Configure Low-Speed Serial Interfaces

For examples of conguration tasks shown in this chapter, see "Serial Interface Configuration Examples" at the end of this chapter.

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

Configure a High-Speed Serial Interface (HSSI)

The HSSI Interface Processor (HIP) provides a single HSSI network interface. The network interface resides on a modular interface processor that provides a direct connection between the high-speed CiscoBus and an external network.

HSSI Configuration Task List

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

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

Specify a HSSI

To specify a 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 7500 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 "Configure SLIP and PPP" chapter of the *Access Services Configuration Guide* and the "Configure PPP for Wide-Area Networking" chapter of the *Wide-Area Networking Configuration Guide*.

Invoke ATM on a 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.	**dxi map** *protocol address vpi vci* [**broadcast**]

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

To configure an ATM interface using an AIP card, see the “Configuring ATM” chapter in the *Wide-Area Networking Configuration Guide*.

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 Synchronous Serial Interface

Synchronous serial interfaces are supported on various serial network interface cards or systems. These interfaces support full-duplex operation at T1 (1.544 Mbps) and E1 (2.048 Mbps) speeds. Refer to the *Cisco Product Catalog* for specific information regarding platform and hardware compatibility.

Synchronous Serial Configuration 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 a Channelized T3 Interface Processor
- Configure PPP
- Configure Half-Duplex and Bisync for Synchronous Serial Port Adapters on Cisco 7200 Series Routers
- Configure Compression Service Adapters on Cisco 7500 Series Routers
- Configure Compression of HDLC Data
- Configure Real-Time Transport Protocol Header Compression
- Invoke ATM over a Serial Line
- Configure the CRC
- Use the NRZI Line-Coding Format
- Enable the Internal Clock
- Set Transmit Delay
- Configure DTR Signal Pulsing
- Ignore DCD and Monitor DSR as Line Up/Down Indicator
- Specify the Serial Network Interface Module Timing
- Specify G.703 Interface Options

See the “Serial Interface Configuration Examples” section at the end of this chapter for examples of configuration tasks described in this chapter.

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 7200 series.	**interface serial** *slot***/***port*
Begin interface configuration for the Cisco 7500 series.	**interface serial** *slot***/***port-adapter***/***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 in the Cisco IOS software. ATM-DXI is described in this chapter in the section “Invoke ATM over a Serial Line.” PPP is described in the “Configure PPP for Wide-Area Networking” chapter. The remaining encapsulation methods are defined in their respective books and chapters describing the protocols or applications. Serial encapsulation methods are also discussed in the *Configuration Fundamentals Command Reference* in the chapter “Interface Commands” under the **encapsulation** command.

By default, synchronous interfaces operate in full-duplex mode. To configure an SDLC interface for half-duplex mode, perform the following task in interface configuration mode:

Task	Command
Configure an SDLC interface for half-duplex mode.	**half-duplex**

BSC is a half-duplex protocol. Each block of transmission is acknowledged explicitly. To avoid the problem associated with simultaneous transmission, there is an implicit role of primary and secondary station. The primary resends the last block if there is no response from the secondary within the period of block receive timeout.

To configure the serial interface for full-duplex mode, perform the following task in interface configuration mode:

Task	Command
Specify that the interface can run BSC using switched RTS signals.	**full-duplex**

Configure a Channelized T3 Interface Processor

The CT3IP is a fixed-configuration interface processor based on the second-generation Versatile Interface Processor (VIP2). It is supported on the Cisco 7500 series routers. The CT3IP has four T1 connections via DB-15 connectors and one DS3 connection via BNC connectors. Each DS3 interface can provide up to 28 T1 channels (a single T3 group). Each channel is presented to the system as a serial interface that can be configured individually. The CT3IP can transmit and receive data bidirectionally at the T1 rate of 1.536 Mbps. The four T1 connections use 100-ohm twisted-pair serial cables to external channel service units (CSUs) or to a MultiChannel Interface Processor (MIP) on the same router or on another router. For wide-area networking, the CT3IP can function as a concentrator for a remote site.

The CT3IP provides 28 T1 channels for serial transmission of data. Each T1 channel can be configured to use a portion of the T1 bandwidth or the entire T1 bandwidth for data transmission. Bandwidth for each T1 channel can be configured for *n* x 56 kbps or *n* x 64 kbps (where *n* is 1 to 24). The unused portion of the T1 bandwidth, when not running at full T1 speeds, is filled with idle channel data. The CT3IP does not support the aggregation of multiple T1 channels (called *inverse muxing* or *bonding*) for higher bandwidth data rates.

The first three T1 channels of the CT3IP can be broken out to the three DSUP-15 connectors on the CPT3IP so the T1 can be further demultiplexed by the MIP on the same router or on another router, or by other multiplexing equipment. When connecting to the MIP, you configure a channelized T1 as described in the "Configure External T1 Channels" section. This is referred to as an external T1 channel.

The CT3IP supports RFC 1406 and RFC 1407 (CISCO-RFC-1407-CAPABILITY.my). For information Cisco MIBs, refer to the current Cisco IOS release note for the location of the Management Information Base (MIB) online reference.

For RFC 1406, Cisco supports all tables except the "Frac" table. For RFC 1407, Cisco supports all tables except the "FarEnd" tables.

The CT3IP supports the following WAN protocols:

- Frame Relay
- HDLC

- PPP
- SMDS Data Exchange Interface (DXI)

The CT3IP meets ANSI T1.102-1987 and BELCORE TR-TSY-000499 specifications for T3 and meets ANSI 62411 and BELCORE TR499 specifications for T1. The CT3IP provides internal CSU functionality and includes reporting performance data statistics, transmit and receive statistics, and error statistics. The CT3IP supports RFC 1406 (T1 MIB) and RFC 1407 (T3 MIB).

External T1 channels do not provide CSU functionality and must connect to an external CSU.

The CT3IP supports RFC 1406 (T1 MIB) and RFC 1407 (T3 MIB).

Channelized T3 Configuration Task List

Perform the tasks in the following sections to configure the CT3IP (all tasks are optional except for the second task):

- Configure the T3 Controller
- Configure Each T1 Channel
- Configure External T1 Channels
- Troubleshoot the T3 and T1 Channels
- Loopback T1 Channels
- Loopback T3 Lines
- Monitor and Maintain the CT3IP
- Enable Performance Report Monitoring
- Enable BERT Test Pattern
- Enable Remote FDL Loopbacksks

After you configure the T1 channels on the CT3IP, you can continue configuring it as you would a normal serial interface. All serial interface commands might not be applicable to the T1 channel. For more information, see the "Configure a Synchronous Serial Interface" section earlier in this chapter.

For CT3IP configuration examples, see the "Channelized T3 Interface Processor Configuration Examples" section, later in this chapter.

Configure the T3 Controller

If you do not modify the configuration of the CT3IP, the configuration defaults shown in Table 11 are used.

Table 11 CT3IP Controller Defaults

Attribute	Default Value
Framing	auto-detect
Cable length	224 feet
Clock source	internal

If you must change any of the default configuration attributes, complete the first task in global configuration mode followed by any of the optional tasks in controller configuration mode:

Task	Command
Select the CT3IP and enter controller configuration mode.	**controller t3** *slot*/*port-adapter*/*port*
Change the framing format.	**framing** {**c-bit** | **m23** | **auto-detect**}
Change the cable length (values are 0 to 450 feet).	**cablelength** *feet*
Change the clock source used by the T3 controller.	**clock source** {**internal** | **line**}

Note The port adapter and port numbers for the CT3IP are 0.

Note Although you can specify a cable length from 0 to 450 feet, the hardware only recognizes two ranges: 0 to 224 and 225 to 450. For example, entering 150 feet uses the 0 to 224 range. If you later change the cable length to 200 feet, there is no change because 200 is within the 0 to 224 range. However, if you change the cable length to 250, the 225 to 450 range is used. The actual number you enter is stored in the configuration file.

Configure Each T1 Channel

You must configure the timeslots used by each T1 channel on the CT3IP. Optionally, you can specify the speed, framing format, and clock source used by each T1 channel. If you do not specify the speed, framing format, and clock source used by each T1 channel, the configuration defaults shown in Table 12 are used.

Table 12 CT3IP T1 Channel Defaults

Attribute	Default Value
Speed	64 kbps
Framing	esf
Clock source	internal
Linecode	b8zs
T1 yellow alarm	detection and generation

To specify the timeslots used by each T1 channel, complete the following tasks beginning in global configuration mode:

Task		Command
Step 1	Select the CT3IP and enter controller configuration mode.	**controller t3** *slot*/*port-adapter*/*port*
Step 2	Configure the timeslots (values are 1 to 24) for the T1 channel (values are 1 to 28) and optionally specify the speed for each T1 channel.	**t1** *channel* **timeslot** *range* [**speed** {**56** | **64**}]

Note The 56-kbps speed is valid only for T1 channels 21 through 28.

Note T1 channels on the CT3IP are numbered 1 to 28 rather than the more traditional zero-based scheme (0 to 27) used with other Cisco products. This is to ensure consistency with telco numbering schemes for T1 channels within channelized T3 equipment.

If you need to change any of the default configuration attributes, complete the first task in global configuration mode followed by any of the optional tasks in controller configuration mode:

Task	Command
Select the CT3IP and enter controller configuration mode.	**controller t3** *slot***/***port-adapter***/***port*
Change the framing format used by the T1 channel (values are 1 to 28).	**t1** *channel* **framing** {**esf** | **sf**}
Change the clock source used by the T1 channel (values are 1 to 28).	**t1** *channel* **clock source** {**internal** | **line**}
Change the line coding used by the T1 channel (values are 1 to 28).	**t1** *channel* **linecode** {**ami** | **b8zs**}
Disable detection or generation of a yellow alarm on the T1 channel (values are 1 to 28).	**no t1** *channel* **yellow** {**detection** | **generation**}

Note If you select **ami** line coding, you must also invert the data on the T1 channel by using the **invert data** interface command. To do so, first use the **interface serial** *slot/port-adapter/port:t1-channel* global configuration command to select the T1 channel and enter interface configuration mode.

Note If you select **sf** framing, you should consider disabling yellow alarm detection because the yellow alarm can be incorrectly detected with **sf** framing.

After you configure the T1 channels on the CT3IP, you can continue configuring it as you would a normal serial interface. All serial interface commands might not be applicable to the T1 channel. For more information, refer to the "Configure a Synchronous Serial Interface" in this chapter.

To enter interface configuration mode and configure the serial interface that corresponds to a T1 channel, perform the following task in global configuration mode:

Task	Command
Define the serial interface for a T1 channel (values are 1 to 28) and enter interface configuration mode.	**interface serial** *slot***/***port-adapter***/***port***:***t1-channel*

Note The port adapter and port numbers for the CT3IP are 0.

In addition to the commands in the "Configure a Synchronous Serial Interface" section, the **invert data** interface command can be used to configure the T1 channels on the CT3IP. If the T1 channel on the CT3IP is using AMI line coding, you must invert the data. For information on the **invert data** interface command, refer to "Invert the Data" later in this chapter. For more information, see the **t1 linecode** controller command.

Configure External T1 Channels

The first three T1 channels (1, 2, and 3) of the CT3IP can be broken out to the DSUP-15 connectors on the CPT3IP so the T1 channel can be further demultiplexed by the MIP on the same router, another router, or other multiplexing equipment.

Note If a T1 channel that was previously configured as a serial interface is broken out to the external T1 port, that interface and its associated configuration remain intact while the channel is broken out to the external T1 port. The serial interface is not usable during the time the T1 channel is broken out to the external T1 port; however, the configuration remains to facilitate the return of the T1 channel to a serial interface with the **no t1 external** command.

To configure a T1 channel as an external port, complete the following tasks beginning in EXEC mode:

Task		Command
Step 1	Determine if the external device connected to the external T1 port is configured and cabled correctly by locating the line `Ext1...` in the display output. If the line status is `OK,` a valid signal is being received and the signal is not an all-ones signal.	**show controller t3** *slot***/***port-adapter***/***port*
Step 2	Enter configuration mode.	**configure terminal**
Step 3	Select the CT3IP and enter controller configuration mode.	**controller t3** *slot***/***port-adapter***/***port*
Step 4	Configure the T1 channel (values are 1, 2, and 3) as an external port and optionally specify the cable length and line code. The default cable length is 133 feet, and the default line code is b8zs.	**t1 external** *channel* [**cablelength** *feet*] [**linecode** {**ami** | **b8zs**}]

Note Only T1 channels 1 through 3 can be configured as an external T1.

Note Although you can specify a cable length from 0 to 655 feet, the hardware only recognizes the following ranges: 0 to 133, 134 to 266, 267 to 399, 400 to 533, and 534 to 655. For example, entering 150 feet uses the 134 to 266 range. If you later change the cable length to 200 feet, there is no change because 200 is within the 134 to 266 range. However, if you change the cable length to 399, the 267 to 399 range is used. The actual number you enter is stored in the configuration file.

After you configure the external T1 channel, you can continue configuring it as a channelized T1 from the MIP. All channelized T1 commands might not be applicable to the T1 interface. To define the T1 controller and enter controller configuration mode, perform the following task in global configuration mode:

Task	Command
Select the MIP and enter controller configuration mode.	**controller t1** *slot***/***port*

After you configure the channelized T1 on the MIP, you can continue configuring it as you would a normal serial interface. All serial interface commands might not be applicable to the T1 interface. To enter interface configuration mode and configure the serial interface that corresponds to a T1 channel group, perform the following task beginning in global configuration mode:

Task	Command
Define the serial interface for a T1 channel on the MIP (values are 1 to 28) and enter interface configuration mode.	**interface serial** *slot***/***port***:***t1-channel*

For more information, refer to the "Configure Channelized T1" section and the "Configure a Synchronous Serial Interface" section in the "Configuring Interfaces" chapter of this guide.

For an example of configuring an external T1 channel, see the "Channelized T3 Interface Processor Configuration Examples" section later in this chapter.

Troubleshoot the T3 and T1 Channels

You can use the following methods to troubleshoot the CT3IP using Cisco IOS software:

- Test the T1 by using the **t1 test** controller configuration command and the test port.
- Loop the T1 by using **loopback** interface configuration commands.
- Loop the T3 by using **loopback** controller configuration commands.

Enable Test Port

You can use the T1 test port available on the CT3IP to break out any of the 28 T1 channels for testing (for example, 24-hour BERT testing as is commonly done by telephone companies before a line is brought into service).

The T1 test port is also available as an external port. For more information on configuring an external port, see the previous section, "Configure External T1 Channels."

Note If a T1 channel that was previously configured as a serial interface is broken out to the T1 port test, that interface and its associated configuration remain intact while the channel is broken out to the T1 port test . The serial interface is not usable during the time the T1 channel is broken out to the T1 test port; however, the configuration remains to facilitate the return of the T1 channel to a serial interface with the **no t1 test** command.

To enable a T1 channel as a test port, complete the following tasks beginning in global configuration mode:

Task		Command
Step 1	Determine if the external device connected to the external T1 port is configured and cabled correctly by locating the line `Ext1...` in the display output. If the line status is `OK,` a valid signal is being received and the signal is not an all-ones signal.	**show controller t3** *slot***/***port-adapter***/***port*
Step 2	Select the CT3IP and enter controller configuration mode.	**controller t3** *slot***/***port-adapter***/***port*
Step 3	Enable the T1 channel (values are 1 to 28) as a test port and optionally specify the cable length and line code. The default cable length is 133 feet, and the default line code is b8zs.	**t1 test** *channel* [**cablelength** *feet*] [**linecode** {**ami** | **b8zs**}]

To disable a T1 channel as a test port, complete the following tasks beginning in global configuration mode:

Task		Command
Step 1	Select the CT3IP and enter controller configuration mode.	**controller t3** *slot***/***port-adapter***/***port*
Step 2	Disable the T1 channel (values are 1 to 28) as a test port.	**no t1 test** *channel*

Note Although you can specify a cable length from 0 to 655 feet, the hardware only recognizes the following ranges: 0 to 133, 134 to 266, 267 to 399, 400 to 533, and 534 to 655. For example, entering 150 feet uses the 134 to 266 range. If you later change the cable length to 200 feet, there is no change because 200 is within the 134 to 266 range. However, if you change the cable length to 399, the 267 to 399 range is used. The actual number you enter is stored in the configuration file.

Loopback T1 Channels

You can perform the following types of loopbacks on a T1 channel:

- Local—Loops the router output data back toward the router at the T1 framer and sends an AIS signal out toward the network (see Figure 26).
- Network line—Loops the data back toward the network before the T1 framer and automatically sets a local loopback (see Figure 27).
- Network payload—Loops just the payload data back toward the network at the T1 framer and automatically sets a local loopback (see Figure 28).
- Remote line inband—Sends a repeating 5-bit inband pattern (00001) to the remote end requesting that it enter into a network line loopback (see Figure 29).

To enable loopbacks on a T1 channel, complete the first task beginning in global configuration mode followed by any one of the following tasks:

Task	Command
Select the T1 channel (values are 1 to 28) on the CT3IP and enter interface configuration mode.	**interface serial** *slot***/***port-adapter***/***port***:***t1-channel*
Enable the local loopback on the T1 channel.	**loopback local**
Enable the network line loopback on the T1 channel.	**loopback network line**
Enable the network payload loopback on the T1 channel.	**loopback network payload**
Enable the remote line inband loopback on the T1 channel.	**loopback remote line inband**

Note The port adapter and port numbers for the CT3IP are 0.

Figure 26 shows an example of a local loopback in which the loopback occurs in the T1 framer.

Figure 26 CT3IP Local Loopback

Figure 27 shows an example of a network line loopback in which just the data is looped back toward the network (before the T1 framer).

Figure 27 CT3IP Network Line Loopback

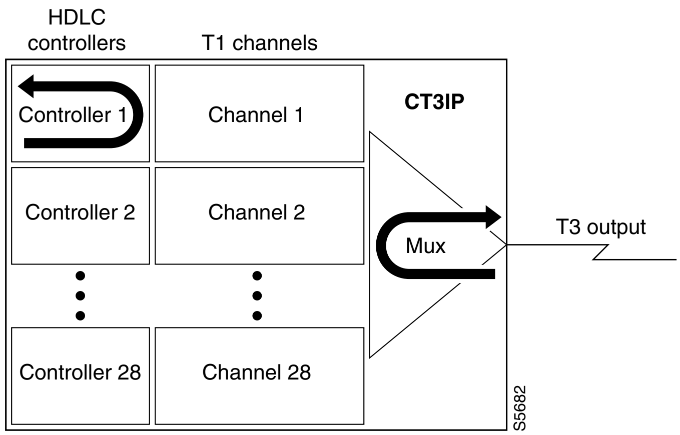


Figure 28 shows an example of a network payload loopback in which just the payload data is looped back toward the network at the T1 framer.

Figure 28 CT3IP Network Payload Loopback

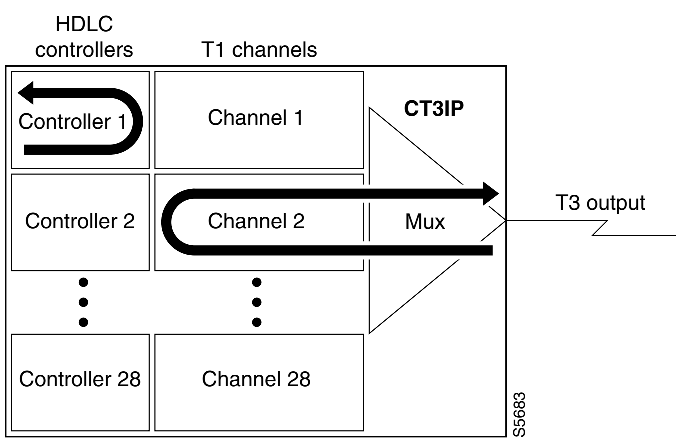
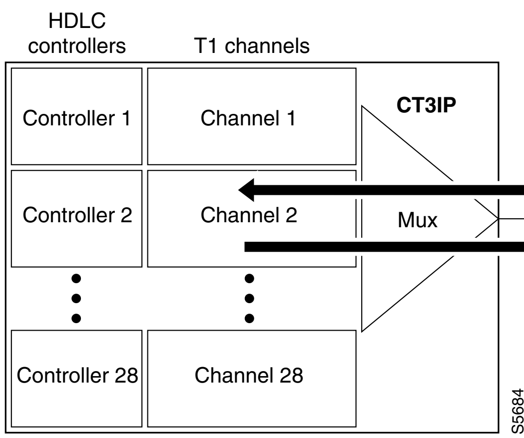


Figure 29 shows an example of a remote inband loopback in which the network line enters a line loopback.

Figure 29 CT3IP Remote Loopback



Loopback T3 Lines

You can put the entire T3 line into loopback mode (that is, all T1 channels are looped) by using the following types of loopbacks:

- local—Loops the router output data back toward the router at the T1 framer and sends an AIS signal out toward the network.
- network—Loops the data back toward the network (before the T1 framer).
- remote —Sends a FEAC (far-end alarm control) request to the remote end requesting that it enter into a network line loopback. FEAC requests (and therefore remote loopbacks) are only possible when the T3 is configured for C-bit framing. The type of framing used is determined by the equipment you are connecting to. (For more information, see the **framing** controller command.)

To enable loopbacks on the T3 (and all T1 channels), complete the first task beginning in global configuration mode followed by any one of the following tasks:

Task	Command
Select the CT3IP and enter controller configuration mode.	**controller t3** *slot***/***port-adapter***/***port*
Enable the local loopback.	**loopback local**
Enable the network loopback.	**loopback network**
Enable the remote loopback.	**loopback remote**

Note The port adapter and port numbers for the CT3IP are 0.

Monitor and Maintain the CT3IP

After configuring the new interface, you can monitor the status and maintain the CT3IP in the Cisco 7000 series routers with an RSP7000 or in the Cisco 7500 series routers by using the **show** commands. To display the status of any interface, complete one of the following tasks in EXEC mode:

Task	Command
Display the internal status of each interface processor and list each interface.	**show controller cbus**
Display the status of the T3 and T1 channels (values are 1 to 28) including the T3 alarms and T1 alarms for all 28 T1 channels or only the T1 channel specified.	**show controller t3** [*slot*/*port-adapter*/*port*[**:***t1-channel*]] [**brief** | **tabular**]
Display statistics about the serial interface for the specified T1 channel (values are 1 to 28) on the router.	**show interfaces serial** *slot*/*port-adapter*/*port***:***t1-channel* [**accounting** | **crb**]

Configure Maintenance Data Link (MDL) Messages

The CT3IP can be configured to send a Maintenance Data Link (MDL) message as defined in the ANSI T1.107a-1990 specification. To specify the transmission of the MDL messages, complete the following tasks beginning in global configuration mode:

Task		Command
Step 1	Select the CT3IP and enter controller configuration mode.	**controller t3** *slot*/*port-adapter*/*port*
Step 2	Configure the Maintenance Data Link (MDL) message.	**mdl** {**transmit** {**path** | **idle-signal** | **test-signal**} | **string** {**eic** | **lic** | **fic** | **unit** | **pfi** | **port** | **generator**} *string*}

Note Specify one **mdl** command for each message. For example, use **mdl string eic Router A** to transmit “Router A” as the equipment identification code and use **mdl string lic Test Network** to transmit “Test Network” as the location identification code.

Use the **show controllers t3** command to display MDL information (received strings). MDL information is displayed only when framing is set to C-bit.

Enable Performance Report Monitoring

The CT3IP supports performance reports via the Facility Data Link (FDL) per ANSI T1.403. By default, performance reports are disabled. To enable FDL performance reports, complete the following tasks beginning in global configuration mode:

Task		Command
Step 1	Select the CT3IP and enter controller configuration mode.	**controller t3** *slot*/*port-adapter*/*port*
Step 2	Enable one-second transmission of the performance report for a specific T1 channel (values are 1 to 28).	**t1** *channel* **fdl ansi**

Note Performance reporting is available only on T1 channels configured for ESF framing.

To display the remote performance report information, complete the following task in EXEC command mode:

Task	Command
Display the remote performance report information for the T1 channel (values are 1 to 28).	**show controller t3** [*slot*/*port-adapter*/*port*[**:***t1-channel*]] **remote performance** [**brief** | **tabular**]

Enable BERT Test Pattern

To enable and disable generation of a bit error rate testing (BERT) test pattern for a specified interval for a specific T1 channel, complete the following tasks beginning in global configuration mode:

Task		Command
Step 1	Select the CT3IP and enter controller configuration mode.	**controller t3** *slot*/*port-adapter*/*port*
Step 2	Enable a BERT test pattern on a T1 channel (values are 1 to 28).	**t1** *channel* **bert pattern** {**0s** | **1s** | **2^15** | **2^20** | **2^23**} **interval** *minutes*
Step 3	Disable a BERT test pattern on a T1 channel (values are 1 to 28).	**no t1** *channel* **bert pattern** {**0s** | **1s** | **2^15** | **2^20** | **2^23**} **interval** *minutes*

The BERT test patterns from the CT3IP are framed test patterns (that is, the test patterns are inserted into the payload of the framed T1 signal).

To view the BERT results, use the **show controller t3** or **show controller t3 brief** EXEC command. The BERT results include the following information:

- Type of test pattern selected
- Status of the test
- Interval selected
- Time remaining on the BERT test
- Total bit errors
- Total bits received

When the T1 channel has a BERT test running, the line state is DOWN. Also, when the BERT test is running and the Status field is Not Sync, the information in the total bit errors field is not valid. When the BERT test is done, the Status field is not relevant.

The **t1 bert pattern** command is not written to NVRAM because it is only used for testing the T1 channel for a short predefined interval and to avoid accidentally saving the command, which could cause the interface not to come up the next time the router reboots.

Enable Remote FDL Loopbacks

You can perform the following types of remote Facility Data Link (FDL) loopbacks on a T1 channel:

- Remote payload FDL ANSI—Sends a repeating, 16-bit ESF data link code word (00010100 11111111) to the remote end requesting that it enter into a network payload loopback.
- Remote line FDL ANSI—Sends a repeating, 16-bit ESF data link code word (00001110 11111111) to the remote end requesting that it enter into a network line loopback.

To enable loopbacks on a T1 channel, complete the first task beginning in global configuration mode followed by Step 2 or Step 3 depending on the type of loopback you want to perform:

Task		Command
Step 1	Select the T1 channel (values are 1 to 28) on the CT3IP and enter interface configuration mode.	**interface serial** *slot***/***port-adapter***/***port***:***t1-channel*
Step 2	Enable the remote payload FDL ANSI bit loopback on the T1 channel.	**loopback remote payload** [**fdl**] [**ansi**]
Step 3	Enable the remote line FDL ANSI bit loopback on the T1 channel.	**loopback remote line** [**fdl**] [**ansi**]

Note The port adapter and port numbers for the CT3IP are 0.

Configure PPP

To configure PPP, see the "Configuring Media-Independent PPP and Multilink PPP" chapter of the *Dial Solutions Configuration Guide*.

Configure Half-Duplex and Bisync for Synchronous Serial Port Adapters on Cisco 7200 Series Routers

The synchronous serial port adapters (PA-8T-V35, PA-8T-X21, PA-8T-232, and PA-4T+) on Cisco 7200 series routers support half-duplex and binary synchronous communications (Bisync). Bisync is a character-oriented data-link layer protocol for half-duplex applications. In half-duplex mode, data is sent one direction at a time. Direction is controlled by handshaking the RST and CTS control lines. These are described in the following sections:

- Configure Bisync
- Configure Half-Duplex Carrier Modes and Timers

For more information about the PA-8T-V35, PA-8T-X21, PA-8T-232, and PA-4T+ synchronous serial port adapters, refer to the following publications:

- *PA-8T-V35 Synchronous Serial Port Adapter Installation and Configuration*
- *PA-8T-X21 Synchronous Serial Port Adapter Installation and Configuration*
- *PA-8T-232 Synchronous Serial Port Adapter Installation and Configuration*
- *PA-4T+ Synchronous Serial Port Adapter Installation and Configuration*

Configure Bisync

To configure the Bisync feature on the synchronous serial port adapters (PA-8T-V35, PA-8T-X21, PA-8T-232, and PA-4T+) on Cisco 7200 series routers, refer to the "Block Serial Tunnelling (BSTUN)" section of the "Configuring Serial Tunnel (STUN) and Block Serial Tunnel (BSTUN)" chapter of the *Bridging and IBM Networking Configuration Guide*. All commands listed in the "Block Serial Tunnelling (BSTUN)" section apply to the synchronous serial port adapters on Cisco 7200 series routers. Any command syntax that specifies an interface *number* supports the Cisco 7200 series *slot*/*port* syntax.

Configure Half-Duplex Carrier Modes and Timers

This section describes how to configure the synchronous serial port adapters (PA-8T-V35, PA-8T-X21, PA-8T-232, and PA-4T+) on Cisco 7200 series routers. To configure the half-duplex feature on synchronous serial port adapters, perform the tasks described in the following sections, which appear earlier in this chapter:

- Understand Half-Duplex DTE and DCE State Machines
- Change between Controlled-Carrier and Constant-Carrier Modes Examples
- Tune Half-Duplex Timers

Configure Compression Service Adapters on Cisco 7500 Series Routers

The SA-Comp/1 and SA-Comp/4 data compression service adapters (CSAs) are available on Cisco 7200 series routers, on second-generation Versatile Interface Processors (VIP2s) in Cisco 7500 series routers. (CSAs require VIP2 model VIP2-40.)

These service adapters provide high-performance, hardware-based data compression capabilities via simultaneous Stacker compression data compression algorithms with independent full-duplex compression and decompression capabilities on point-to-point (PPP) encapsulated packets.

The SA-Comp/1 supports up to 64 WAN interfaces and the SA-Comp/4 supports up to 256 WAN interfaces.

On the Cisco 7200 series routers you can optionally specify which CSA the interface uses to perform hardware compression.

You can configure point-to-point compression on serial interfaces that use PPP encapsulation. Compression reduces the size of a PPP frame via lossless data compression. PPP encapsulations support both predictor and Stacker compression algorithms.

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

When you configure Stacker compression on Cisco 7200 series routers, and on Cisco 7500 series routers, there are three methods of compression: hardware compression, distributed compression, and software compression. Specifying the **compress stac** command with no options causes the router to use the fastest available compression method, as described here:

- If the router contains a compression service adapter (CSA), compression is performed in the CSA hardware (hardware compression).
- If the CSA is not available, compression is performed in the software installed on the VIP2 (distributed compression).
- If the VIP2 is not available, compression is performed in the router's main processor (software compression).

Using hardware compression in the CSA frees the router's main processor for other tasks. You can also configure the router to use the VIP2 to perform compression by using the **distributed** option, or to use the router's main processor by using the **software** option. If the VIP2 is not available, compression is performed in the router's main processor.

When compression is performed in software installed in the router's main processor, it might significantly affect system performance. You should disable compression in the router's main processor if the router CPU load exceeds 40 percent. To display the CPU load, use the **show process cpu** EXEC command.

For instructions on configuring compression over PPP, refer to the "Media-Independent PPP" chapter in the *Dial Solutions Configuration Guide*.

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 percent. 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**

Configure Real-Time Transport Protocol Header Compression

Real-time Transport Protocol (RTP) is a protocol used for carrying packetized audio and video traffic over an IP network. RTP is described in RFC 1889. RTP is not intended for data traffic, which uses TCP or UDP. RTP provides end-to-end network transport functions intended for applications with real-time requirements, such as audio, video, or simulation data over multicast or unicast network services.

For information and instructions for configuring RTP header compression, refer to the "Configuring IP Multicast Routing" chapter in the *Network Protocol Configuration Guide, Part 1*.

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.	**dxi map** *protocol address vpi vci* [**broadcast**]

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

To configure an ATM interface, see the "Configuring ATM Access over a Serial Interface" chapter in the *Wide-Area Networking Configuration Guide*.

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 7500 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

All Fast Serial Interface Processor (FSIP) interface types on the Cisco 7500 and the PA-8T and PA-4T+ synchronous serial port adapters on the Cisco 7000 series routers with RSP7000, Cisco 7200 series routers, and Cisco 7500 series routers 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 EIA/TIA-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** or **nrzi-encoding** [**mark**] (Cisco 7200 series routers and Cisco 7500 series routers)

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 Data

If the interface on the PA-8T and PA-4T+ synchronous serial port adapters is used to drive a dedicated T1 line that does not have B8ZS encoding, you must invert the data stream on the connecting CSU/DSU or on the interface. Be careful not to invert data on both the CSU/DSU and the interface because two data inversions will cancel each other out.

If the T1 channel on the CT3IP is using AMI line coding, you must invert the data. For more information, see the **t1 linecode** controller command. For more information on the CT3IP, refer to the "Configure a Channelized T3 Interface Processor" section in this chapter.

To invert the data stream, complete the following task in interface configuration mode:

Task	Command
Invert the data on an interface.	**invert data**

Invert the Transmit Clock Signal

Systems that use long cables or cables that are not transmitting the TxC signal (transmit echoed clock line, also known as TXCE or SCTE clock) can experience high error rates when operating at the higher transmission speeds. For example, if the interface on the PA-8T and PA-4T+ synchronous serial port adapters is reporting a high number of error packets, a phase shift might be the problem. Inverting the clock signal can correct this shift. To invert the clock signal, complete the following task in interface configuration mode:

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

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 or MIP. 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 or MIP.	**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**

Caution Unless you know for certain that you really need this feature, be very careful using this command. It will hide the real status of the interface. The interface could actually be down and you will not know by looking at show displays.

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 serial interface (a serial interface that meets the G.703 electrical and mechanical specifications and operates at E1 data rates). G.703 interfaces are available on port adapters for the Fast Serial Interface Processor (FSIP) on a Cisco 4000 series or Cisco 7500 series router. Configuration tasks are described in these sections:

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

Enable Framed Mode

G.703 interfaces have two modes of operation: framed and unframed. By default, G.703 serial 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 CRC4, which is useful for checking data integrity while operating in framed mode, is not generated. To enable generation of the G.703 CRC4, 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 specify 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 Source

A G.703 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 interface 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 Packet OC-3 Interface

The Cisco Packet OC-3 Interface Processor (POSIP) is available on Cisco 7500 series routers.

The POSIP is a fixed-configuration interface processor that uses second-generation Versatile Interface Processor (VIP2) technology. The POSIP provides a single 155.520-Mbps, OC-3 physical layer interface for packet-based traffic. This OC-3 interface is fully compatible with SONET and Synchronous Digital Hierarchy (SDH) network facilities and is compliant with RFC 1619, "PPP over SONET/SDH," and RFC 1662, "PPP in HDLC-like Framing." The Packet-Over-SONET specification is primarily concerned with the use of the PPP encapsulation over SONET/SDH links.

Table 13 describes the default values set in the initial configuration of a Packet OC-3 interface.

Table 13 Packet OC-3 Interface Default Configuration

Attribute	Default Value
Maximum transmission unit (MTU)	4470 bytes
Framing	SONET STS-3c framing
Loopback internal	No internal loopback
Loopback line	No line loopback
Transmit clocking	Recovered receive clock
Enabling	Shut down

Because the Packet OC-3 interface is partially configured, you might not need to change its configuration before enabling it. However, when the router is powered up, a new Packet OC-3 interface is shut down. To enable the Packet OC-3 interface, you must enter the **no shutdown** command in the global configuration mode.

Packet OC-3 Interface Configuration Task List

The values of all Packet OC-3 configuration parameters can be changed to match your network environment. Perform the optional tasks in the following sections if you need to customize the POSIP configuration:

- Select a Packet OC-3 Interface
- Set the MTU Size
- Configure Framing
- Configure an Interface for Internal Loopback
- Configure an Interface for Line Loopback
- Set the Source of the Transmit Clock
- Enable the Packet OC-3 Interface
- Assign a Network Protocol Address
- Save the Configuration

For Packet OC-3 interface configuration examples, see the "CSU/DSU Service Module Examples" section, later in this chapter.

Select a Packet OC-3 Interface

The Packet OC-3 interface is referred to as *pos* in the configuration commands. An interface is created for each POSIP found in the system at reset time.

If you need to change any of the default configuration attributes or otherwise reconfigure the Packet OC-3 interface, first complete the following task in global configuration mode:

Task	Command
Select the Packet OC-3 interface and enter interface configuration mode.	**interface pos** *slot*/*port*

Set the MTU Size

To set the maximum transmission unit (MTU) size for the interface, complete the following task in interface configuration mode:

Task	Command
Set the MTU size.	**mtu** *bytes*

The value of the *bytes* argument is in the range 64 to 4470 bytes; the default is 4470 bytes. (4470 bytes exactly matches FDDI and HSSI interfaces for autonomous switching.) The **no** form of the command restores the default.

Configure Framing

To configure framing on the Packet OC-3 interface, complete one of the following tasks in interface configuration mode:

Task	Command
Select SDH STM-1 framing.	**pos framing-sdh**
Revert to the default SONET STS-3c framing.	**no pos framing-sdh**

Configure an Interface for Internal Loopback

With the **loopback internal** command, packets from the router are looped back in the framer. Outgoing data gets looped back to the receiver without actually being transmitted. With the **loopback line** command, the receive (RX) fiber is logically connected to the transmit fiber (TX) so that packets from the remote router are looped back to it. Incoming data gets looped around and retransmitted without actually being received.

To enable or disable internal loopback on the interface, complete one of the following tasks in interface configuration mode:

Task	Command
Enable internal loopback.	**loop internal**
Disable internal loopback.	**no loop internal**

Local loopback is useful for checking that the POSIP is working. Packets from the router are looped back in the framer.

Configure an Interface for Line Loopback

Line loopback is used primarily for debugging purposes.

To enable or disable an interface for line loopback, complete one of the following tasks in interface configuration mode:

Task	Command
Enable line loopback.	**loop line**
Disable line loopback.	**no loop line**

The receive fiber (RX) is logically connected to the transmit fiber (TX) so that packets from the remote router are looped back to it.

Set the Source of the Transmit Clock

By default, the Packet OC-3 interface uses the recovered receive clock to provide transmit clocking. To change the transmit clock source, complete one of the following tasks in interface configuration mode:

Task	Command
Set the internal clock as the transmit clock source.	**pos internal-clock**
Set the recovered receive clock to provide transmit clocking.	**no pos internal-clock**

Enable the Packet OC-3 Interface

To enable the Packet OC-3 interface when it is first installed or after it has been disabled, complete the following task in interface configuration mode:

Task	Command
Enable the interface.	**no shutdown**

Assign a Network Protocol Address

You can now enter one or more network protocol addresses and otherwise configure the interface for LAN or WAN uses. For example, if IP routing is enabled on the system, perform the following task in interface configuration mode:

Task	Command
Assign an IP address and subnet mask to the interface.	**ip address** *ip-address mask*

For more information about LAN network protocol configuration, see the relevant volume and chapter of the *Network Protocols Configuration Guide*. For information about WAN configuration, see the *Wide-Area Networking Configuration Guide*.

Save the Configuration

To save the new configuration to memory, complete the following task in privileged EXEC mode.

Task	Command
Write the new configuration to memory.	**copy running-config startup-config**

Configure Serial Interfaces for CSU/DSU Service Modules

This section describes how to configure the router to support channel service unit (CSU) and data service unit (DSU) service modules:

- Fractional T1/T1 CSU/DSU
- 2-Wire and 4-Wire 56/64-kbps CSU/DSU

Fractional T1/T1 CSU/DSU Service Module Configuration Task List

To configure fractional T1 and T1 (FT1/T1) service modules, perform the tasks described inthese sections:

- Specify the Clock Source
- Enable Data Inversion before Transmission
- Specify the Frame Type of a FT/T1 Line
- Specify the CSU Line Build Out
- Specify FT1/T1 Line-Code Type
- Enable Remote Alarms
- Enable Loopcodes that Initiate Remote Loopbacks
- Specify Timeslots

Specify the Clock Source

To specify the clock source for the FT1/T1 CSU/DSU module, perform the following task in interface configuration mode:

Task	Command
Specify the clock source, for the CSU/DSU internal clock or the line clock.	**service-module t1 clock source** {**internal** | **line**}

Enable Data Inversion before Transmission

Data inversion is used to guarantee the ones density requirement on an alternate mark inversion (AMI) line when using bit-oriented protocols such as High-Level Data Link Control (HDLC), Point-to-Point Protocol (PPP), X.25, and Frame Relay.

To guarantee the ones density requirement on an AMI line using the FT1/T1 CSU/DSU module, perform the following task in interface configuration mode:

Task	Command
Invert bit codes by changing all 1 bits to 0 bits and all 0 bits to 1 bits.	**service-module t1 data-coding inverted**

If the timeslot speed is set to 56 kbps, this command is rejected because line density is guaranteed when transmitting at 56 kbps. Use this command with the 64 kbps line speed. If you transmit inverted bit codes, both CSU/DSUs must have this command configured for successful communication.

To enable normal data transmission on a FT1/T1 network, perform the following task in interface configuration mode:

Task	Command
Enable normal data transmission on a T1 network.	**service-module tx1 data-coding normal** or **no service-module t1 data-coding inverted**

Specify the Frame Type of a FT/T1 Line

To specify the frame type for a line using the FT1/T1 CSU/DSU module, perform the following task in interface configuration mode:

Task	Command
Specify a FT1/T1 frame type. Choose either D4 Super Frame (sf) or Extended Super Frame (esf).	**service-module t1 framing** {**sf** | **esf**}

In most cases, the service provider determines which framing type, either **esf** or **sf**, is required for your circuit.

Specify the CSU Line Build Out

To decrease the outgoing signal strength to an optimum value for the telecommunication carrier network, perform the following task on the FT1/T1 CSU/DSU module in interface configuration mode:

Task	Command
Decrease the outgoing signal strength in decibels.	**service-module t1 lbo** {**-15 db** | **-7.5 db**}

To transmit packets without decreasing outgoing signal strength, perform the following task in interface configuration mode:

Task	Command
Transmits packets without decreasing outgoing signal strength.	**service-module t1 lbo none**

The ideal signal strength should be between -15 dB and -22 dB, which is calculated by adding the phone company loss + cable length loss + line build out.

You may use this command in back-to-back configurations, but it is not needed on most actual T1 lines.

Specify FT1/T1 Line-Code Type

To configure the line code for the FT1/T1 CSU/DSU module, perform the following task in interface configuration mode:

Task	Command
Specify a line-code type. Choose alternate mark inversion (AMI) or binary 8 zero substitution (B8ZS).	**service-module t1 linecode** {**ami** | **b8zs**}

Configuring B8ZS is a method of ensuring the ones density requirement on a T1 line by substituting intentional bipolar violations in bit positions four and seven for a sequence of eight zero bits. When the CSU/DSU is configured for AMI, you must guarantee the ones density requirement in your router configuration using the **service-module t1 data-coding inverted** command or the **service-module t1 timeslots speed 56** command.

In most cases, your T1 service provider determines which line-code type, either **ami** or **b8zs**, is required for your T1 circuit.

Enable Remote Alarms

To generate remote alarms (yellow alarms) at the local CSU/DSU or detect remote alarms sent from the remote CSU/DSU, perform the following task in interface configuration mode:

Task	Command
Enable remote alarms.	**service-module t1 remote-alarm-enable**

Remote alarms are transmitted by the CSU/DSU when it detects an alarm condition, such as a red alarm (loss of signal) or blue alarm (unframed 1's). The receiving CSU/DSU then knows there is an error condition on the line.

With D4 super frame configured, a remote alarm condition is transmitted by setting the bit 2 of each time slot to zero. For received user data that has the bit 2 of each time slot set to zero, the CSU/DSU interprets the data as a remote alarm and interrupts data transmission, which explains why remote alarms are disabled by default. With Extended Super Frame configured, the remote alarm condition is signalled out of band in the facility data link.

You can see if the FT1/T1 CSU/DSU is receiving a remote alarm (yellow alarm) by issuing the **show service-module** command.

To disable remote alarms, perform the following task in interface configuration mode:

Task	Command
Disable remote alarms.	**no service-module t1 remote-alarm-enable**

Enable Loopcodes that Initiate Remote Loopbacks

To specify if the fractional T1/T1 CSU/DSU module goes into loopback when it receives a loopback code on the line, perform the following task in interface configuration mode:

Task	Command
Configures the remote loopback code used to transmit or accept CSU loopback requests.	**service-module t1 remote-loopback full**

Task	Command
Configures the loopback code used by the local CSU/DSU to generate or detect payload-loopback commands.	**service-module t1 remote-loopback payload** [**alternate** | **v54**]

Note By entering the **service-module t1 remote-loopback** command without specifying any keywords, you enable the standard-loopup codes, which use a 1-in-5 pattern for loopup and a 1-in-3 pattern for loopdown.

You can simultaneously configure the **full** and **payload** loopback points. However, only one loopback payload code can be configured at a time. For example, if you configure the **service-module t1 remote-loopback payload alternate** command, a payload v.54 request, which is the industry standard and default, cannot be transmitted or accepted. Full and payload loopbacks with standard-loopup codes are enabled by default.

The **no** form of this command disables loopback requests. For example, the **no service-module t1 remote-loopback full** command ignores all full-bandwidth loopback transmissions and requests. Configuring the **no** form of the command may not prevent telco line providers from looping your router in esf mode, because fractional T1/T1 telcos use facilities data-link messages to initiate loopbacks.

If you enable the **service-module t1 remote-loopback** command, the **loopback remote** commands on the FT1/T1 CSU/DSU module will not be successful.

Specify Timeslots

To define timeslots for FT1/T1 module, perform the following task in interface configuration mode:

Task	Command
Specify timeslots.	**service-module t1 timeslots** {*range* | **all**} [**speed** {**56** | **64**}]

This command specifies which timeslots are used in fractional T1 operation and determines the amount of bandwidth available to the router in each timeslot.

The *range* specifies the DS0 timeslots that constitute the FT1/T1 channel. The range is from 1 to 24, where the first timeslot is numbered 1 and the last timeslot is numbered 24. Specify this field by using a series of subranges separated by commas. The timeslot range must match the timeslots assigned to the channel group. In most cases, the service provider defines the timeslots that comprise a channel group. Use the **no** form of this command to select all FT1/T1 timeslots transmitting at 64 kbps, which is the default.

To use the entire T1 line, enable the **service-module T1 timeslots all** command.

2-Wire and 4-Wire 56/64-kbps CSU/DSU Service Module Configuration Task List

To configure 2- and 4-wire 56/64 kbps service modules, perform the tasks described in these sections:

- Set the Clock Source
- Set the Network Line Speed
- Enable Scrambled Data Coding
- Change between Digital Data Service and Switched Dial-Up Modes
- Enable Acceptance of a Remote Loopback Request
- Select a Service Provider

Set the Clock Source

In most applications, the CSU/DSU should be configured with the **service-module 56k clock source line** command. For back-to-back configurations, use the **internal** keyword to configure one CSU/DSU and use the **line** keyword to configure the other CSU/DSU.

To configure the clock source for a 4-wire 56/64-kbps CSU/DSU module, perform the following task for a serial interface in interface configuration mode:

Task	Command
Configure the clock source.	**service-module 56k clock source** {**line** | **internal**}

Use the **no** form of this command to revert to the default clock source, which is the line clock.

Set the Network Line Speed

To configure the network line speed for a 4-wire 56/64-kbps CSU/DSU module, perform the following task for a serial interface in interface configuration mode:

Task	Command
Set the network line speed.	**service-module 56k clock rate** *speed*

You can use the following line speed settings: 2.4, 4.8, 9.6, 19.2, 38.4, 56, 64 kpbs, and an **auto** setting.

The 64-kbps line speed cannot be used with back-to-back digital data service (DDS) lines. The subrate line speeds are determined by the service provider.

Only the 56-kbps line speed is available in switched mode. Switched mode is the default on the 2-wire CSU/DSU and is enabled by the **service-module 56k network-type** interface configuration command on the 4-wire CSU/DSU.

The **auto** linespeed setting enables the CSU/DSU to decipher current line speed from the sealing current running on the network. Because back-to-back DDS lines do not have sealing current, use the **auto** setting only when transmitting over telco DDS lines and using the line clock as the clock source.

Use the **no** form of this command to enable a network line speed of 56 kbps, which is the default.

Enable Scrambled Data Coding

To prevent application data from replicating loopback codes when operating at 64-kbps on a 4-wire CSU/DSU, perform the following task for a serial interface in interface configuration mode:

Task	Command
Scramble bit codes before transmission.	**service-module 56k data-coding scrambled**

Enable the scrambled configuration only in 64-kbps digital data service (DDS) mode. If the network type is set to switched, the configuration is refused.

If you transmit scrambled bit codes, both CSU/DSUs must have this command configured for successful communication.

To enable normal data transmission for the 4-wire 56/64-kbps module, perform the following task for a serial interface in interface configuration mode:

Task	Command
Specify normal data transmission.	**service-module 56k data-coding normal** or **no service-module 56k data-coding**

Change between Digital Data Service and Switched Dial-Up Modes

To transmit packets in Digital Data Service (DDS) mode or switched dial-up mode using the 4-wire 56/64-kbps CSU/DSU module, perform the following task for a serial interface in interface configuration mode:

Task	Command
Transmit packets in switched dial-up mode or DDS mode.	**service-module 56k network-type dds** or **service-module 56k network-type switched**

Use the **no** form of these commands to transmit from a dedicated leased line in DDS mode. DDS is enabled by default for the 4-wire CSU/DSU. Switched is enabled by default for the 2-wire CSU/DSU.

In switched mode, you need additional dialer configuration commands to configure dial-out numbers. Before you enable the **service-module 56k network-type switched** command, both CSU/DSU's must use a clock source coming from the line and the clock rate configured to **auto** or **56k** kbps. If the clock rate is not set correctly, this command will not be accepted.

The 2-wire and 4-wire 56/64-kbps CSU/DSU modules use V.25 *bis* dial commands to interface with the router. Therefore, the interface must be configured using the **dialer in-band** command. DTR dial is not supported.

Note Any loopbacks in progress are terminated when switching between modes.

Enable Acceptance of a Remote Loopback Request

To enable the acceptance of a remote loopback request on a 2- or 4-wire 56/64-kbps CSU/DSU module, perform the following task for a serial interface in interface configuration mode:

Task	Command
Enable a remote loopback request.	**service-module 56k remote-loopback**

The **no service-module 56k remote-loopback** command prevents the local CSU/DSU from being placed into loopback by remote devices on the line. Unlike the T1 module, the 2- or 4-wire 56/64-kbps CSU/DSU module can still initiate remote loopbacks with the **no** form of this command configured.

Select a Service Provider

To select a service provider to use with a 2- or 4-wire 56/64 kbps dial-up line, perform the following task for a serial interface in interface configuration mode:

Task	Command
Select a service provider for a 2 or 4 wire switched 56/64 kbps dialup line.	**service-module 56k switched-carrier** {**att** | **other** | **sprint**}

The **att** keyword specifies AT&T or another digital network service provider as the line carrier, which is the default for the 4-wire 56/64-kbps CSU/DSU module. The **sprint** keyword specifies Sprint or another service provider whose network carries mixed voice and data as the line carrier, which is the default for the 2-wire switched 56-kbps CSU/DSU module.

In a Sprint network, echo-canceler tones are sent during call setup to prevent echo cancelers from damaging digital data. The transmission of these cancelers may increase call setup times by 8 seconds on the 4-wire module. Having echo cancellation enabled does not affect data traffic.

This configuration command is ignored if the network type is DDS.

Use the **no** form of this command to enable the default service provider. AT&T is enabled by default on the 4-wire 56/64 module. Sprint is enabled by default on the 2-wire switched 56 module.

Configure Low-Speed Serial Interfaces

This section describes how to configure low-speed serial interfaces. In addition to the background information described in the "Understand Half-Duplex DTE and DCE State Machines" section, these configuration guidelines are provided for configuring low-speed serial interfaces:

- Change between Controlled-Carrier and Constant-Carrier Modes
- Tune Half-Duplex Timers
- Change between Synchronous and Asynchronous Modes

See the "Serial Interface Configuration Examples" section at the end of this chapter for configuration examples.

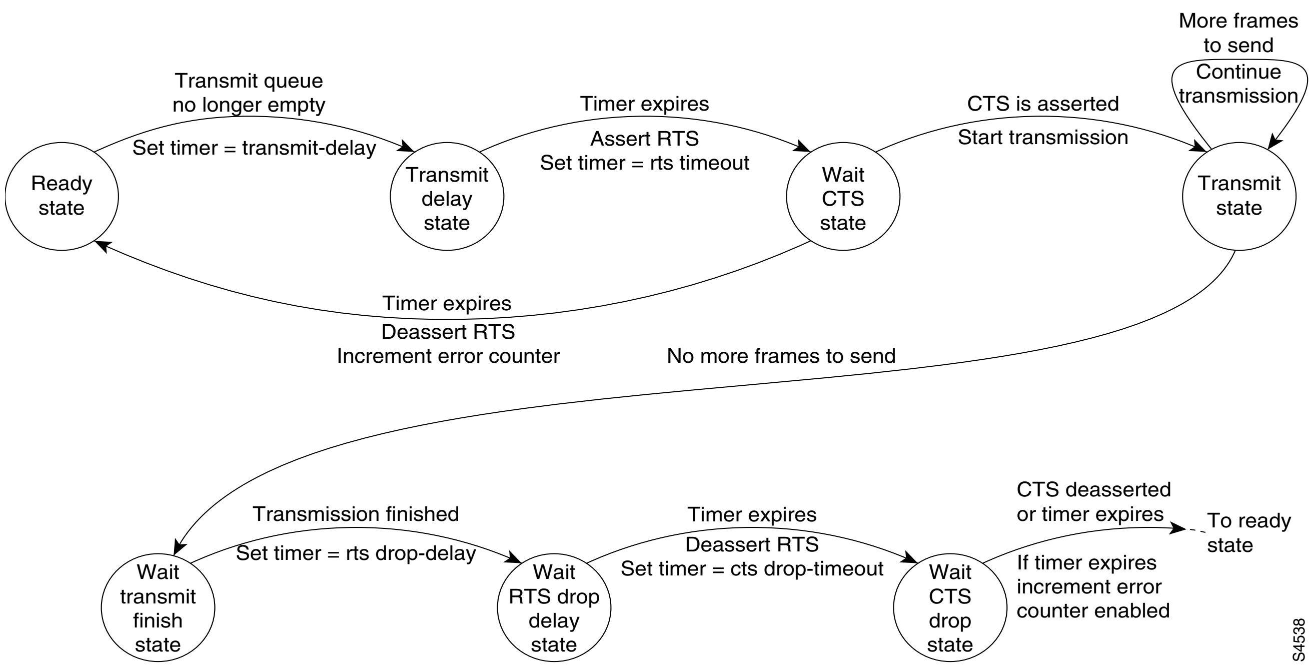
Understand Half-Duplex DTE and DCE State Machines

The following section describes the communication between half-duplex DTE transmit and receive state machines and half-duplex DCE transmit and receive state machines.

Half-Duplex DTE State Machines

As shown in Figure 30, the half-duplex DTE transmit state machine for low-speed interfaces remains in the ready state when it is quiescent. When a frame is available for transmission, the state machine enters the transmit delay state and waits for a time period, which is defined by the **half-duplex timer transmit-delay** command. The default is 0 ms. Transmission delays are used for debugging half-duplex links and assisting lower-speed receivers that cannot process back-to-back frames.

Figure 30 Half-Duplex DTE Transmit State Machine



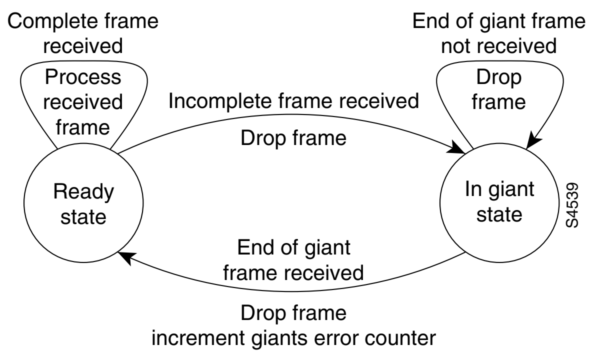
After idling for a defined number of milliseconds, the state machine asserts a request to send (RTS) signal and changes to the wait-clear-to-send (CTS) state for the data communications equipment (DCE) to assert CTS. A timeout timer with a value set by the **half-duplex timer rts-timeout** command starts. This default is 3 ms. If the timeout timer expires before CTS is asserted, the state machine returns to the ready state and deasserts RTS. If CTS is asserted prior to the timer's expiration, the state machine enters the transmit state and sends the frames.

Once there are no more frames to transmit, the state machine transitions to the wait transmit finish state. The machine waits for the transmit first in first out (FIFO) in the serial controller to empty, starts a delay timer with a value defined by the **half-duplex timer rts-drop-delay** interface command, and transitions to the wait RTS drop delay state.

When the timer in the wait RTS drop delay state expires, the state machine deasserts RTS and transitions to the wait CTS drop state. A timeout timer with a value set by the **half-duplex timer cts-drop-timeout** interface command starts, and the state machine waits for the CTS to deassert. The default is 250 ms. Once the CTS signal is deasserted or the timeout timer expires, the state machine transitions back to the ready state. If the timer expires before CTS is deasserted, an error counter is incremented, which can be displayed by issuing the **show controllers** command for the serial interface in question.

As shown in Figure 31, a half-duplex DTE receive state machine for low-speed interfaces idles and receives frames in the ready state. A giant frame is any frame whose size exceeds the maximum transmission unit (MTU). If the beginning of a giant frame is received, the state machine transitions to the in giant state and discards frame fragments until it receives the end of the giant frame. At this point, the state machine transitions back to the ready state and waits for the next frame to arrive.

Figure 31 Half-Duplex DTE Receive State Machine

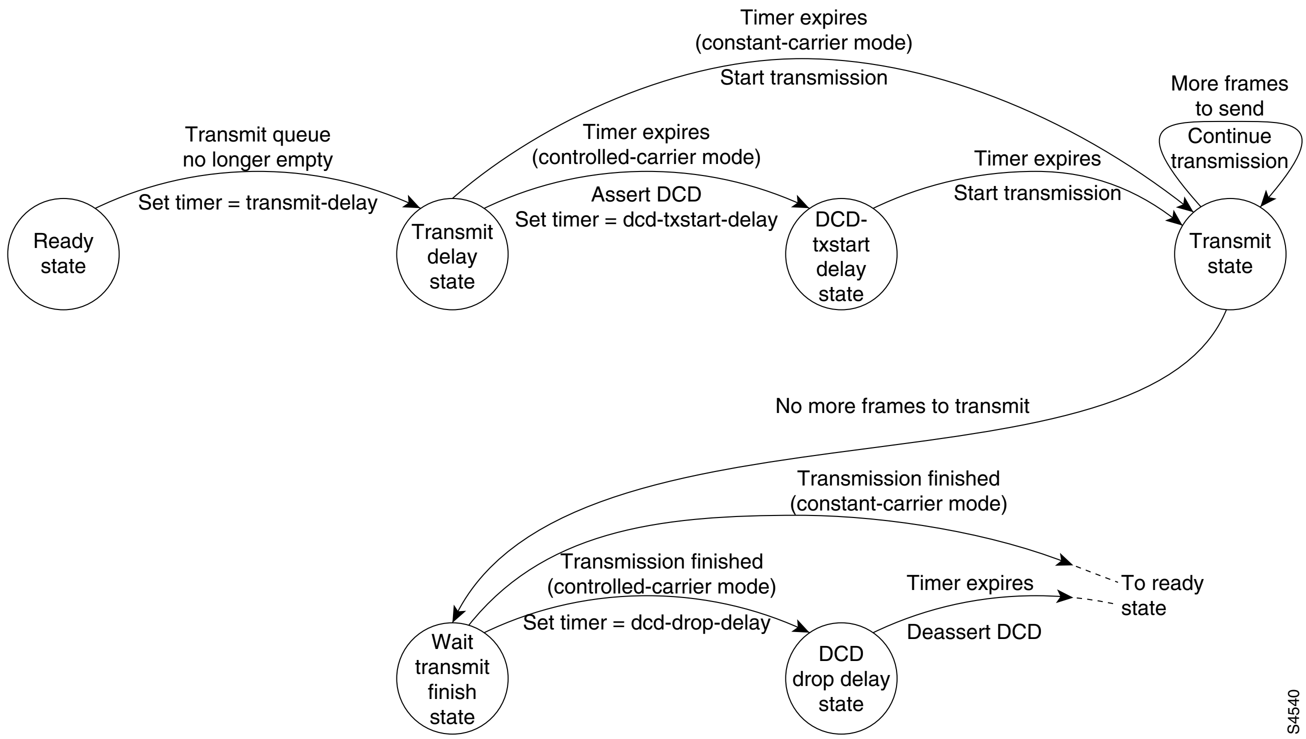


An error counter is incremented upon receipt of the giant frames. To view the error counter, enter the **show interface** command for the serial interface in question.

Half-Duplex DCE State Machines

As shown in Figure 32, for a low-speed serial interface in DCE mode, the half-duplex DCE transmit state machine idles in the ready state when it is quiescent. When a frame is available for transmission on the serial interface, such as when the output queues are no longer empty, the state machine starts a timer (based on the value of the **transmit-delay** command, in milliseconds) and transitions to the transmit delay state. Similar to the DTE transmit state machine, the transmit delay state gives you the option of setting a delay between the transmission of frames; for example, this feature lets you compensate for a slow receiver that loses data when multiple frames are received in quick succession. The default **transmit-delay** value is 0 ms; use the **half-duplex timer transmit-delay** interface configuration command to specify a delay value not equal to 0.

Figure 32 Half-Duplex DCE Transmit State Machine



After the transmit delay state, the next state depends on whether the interface is in constant-carrier mode (the default) or controlled-carrier mode.

If the interface is in constant-carrier mode, it passes through the following states:

1 The state machine passes to the transmit state when the **transmit-delay** timer expires. The state machine stays in the transmit state until there are no more frames to transmit.

2 When there are no more frames to transmit, the state machine passes to the wait transmit finish state, where it waits for the transmit FIFO to empty.

3 Once the FIFO empties, the DCE passes back to the ready state and waits for the next frame to appear in the output queue.

If the interface is in controlled-carrier mode, the interface performs a handshake using the data carrier detect (DCD) signal. In this mode, DCD is deasserted when the interface is idle and has nothing to transmit. The transmit state machine transitions through the states as follows:

1 After the **transmit-delay** timer expires, the DCE asserts DCD and transitions to the DCD-txstart delay state to ensure a time delay between the assertion of DCD and the start of transmission. A timer with the value **dcd-txstart-delay** is started. (This timer has a default value of 100 ms; use the **half-duplex timer dcd-txstart-delay** interface configuration command to specify a delay value.)

2 When this delay timer expires, the state machine transitions to the transmit state and transmits frames until there are no more frames to transmit.

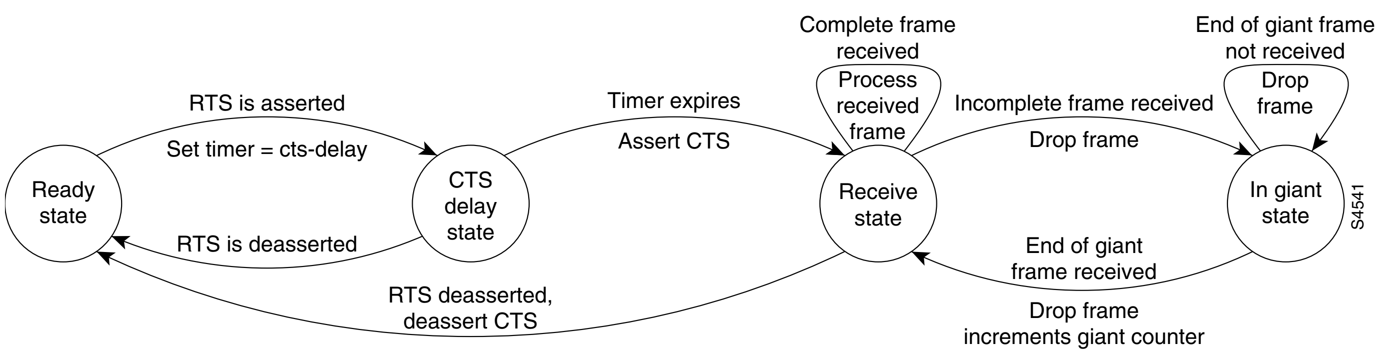
3 After the DCE transmits the last frame, it transitions to the wait transmit finish state, where it waits for transmit FIFO to empty and the last frame to transmit to the wire. Then DCE starts a delay timer with the value **dcd-drop-delay**. (This timer has the default value of 100 ms; use the **half-duplex timer dcd-drop-delay** interface configuration command to specify a delay value.)

4 The DCE transitions to the wait DCD drop delay state. This state causes a time delay between the transmission of the last frame and the deassertion of DCD in the controlled-carrier mode for DCE transmits.

5 When the timer expires, the DCE deasserts DCD and transitions back to the ready state and stays there until there is a frame to transmit on that interface.

As shown in Figure 33, the half-duplex DCE receive state machine idles in the ready state when it is quiescent. It transitions out of this state when the DTE asserts RTS. In response, the DCE starts a timer with the value **cts-delay**. This timer delays the assertion of CTS because some DTE interfaces expect this delay. (The default value of this timer is 0 ms; use the **half-duplex timer cts-delay** interface configuration command to specify a delay value.)

Figure 33 Half-Duplex DCE Receive State Machine



When the timer expires, the DCE state machine asserts CTS and transitions to the receive state. It stays in the receive state until there is a frame to receive. If the beginning of a giant frame is received, it transitions to the in giant state and keeps discarding all the fragments of the giant frame and transitions back to the receive state.

Transitions back to the ready state occur when RTS is deasserted by the DTE. The response of the DCE to the deassertion of RTS is to deassert CTS and go back to the ready state.

Change between Controlled-Carrier and Constant-Carrier Modes

The **half-duplex controlled-carrier** command enables you to change between controlled-carrier and constant-carrier modes for low-speed serial DCE interfaces in half-duplex mode. Configure a serial interface for half-duplex mode by using the **half-duplex** command. Full-duplex mode is the default for serial interfaces. This interface configuration is available on Cisco 2520 through Cisco 2523 routers.

Controlled-carrier operation means that the DCE interface will have DCD deasserted in the quiescent state. When the interface has something to transmit, it will assert DCD, wait a user-configured amount of time, then start the transmission. When it has finished transmitting, it will again wait a user-configured amount of time, then deassert DCD.

Place a Low-Speed Serial Interface in Controlled-Carrier Mode

To place a low-speed serial interface in controlled-carrier mode, perform the following task in interface configuration mode:

Task	Command
Place a low-speed serial interface in controlled-carrier mode.	**half-duplex controlled-carrier**

Place a Low-Speed Serial Interface in Constant-Carrier Mode

To return a low-speed serial interface to constant-carrier mode from controlled-carrier mode, perform the following task in interface configuration mode:

Task	Command
Place a low-speed serial interface in constant-carrier mode.	**no half-duplex controlled-carrier**

Tune Half-Duplex Timers

To tune half-duplex timers, perform the following task in interface configuration mode:

Task	Command
Tune half-duplex timers.	**half-duplex timer** {**cts-delay** *value* | **cts-drop-timeout** *value* | **dcd-drop-delay** *value* | **dcd-txstart-delay** *value* | **rts-drop-delay** *value* | **rts-timeout** *value* | **transmit-delay** *value*}

The timer tuning commands permit you to adjust the timing of the half-duplex state machines to suit the particular needs of their half-duplex installation.

Note that the **half-duplex timer** command and its options deprecates the following two timer tuning commands that are available only on high-speed serial interfaces:

- **sdlc cts-delay**
- **sdlc rts-timeout**

Change between Synchronous and Asynchronous Modes

To specify the mode of a low-speed serial interface as either synchronous or asynchronous, perform the following task in interface configuration mode:

Task	Command
Specify the mode of a low-speed interface as either synchronous or asynchronous.	**physical-layer** {**sync** | **async**}

This command applies only to low-speed serial interfaces available on Cisco 2520 through Cisco 2523 routers.

In synchronous mode, low-speed serial interfaces support all interface configuration commands available for high-speed serial interfaces, except the following two commands:

- **sdlc cts-delay**
- **sdlc rts-timeout**

When placed in asynchronous mode, low-speed serial interfaces support all commands available for standard asynchronous interfaces. The default is synchronous mode.

Note that when you enter this command, it does not appear in the output of **show running config** and **show startup config** command, because the command is a physical-layer command.

Return a Low-Speed Serial Interface to Synchronous Mode

To return to the default mode (synchronous) of a low-speed serial interface on a Cisco 2520 through Cisco 2523 router, perform the following task in interface configuration mode:

Task	Command
Return the interface to its default mode, which is synchronous.	**no physical-layer**

Serial Interface Configuration Examples

This section includes the following example groups:

- Channelized T3 Interface Processor Configuration Examples
- Low-Speed Serial Interface Examples
- CSU/DSU Service Module Examples

Channelized T3 Interface Processor Configuration Examples

The examples in this section show how to configure the Channelized T3 Interface Processor (CT3IP). The first example shows how to configure two of the T1 channels of the channelized T3 controller. The second example shows how to configure one of the T1 channels of the channelized T3 controller as an external port for further channelization on the Multichannel Interface Processor (MIP).

For more information, refer to the “Configure the T3 Controller” and “Configure External T1 Channels” sections earlier in this chapter. Examples included in this section are:

- CT3IP Configuration with Default Values Accepted Example
- CT3IP External Ports Configuration Example
- CT3IP Maintenance Data Link (MDL) Example
- CT3IP Performance Monitoring Example
- CT3IP BERT Test Pattern Example
- CT3IP Remote FDL Loopback Example

CT3IP Configuration with Default Values Accepted Example

In the following example, timeslots 1 through 24 (the entire T1 bandwidth) are assigned to T1 channel 16 and timeslots 1 through 5 and 20 through 23 (fractional T1 bandwidth) are assigned to T1 channel 10 for the CT3IP in slot 9. The default framing, cable length, and clock source are accepted for the T3, and the default speed, framing, clock source, and line code are accepted for each T1 channel. Each T1 channel is assigned an IP address. Other interface configuration commands can be assigned to the T1 channel at this time.

```
controller t3 9/0/0
 t1 16 timeslot 1-24
 t1 10 timeslot 1-5,20-23
interface serial 9/0/0:16
 ip address 10.20.20.1 255.255.255.0
interface serial 9/0/0:10
 ip address 10.20.20.3 255.255.255.0
```

CT3IP External Ports Configuration Example

In the following example, T1 channel 1 on the CT3IP in slot 9 is broken out as an external port so that it can be further channelized on the MIP in slot 3. The cable length is 300 feet, and the default line coding format on the T1 channel is used. Because the default line coding format on the T1 channel is B8ZS and the default line coding on the MIP is AMI, the line coding on the MIP is changed to B8ZS.

```
controller t3 9/0/0
 t1 external 1 cablelength 300
controller t1 3/0
 linecode b8zs
 channel-group 1 timeslots 1
interface serial 3/0:1
 ip address 10.20.20.5 255.255.255.0
```

CT3IP Maintenance Data Link (MDL) Example

The following examples show several of the Maintenance Data Link (MDL) messages for the CT3IP in slot 9:

```
controller t3 9/0/0
 mdl string eic Router C
 mdl string lic Network A
 mdl string fic Bldg 102
 mdl string unit 123ABC
```

CT3IP Performance Monitoring Example

In the following example, the performance reports are generated for T1 channel 6 on the CT3IP in slot 9:

```
controller t3 9/0/0
 t1 6 fdl ansi
```

CT3IP BERT Test Pattern Example

The following example shows how to enable a BERT test pattern that consists of a repeating pattern of ones (...111...) and runs for 30 minutes for T1 channel 8 on CT3IP in slot 9:

```
controller t3 9/0/0
 t1 8 bert pattern 1s interval 30
```

CT3IP Remote FDL Loopback Example

The following example shows how to enable a remote payload FDL ANSI bit loopback for T1 channel 6 on CT3IP in slot 3:

```
interface serial 3/0/0:6
 loopback remote payload fdl ansi
```

Low-Speed Serial Interface Examples

These configuration examples are provided for low-speed serial interfaces:

- Set Synchronous or Asynchronous Mode Examples
- Change between Controlled-Carrier and Constant-Carrier Modes Examples
- Tune Half-Duplex Timers Example
- Cisco 40000 Series Router with 2T16S Serial Network Processor Example

Set Synchronous or Asynchronous Mode Examples

The following example shows how to change a low-speed serial interface from synchronous to asynchronous mode:

```
interface serial 2
 physical-layer async
```

The following examples show how to change a low-speed serial interface from asynchronous mode back to its default synchronous mode:

```
interface serial 2
 physical-layer sync
```

or

```
interface serial 2
 no physical-layer
```

The following example shows some typical asynchronous interface configuration commands:

```
interface serial 2
 physical-layer async
 ip address 1.0.0.2 255.0.0.0
 async default ip address 1.0.0.1
 async mode dedicated
 async default routing
```

The following example shows some typical synchronous serial interface configuration commands available when the interface is in synchronous mode:

```
interface serial 2
 physical-layer sync
 ip address 1.0.0.2 255.0.0.0
 no keepalive
 ignore-dcd
 nrzi-encoding
 no shutdown
```

Change between Controlled-Carrier and Constant-Carrier Modes Examples

The following example shows how to change to controlled-carrier mode from the default of constant-carrier operation:

```
interface serial 2
 half-duplex controlled-carrier
```

The following example shows how to change to constant-carrier mode from controlled-carrier mode:

```
interface serial 2
 no half-duplex controlled-carrier
```

Tune Half-Duplex Timers Example

The following examples show how to set the cts-delay timer to 1234 ms and the transmit-delay timer to 50 ms.

```
interface serial 2
 half-duplex timer cts-delay 1234
 half-duplex timer transmit-delay 50
```

Cisco 40000 Series Router with 2T16S Serial Network Processor Example

The 2T16S network processor module provides high-density serial interfaces for the Cisco 4000 series routers. This module has two high-speed interfaces that support full-duplex T1 and E1 rates (up to 2 MB per second) and 16 low-speed interfaces. The 16 lower-speed ports can be individually configured as either synchronous ports at speeds up to 128 kbps or as asynchronous ports at speeds up to 115 kbps.

For the slow-speed interfaces, both synchronous and asynchronous serial protocols are supported. For for the high-speed interfaces, only the synchronous protocols are supported. Synchronous protocols include IBM's BSC, SDLC, and HDLC. Asynchronous protocols include PPP, SLIP, and ARAP for dial-up connections using external modems.

This example shows a Cisco 4500 router equipped with two 2T16S serial network processor modules and two conventional Ethernet ports.

This router is configured for WAN aggregation using X.25, Frame Relay, PPP, and HDLC encapsulation. Serial interfaces 0, 1, 18, and 19 are the synchronous high-speed interfaces. Serial interfaces 2 through 17 and 20 through 35 are the synchronous/asynchronous low-speed interfaces.

```
 version 11.2
!
 hostname c4X00
!
 username brad password 7 13171F1D0A080139
 username jim password 7 104D000A0618
!
```

Ethernet interfaces and their subinterfaces are configured for LAN access.

```
 interface Ethernet0
  ip address 10.1.1.1 255.255.255.0
  media-type 10BaseT
 !
 interface Ethernet1
  ip address 10.1.2.1 255.255.255.0
  media-type 10BaseT
 !
```

Interfaces Serial 0 and Serial 1 are the high-speed serial interfaces on the first 2T16S module. In this example, subinterfaces are also configured for remote offices connected in to interface Serial 0.

```
interface Serial0
 description Frame relay configuration sample
 no ip address
 encapsulation frame-relay
!
interface Serial0.1 point-to-point
 description PVC to first office
 ip address 10.1.3.1 255.255.255.0
 frame-relay interface-dlci 16
!
interface Serial0.2 point-to-point
 description PVC to second office
 ip address 10.1.4.1 255.255.255.0
 frame-relay interface-dlci 17
!
interface Serial1
 description X25 configuration sample
 ip address 10.1.5.1 255.255.255.0
 no ip mroute-cache
 encapsulation x25
 x25 address 6120184321
 x25 htc 25
 x25 map ip 10.1.5.2 6121230073
```

Serial interfaces 2 to 17 are the low-speed interfaces on the 2T16S network processor module. In this example, remote routers are connected to various configurations.

```
interface Serial2
 description DDR connection router dial out to remote sites only
 ip address 10.1.6.1 255.255.255.0
 dialer in-band
 dialer wait-for-carrier-time 60
 dialer string 0118527351234
 pulse-time 1
 dialer-group 1
!
interface Serial3
 description DDR interface to answer calls from remote office
 ip address 10.1.7.1 255.255.255.0
 dialer in-band
!
interface Serial4
 description configuration for PPP interface
 ip address 10.1.8.1 255.255.255.0
 encapsulation ppp
!
 interface Serial5
 description Frame relay configuration sample
 no ip address
 encapsulation frame-relay
!
interface Serial5.1 point-to-point
 description PVC to first office
 ip address 10.1.9.1 255.255.255.0
 frame-relay interface-dlci 16
!
interface Serial5.2 point-to-point
 description PVC to second office
 ip address 10.1.10.1 255.255.255.0
 frame-relay interface-dlci 17
!
```

```
interface Serial6
 description configuration for PPP interface
 ip address 10.1.11.1 255.255.255.0
 encapsulation ppp
!
interface Serial7
 no ip address
 shutdown
!
interface Serial8
 ip address 10.1.12.1 255.255.255.0
 encapsulation ppp
 async default routing
 async mode dedicated
!
interface Serial9
 physical-layer async
 ip address 10.1.13.1 255.255.255.0
 encapsulation ppp
 async default routing
 async mode dedicated
!
interface Serial10
 physical-layer async
 no ip address
!
interface Serial11
 no ip address
 shutdown
!
interface Serial12
 physical-layer async
 no ip address
 shutdown
!
interface Serial13
 no ip address
 shutdown
!
interface Serial14
 no ip address
 shutdown
!
interface Serial15
 no ip address
 shutdown
!
interface Serial16
 no ip address
 shutdown
!
interface Serial17
 no ip address
 shutdown
```

Interface serial 18 is the first high-speed serial interface of the second 2T16S module. Remote sites on different subnets are dialing in to this interface with point-to-point and multipoint connections.

```
interface Serial18
 description Frame relay sample
 no ip address
 encapsulation frame-relay
!
```

```
interface Serial18.1 point-to-point
 description Frame relay subinterface
 ip address 10.1.14.1 255.255.255.0
 frame-relay interface-dlci 16
!
interface Serial18.2 point-to-point
 description Frame relay subinterface
 ip address 10.1.15.1 255.255.255.0
 frame-relay interface-dlci 17
!
interface Serial18.3 point-to-point
 description Frame relay subinterface
 ip address 10.1.16.1 255.255.255.0
 frame-relay interface-dlci 18
!
interface Serial18.5 multipoint
 ip address 10.1.17.1 255.255.255.0
 frame-relay map ip 10.1.17.2 100 IETF
```

This second high-speed serial interface is configured to connect a X.25 link. Serial interfaces 20 through 35 are the low-speed interfaces. However, some of the interfaces are not displayed in this example.

```
interface Serial19
 description X25 sample config
 ip address 10.1.18.1 255.255.255.0
 no ip mroute-cache
 encapsulation x25
 x25 address 6120000044
 x25 htc 25
 x25 map ip 10.1.18.2 6120170073
!
interface Serial20
 ip address 10.1.19.1 255.255.255.0
!
interface Serial21
 physical-layer async
 ip unnumbered e0
 encap ppp
 async mode dedicated
 async dynamic routing
 ipx network 45
 ipx watchdog-spoof
 dialer in-band
 dialer-group 1
 ppp authentication chap
!
interface Serial22
 no ip address
 shutdown
!
interface Serial23
 no ip address
 shutdown
!
interface Serial24
 no ip address
 shutdown
!
!Serial interfaces 23 through 35 would appear here.
!...

 router eigrp 10
 network 10.0.0.0
```

```
!
 dialer-list 1 protocol ip permit
!
 line con 0
 exec-timeout 15 0
 password david
 login
```

The following basic line configuration configures some of the modules' low-speed serial interfaces.

```
line 8 10
 modem InOut
 transport input all
 rxspeed 64000
 txspeed 64000
 flowcontrol hardware
line 12
 transport input all
 rxspeed 64000
 txspeed 64000
 flowcontrol hardware
 modem chat-script generic
line 21
 transport input all
 rxspeed 64000
 txspeed 64000
 flowcontrol hardware
!
 end
```

CSU/DSU Service Module Examples

Two main categories of service module examples are provided:

- FT1/T1 Examples
- 2- and 4-Wire 56/64-kpbs Service Module Examples

FT1/T1 Examples

FT1/T1 examples are provided for these configurations:

- Specify a T1 Frame Type Example
- Specify the CSU Line Build Out Example
- Specify T1 Line-Code Type Example
- Enable Loopcodes Example
- Specify Timeslots Example
- Display a Performance Report Example
- Enable Loopback Line Examples
- Loopback DTE Example
- Setting the Clock Source Example

Specify a T1 Frame Type Example

The following example enables super frame as the FT1/T1 frame type:

```
service-module t1 framing sf
```

Specify the CSU Line Build Out Example

The following example shows a line build out setting of -7.5 dB:

```
service-module t1 lbo -7.5db
```

Specify T1 Line-Code Type Example

The following example specifies AMI as the line-code type:

```
service-module t1 linecode ami
```

Enable Loopcodes Example

The following interactive example displays two routers connected back-to-back through an FT1/T1 line:

```
router# no service-module t1 remote-loopback full
router# service-module t1 remote-loopback payload alternate

router# loopback remote full
%SERVICE_MODULE-5-LOOPUPFAILED: Unit 0 - Loopup of remote unit failed

router# service-module t1 remote-loopback payload v54
router# loopback remote payload
%SERVICE_MODULE-5-LOOPUPFAILED: Unit 0 - Loopup of remote unit failed

router# service-module t1 remote-loopback payload alternate
router# loopback remote payload
%SERVICE_MODULE-5-LOOPUPREMOTE: Unit 0 - Remote unit placed in loopback
```

Specify Timeslots Example

The following example displays a series of timeslot ranges and a speed of 64 kbps:

```
Router# service-module t1 timeslots 1-10,15-20,22 speed 64
```

Display a Performance Report Example

The following example is sample output from the **show service-module** command:

```
Router1# show service-module s 0
Module type is T1/fractional
    Hardware revision is B, Software revision is 1.1i,
    Image checksum is 0x21791D6, Protocol revision is 1.1
Receiver has AIS alarm,
Unit is currently in test mode:
    line loopback is in progress
Framing is ESF, Line Code is B8ZS, Current clock source is line,
Fraction has 24 timeslots (64 Kbits/sec each), Net bandwidth is 1536 Kbits/sec.
Last user loopback performed:
    remote loopback
    Failed to loopup remote
Last module self-test (done at startup): Passed
```

```
Last clearing of alarm counters 0:05:50
    loss of signal        :    1, last occurred 0:01:50
    loss of frame         :    0,
    AIS alarm             :    1, current duration 0:00:49
    Remote alarm          :    0,
    Module access errors  :    0,
Total Data (last 0 15 minute intervals):
    1466 Line Code Violations, 0 Path Code Violations
    0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
    0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
Data in current interval (351 seconds elapsed):
    1466 Line Code Violations, 0 Path Code Violations
    25 Slip Secs, 49 Fr Loss Secs, 40 Line Err Secs, 1 Degraded Mins
    0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 49 Unavail Secs
```

Enable Loopback Line Examples

The following example shows how to configure a payload loopback:

```
Router1# loopback line payload
Loopback in progress
Router1# no loopback line
```

The following example shows the output when you loop a packet in switched mode without an active connection:

```
Router1# service-module 56k network-type switched
Router1# loopback line payload
Need active connection for this type of loopback
% Service module configuration command failed: WRONG FORMAT.
```

Loopback DTE Example

The following example loops a packet from a module to the serial interface:

```
Router1# loopback dte
Loopback in progress
Router1# ping 12.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/12/28 ms
```

Setting the Clock Source Example

The following example shows a router using internal clocking while transmitting frames at 38.4 kbps:

```
Router1# service-module 56k clock source internal
Router1# service-module 56k clock rate 38.4
```

2- and 4-Wire 56/64-kpbs Service Module Examples

2- and 4-wire 56/64 kpbs service module examples are provided for these configurations:

- Set the Network Line Speed Examples
- Enable Scrambled Data Coding Example
- Enable Switched Dial-Up Mode Example
- Display a Performance Report Example

- Remote Loopback Request Example
- Select a Service Provider Example

Set the Network Line Speed Examples

The following interactive example displays two routers connected in back-to-back DDS mode. However, the configuration fails because the **auto** rate is used.

```
Router1# service-module 56k clock source internal
Router1# service-module 56k clock rate 38.4

Router2# service-module 56k clock rate auto
% WARNING - auto rate will not work in back-to-back DDS.

a1# ping 10.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Router2# service-module 56k clock rate 38.4

Router1# ping 10.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/54/56 ms
```

When transferring from DDS mode to switched mode, you must set the correct clock rate, as shown in the following example:

```
Router2# service-module 56k network-type dds
Router2# service-module 56k clock rate 38.4
Router2# service-module 56k network-type switched
% Have to use 56k or auto clock rate for switched mode
% Service module configuration command failed: WRONG FORMAT.

Router2# service-module 56k clock rate auto
% WARNING - auto rate will not work in back-to-back DDS.
Router2# service-module 56k network-type switched
```

Enable Scrambled Data Coding Example

The following example scrambles bit codes in 64 kbps DDS mode:

```
Router# service-module 56k clock rate 56
Router# service-module 56k data-coding scrambled
Can configure scrambler only in 64k speed DDS mode
% Service module configuration command failed: WRONG FORMAT.
Router# service-module 56k clock rate 64
Router# service-module 56k data-coding scrambled
```

Enable Switched Dial-Up Mode Example

The following example displays transmission in switched dial-up mode:

```
Router# service-module 56k clock rate 19.2
Router# service-module 56k network-type switched
% Have to use 56k or auto clock rate for switched mode
% Service module configuration command failed: WRONG FORMAT.
Router# service-module 56k clock rate auto
```

```
Router# service-module 56k network-type switched
Router# dialer in-band
Router# dialer string 2576666
Router# dialer-group 1
```

Display a Performance Report Example

The following example is sample output from the **show service-module serial** command:

```
Router1# show service-module serial 1

Module type is 4-wire Switched 56
    Hardware revision is B, Software revision is X.07,
    Image checksum is 0x45354643, Protocol revision is 1.0
Connection state: active,
Receiver has loss of signal, loss of sealing current,
Unit is currently in test mode:
    line loopback is in progress
Current line rate is 56 Kbits/sec
Last user loopback performed:
    dte loopback
    duration 00:00:58
Last module self-test (done at startup): Passed
Last clearing of alarm counters 0:13:54
    oos/oof              :    3, last occurred 0:00:24
    loss of signal       :    3, current duration 0:00:24
    loss of sealing curren:    2, current duration 0:04:39
    loss of frame        :    0,
    rate adaption attempts:    0,
```

Remote Loopback Request Example

The following example enables you to transmit and receive remote loopbacks using the **service-module 56k remote-loopback** command:

```
service-module 56k remote-loopback
```

Select a Service Provider Example

The following example selects AT&T as the service provider:

```
service-module 56k network-type switched
service-module 56k switched-carrier att
```